

8TH SOUND AND MUSIC
COMPUTING
CONFERENCE

Creativity rethinks science

6-9 July 2011 – University of Padova
Conservatory «Cesare Pollini»

PROCEEDINGS



PADOVA UNIVERSITY PRESS



Proceedings of
8th Sound and Music
Computing
Conference

Organized by



With the support of the Culture Programme of the European Union



Education and Culture DG

Institutional partners

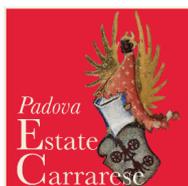
Corporate sponsors



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



**fondazione
ANTONVENETA**



IK MULTIMEDIA



viscount®
the art of sound



Proceedings of SMC 2011
8th Sound and Music Computing Conference
“Creativity rethinks science”

Edited by Serena Zanolla, Federico Avanzini, Sergio Canazza and Amalia de Götzen

Copyright

These proceedings, and all the papers included in it, are an open access publication distributed under the terms of the **Creative Commons Attribution 3.0 Unported License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and source are credited. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

The website of these proceedings is <http://www.padovauniversitypress.it/proceedings-of-the-smc-2011-8th-sound-and-music-computing-conference>

Padova University Press
Università degli Studi di Padova,
Via 8 febbraio 2, Padova
www.padovauniversitypress.it

Year of Publication: 2011

ISBN: 9788897385035

The SMC Conference is a privileged forum for international exchanges around the core interdisciplinary topics of Sound and Music Computing.

The SMC initiative is jointly supervised by the following European associations:

- **AFIM** (Association Française d’Informatique Musicale)
- **AIMI** (Associazione Italiana di Informatica Musicale)
- **DEGEM** (Deutsche Gesellschaft für Elektroakustische Musik)
- **HACI** (Hellenic Association of Music Informatics)
- **SITEMU** (Sociedad(e) Ibérica de Tecnología MUSical)

Chapter cover photo credit

A musical instrument in Music 5 on a IBM System 370 Model 158, the first computer used at the Center of Computational Sonology (CSC) to synthesize sound and voice (University of Padova, ca. 1975). The three screenshots show the general scheme of the CSC HW/SW system, a frequency modulation block scheme and a sound spectrum.

COMMITTEE

General Chair

Federico Avanzini (Università di Padova, Italy)

Technical Program Chairs

Giovanni De Poli (Università di Padova, Italy)

Davide Rocchesso (Università Iuav di Venezia, Italy)

Music Program Chairs

Nicola Bernardini (Conservatorio “Cesare Pollini” di Padova, Italy)

Alvise Vidolin (Conservatorio “Cesare Pollini” di Padova, Italy)

Paolo Zavagna (Conservatorio “Benedetto Marcello” di Venezia, Italy)

Summer School Chair

Sergio Canazza (Università di Padova, Italy)

Local Organizing Committee

Maria Bernini (Università di Padova, Italy)

Antonio Camporese (Università di Padova, Italy)

Amalia de Götzen (Conservatorio “Cesare Pollini” di Padova, Italy)

Michele Geronazzo (Università di Padova, Italy)

Enrico Marchetto (Università di Padova, Italy)

Nicola Montecchio (Università di Padova, Italy)

Aldo Orvieto (Conservatorio “Cesare Pollini” di Padova, Italy)

Antonio Rodà (Università di Padova, Italy)

Simone Spagnol (Università di Padova, Italy)

Serena Zanolla (Università di Udine, Italy)

Technical Program Committee

Alvaro Barbosa (Universidade Católica Portuguesa, Porto, Portugal)

Frédéric Bevilacqua (IRCAM, France)

Stefan Bilbao (University of Edinburgh, UK)

Simon Dixon (Queen Mary University of London, UK)

Cumhur Erkut (Aalto University, Finland)

Federico Fontana (Università di Udine, Italy)

Anastasia Georgaki (University of Athens, Greece)

Bruno Giordano (McGill University, Quebec, Canada)

Fabien Gouyon (INESC Porto, Portugal)

Kjetil Falkenberg Hansen (Royal Institute of Technology, Sweden)

Perfecto Herrera (Universitat Pompeu Fabra, Spain)

Haruhiro Katayose (Kwansei Gakuin University, Japan)

Marc Leman (Universiteit Gent, Belgium)

Yann Orlarey (GRAMME, Centre National de Creation Musical, France)

Stefania Serafin (Aalborg University Copenhagen, Denmark)

Tamara Smyth (Fraser University, BC, Canada)

Bob Sturm (Aalborg University Copenhagen, Denmark)

Gualtiero Volpe (Università di Genova, Italy)

Stefan Weinzierl (Technische Universität Berlin, Germany)

Gerhard Widmer (Johannes Kepler Universität, Austria)

REVIEWERS

Carlos Agon, Andreas Arzt, Anders Askenfelt, Balázs Bank, Karim Barkati, Stephen Barrass, Emmanouil Benetos, Samuel Benveniste, Edgar Berdahl, Dmitry Bogdanov, Roberto Bresin, Emilios Cambouropoulos, F. Amílcar Cardoso, Julien Castet, Chris Chafe, Regina Collecchia, Nick Collins, John Dack, Roger Dannenberg, Laurent Daudet, Matthew Davies, Giovanni De Sanctis, Stefano Delle Monache, Myriam Desainte-Catherine, Carlo Drioli, Hauke Egermann, Georg Essl, Anibal Ferreira, Jean-Julien Filatriau, Rosemary Fitzgerald, Arthur Flexer, Sebastian Flossmann, Dominique Fober, Karmen Franinovic, Chris Frauenberger, Anders Friberg, Daniel Gaertner, Steven Gelineck, Werner Goebel, Maarten Grachten, Massimo Grassi, Thomas Grill, Albert Gräf, Catherine Guastavino, Carlos Guedes, Brian Gygi, Huseyin Hacihabiboglu, Pierre Hanna, Christopher Harte, Mitsuyo Hashida, Piotr Holonowicz, Andre Holzapfel, Daniel Hug, Jordi Janer, Sergi Jorda, Pierre Jouvilot, Jian Kang, Stefan Kersten, Anssi Klapuri, Peter Knees, Ale Koerich, Panayiotis Kokoras, Thibault Langlois, Olivier Lartillot, Cyril Laurier, Sylvain Le Groux, Guillaume Lemaitre, Stéphane Letz, Tapio Lokki, Filipe Cunha Monteiro Lopes, Trond Lossius, Hanna Lukashevich, Esteban Maestre, Sylvain Marchand, Matija Marolt, Gonçalo Marques, Luis Gustavo Martins, Matthias Mauch, Davide Andrea Mauro, Stephen Mcadams, Lesley Mearns, Annamaria Mesaros, Luca Mion, Dirk Moelants, Tomoyasu Nakano, Bernhard Niedermayer, Katy Noland, Kjartan Olafsson, João Lobato Oliveira, Nicola Orio, Rui Pedro Paiva, Jyri Pakarinen, Stefano Papetti, Sandra Pauletto, Henri Penttinen, Alfonso Perez, Karin Petrini, David Pirrò, Mark Plumbley, Pietro Polotti, Pedro J. Ponce De León, Laurent Pottier, Carlos Pérez-Sancho, Rafael Ramirez, Matthias Rath, Michal Rinott, Gerard Roma, Justin Salamon, Christopher Salter, Augusto Sarti, Markus Schedl, Erwin Schoonderwaldt, Diemo Schwarz, Veronique Sebastien, Joan Serrà, Klaus Seyerlehner, Stephen Sinclair, Malcolm Slaney, Julius Smith, Alois Sontacchi, Mohamed Sordo, Marc Sosnick, Simone Spagnol, Tapio Takala, Luis Teixeira, Mari Tervaniemi, Barbara Tillmann, George Tzanetakis, Vesa Valimaki, Leon Van Noorden, Giovanna Varni, Domenico Vicinanza, Marcelo Wanderley, Craig Webb, Tillman Weyde, Kazuyoshi Yoshii, Stefano Zambon, Massimiliano Zanoni.

TABLE OF CONTENTS

| | |
|---|----|
| Introduction..... | 1 |
| KEYNOTE ADDRESS | |
| Time Is of the Essence: Creativity, Symmetry, and Counterintuitive Solutions..... <i>Roberto Casati.</i> | 3 |
| ORAL PRESENTATIONS | |
| Oral Session 1: COMPUTATIONAL MUSICOLOGY | |
| [OS1 - 1] The Plurality of Melodic Similarity..... <i>Alan Marsden.</i> | 5 |
| [OS1 - 2] Real-Time Unsupervised Music Structural Segmentation using Dynamic Descriptors..... <i>André S. Pires and Marcelo Queiroz.</i> | 11 |
| [OS1 - 3] Multiple-Instrument Polyphonic Music Transcription using a Convolutional Probabilistic Model..... <i>Emmanouil Benetos and Simon Dixon.</i> | 19 |
| [OS1 - 4] Automatically Detecting Key Modulations in J.S. Bach Chorale Recordings..... <i>Lesley Mearns, Emmanouil Benetos, and Simon Dixon.</i> | 25 |
| Oral Session 2: MUSICAL HERITAGE | |
| [OS2 - 1] A Survey of Raaga Recognition Techniques and Improvements to the State-of-the-Art..... <i>Koduri Gopala Krishna, Sankalp Gulati, and Preeti Rao.</i> | 33 |
| [OS2 - 2] Version Detection for Historical Musical Automata..... <i>Bernhard Niedermayer, Gerhard Widmer, and Christoph Reuter.</i> | 41 |
| [OS2 - 3] Demetrio Stratos Rethinks Vocal Techniques: a Historical Investigation at ISTC in Padova..... <i>Elena Ceolin, Graziano Tisato, and Laura Zattra.</i> | 48 |
| Oral Session 3: AUGMENTED LEARNING | |
| [OS3 - 1] SoundScape: a Music Composition Environment Designed to Facilitate Collaborative Creativity in the Classroom..... <i>Sylvia Truman.</i> | 56 |
| [OS3 - 2] When Sound Teaches..... <i>Serena Zanolla, Antonio Rodà, Filippo Romano, Francesco Scattolin, Sergio Canazza, and Gian Luca Foresti.</i> | 64 |
| [OS3 - 3] Ljudskrapan/The Soundscraper: Sound Exploration for Children with Complex Needs, Accommodating Hearing Aids and Cochlear Implants..... <i>Kjetil Falkenberg Hansen, Christina Dravins, and Roberto Bresin.</i> | 70 |
| [OS3 - 4] C. Elegans Meets Data Sonification: Can We Hear its Elegant Movement? <i>Hiroko Terasawa, Yuta Takahashi, Keiko Hirota, Takayuki Hamano, Takeshi Yamada, Akiyoshi Fukamizu, and Shoji Makino.</i> | 77 |
| Oral Session 4: SOUND MODELING | |
| [OS4 - 1] Using Physical Models is <i>Necessary</i> to Guarantee Stable Analog Haptic Feedback for any User and Haptic Device..... <i>Edgar Berdahl, Jean-Loup Florens, and Claude Cadoz.</i> | 83 |

| | |
|---|-----|
| [OS4 - 2] Physical Modeling Meets Machine Learning: Teaching Bow Control to a Virtual Violinist..... | 91 |
| <i>Graham Percival, Nicholas Bailey, and George Tzanetakis.</i> | |
| [OS4 - 3] Parametric Trombone Synthesis by Coupling Dynamic Lip Valve and Instrument Models..... | 99 |
| <i>Tamara Smyth and Frederick Scott.</i> | |
| [OS4 - 4] Distance Mapping for Corpus-Based Concatenative Synthesis..... | 105 |
| <i>Diemo Schwarz.</i> | |
| Oral Session 5: EMOTIONS AND EXPRESSION IN MUSIC | |
| [OS5 – 1] Emotional Response to Major Mode Musical Pieces: Score-Dependent Perceptual and Acoustic Analysis..... | 109 |
| <i>Sergio Canazza, Giovanni De Poli, and Antonio Rodà.</i> | |
| [OS5 - 2] Explaining Musical Expression as a Mixture of Basis Functions..... | 115 |
| <i>Maarten Grachten and Gerhard Widmer.</i> | |
| [OS5 - 3] A Comparison of Perceptual Ratings and Computed Audio Features..... | 122 |
| <i>Anders Friberg and Anton Hedblad.</i> | |
| [OS5 - 4] Investigation of the Relationships between Audio Features and Induced Emotions in Contemporary Western Music..... | 128 |
| <i>Konstantinos Trochidis, Charles Delbé, and Emmanuel Bigand.</i> | |
| Oral Session 6: CREATIVITY | |
| [OS6 - 1] Humanities, Art and Science in the Context of Interactive Sonic Systems – Some Considerations on a Cumbersome Relationship..... | 133 |
| <i>Pietro Polotti.</i> | |
| [OS6 - 2] Exploring the Design Space: Prototyping “The Throat V3” for the Elephant Man Opera..... | 141 |
| <i>Ludvig Elblaus, Kjetil Falkenberg Hansen, and Carl Unander-Scharin.</i> | |
| [OS6 - 3] Marco Stroppa’s Compositional Process and Scientific Knowledge between 1980-1991..... | 148 |
| <i>Vincent Tiffon and Noémie Sprenger-Ohana.</i> | |
| [OS6 - 4] Limits of Control..... | 155 |
| <i>Hanns Holger Rutz.</i> | |
| Oral Session 7: MUSIC AUTOMATION | |
| [OS7 - 1] Generating Musical Accompaniment through Functional Scaffolding..... | 161 |
| <i>Amy K. Hoover, Paul A. Szerlip, and Kenneth O. Stanley.</i> | |
| [OS7 - 2] A Rule-Based Generative Music System Controlled by Desired Valence and Arousal..... | 169 |
| <i>Isaac Wallis, Todd Ingalls, Ellen Campana, and Janel Goodman.</i> | |
| [OS7 - 3] Automatic Multi-Track Mixing using Linear Dynamical Systems..... | 177 |
| <i>Jeffrey Scott, Matthew Prockup, Erik M. Schmidt, and Youngmoo E. Kim.</i> | |
| [OS7 - 4] DanceReProducer: an Automatic Mashup Music Video Generation System by Reusing Dance Video Clips on the Web..... | 183 |
| <i>Tomoyasu Nakano, Sora Murofushi, Masataka Goto, and Shigeo Morishima.</i> | |
| Oral Session 8: ENVIRONMENTS FOR SOUND/MUSIC PROCESSING | |
| [OS8 - 1] On the Creative Use of Score Following and its Impact on Research..... | 190 |
| <i>Arshia Cont.</i> | |
| [OS8 - 2] Ensemble: Implementing a Musical Multiagent System Framework..... | 198 |
| <i>Leandro Ferrari Thomaz and Marcelo Queiroz.</i> | |
| [OS8 - 3] Audio Physical Computing..... | 206 |
| <i>Andrea Valle.</i> | |

| | |
|--|-----|
| [OS8 - 4] The Vowel Worm: Real-Time Mapping and Visualisation of Sung Vowels in Music..... | 214 |
| <i>Harald Frostel, Andreas Arzt, and Gerhard Widmer.</i> | |

Oral Session 9: INTERACTION WITH SOUND AND MUSIC

| | |
|--|-----|
| [OS9 - 1] Sonic Gestures as Input in Human-Computer Interaction: towards a Systematic Approach..... | 220 |
| <i>Antti Jylhä.</i> | |
| [OS9 - 2] Improving Performers' Musicality through Live Interaction with Haptic Feedback: a Case Study..... | 227 |
| <i>Tychonas Michailidis and Jamie Bullock.</i> | |
| [OS9 - 3] Where Do You Want Your Ears? Comparing Performance Quality as a Function of Listening Position in a Virtual Jazz Band..... | 233 |
| <i>Adriana Olmos, Paul Rushka, Doyuen Ko, Gordon Foote, Wieslaw Woszczyk, and Jeremy R. Cooperstock.</i> | |
| [OS9 - 4] The EyeHarp: an Eye-Tracking-Based Musical Instrument..... | 239 |
| <i>Zacharias Vamvakousis and Rafael Ramirez.</i> | |

POSTER PRESENTATIONS

Poster Session 1

| | |
|--|-----|
| [PS1 - 1] Improving Tempo-Sensitive and Tempo-Robust Descriptors for Rhythmic Similarity..... | 247 |
| <i>Andre Holzapfel, Arthur Flexer, and Gerhard Widmer.</i> | |
| [PS1 - 2] Gestural Control of Real-Time Speech Synthesis in Luna Park..... | 253 |
| <i>Grégory Beller.</i> | |
| [PS1 - 3] An Interactive Surface Realisation of Henri Pousseur's 'Scambi' | 259 |
| <i>Robin Fencott and John Dack.</i> | |
| [PS1 - 4] Spatio-Temporal Unfolding of Sound Sequences..... | 265 |
| <i>Davide Rocchesso and Stefano Delle Monache.</i> | |
| [PS1 - 5] An Exploration on the Influence of Vibrotactile Cues During Digital Piano Playing..... | 273 |
| <i>Federico Fontana, Marco Civolani, Stefano Papetti, Valentina del Bello, and Balázs Bank.</i> | |
| [PS1 - 6] On Computing Morphological Similarity of Audio Signals..... | 279 |
| <i>Martin Gasser, Arthur Flexer, and Thomas Grill.</i> | |
| [PS1 - 7] Sound Spatialization Control by means of Acoustic Source Localization System..... | 284 |
| <i>Daniele Salvati, Sergio Canazza, and Antonio Rodà.</i> | |
| [PS1- 8] An Analog I/O Interface Board for Audio Arduino Open Sound Card System..... | 290 |
| <i>Smilen Dimitrov and Stefania Serafin.</i> | |
| [PS1 - 9] Designing an Expressive Virtual Percussion Instrument..... | 298 |
| <i>Brian Dolhansky, Andrew McPherson, and Youngmoo E. Kim.</i> | |
| [PS1 - 10] Active Preservation of Electrophone Musical Instruments. The Case of the "Liettizzatore" of "Studio di Fonologia Musicale" (Rai, Milano)..... | 304 |
| <i>Sergio Canazza, Federico Avanzini, Maria Maddalena Novati, and Antonio Rodà.</i> | |
| [PS1 - 11] Design and Applications of a Multi-Touch Musical Keyboard..... | 310 |
| <i>Andrew McPherson and Youngmoo Kim.</i> | |
| [PS1 - 12] Improved Frequency Estimation in Sinusoidal Models through Iterative Linear Programming Schemes..... | 317 |
| <i>Vighnesh Leonardo Shiv.</i> | |
| [PS1 - 13] Personality and Computer Music..... | 323 |
| <i>Sandra Garrido, Emery Schubert, Gunter Kreutz, and Andrea Halpern.</i> | |
| [PS1 - 14] Auditory Feedback in a Multimodal Balancing Task: Walking on a Virtual Plank..... | 329 |
| <i>Stefania Serafin, Luca Turchet, and Rolf Nordahl.</i> | |

| | |
|---|-----|
| [PS1 - 15] Analysis of Social Interaction in Music Performance with Score-Independent Audio Features..... | 335 |
| <i>Gualtiero Volpe, Giovanna Varni, Barbara Mazzarino, Silvia Pisano, and Antonio Camurri.</i> | |

Poster Session 2

| | |
|---|-----|
| [PS2 - 1] Applications of Synchronization in Sound Synthesis..... | 340 |
| <i>Martin Neukom.</i> | |
| [PS2 - 2] Melody Harmonization in Evolutionary Music using Multiobjective Genetic Algorithms..... | 346 |
| <i>Alan R. R. Freitas and Frederico G. Guimarães.</i> | |
| [PS2 - 3] An Adaptive Classification Algorithm for Semiotic Musical Gestures..... | 354 |
| <i>Nicholas Gillian, R. Benjamin Knapp, and Sile O'Modhrain.</i> | |
| [PS2 - 4] An Interactive Music Composition System based on Autonomous Maintenance of Musical Consistency..... | 362 |
| <i>Tetsuro Kitahara, Satoru Fukayama, Shigeki Sagayama, Haruhiro Katayose, and Noriko Nagata.</i> | |
| [PS2 - 5] A Learning Interface for Novice Guitar Players using Vibrotactile Stimulation..... | 368 |
| <i>Marcello Giordano and Marcelo M. Wanderley.</i> | |
| [PS2 - 6] Functional Signal Processing with Pure and Faust using the LLVM Toolkit..... | 375 |
| <i>Albert Gräf.</i> | |
| [PS2 - 7] RaPScoM - A Framework for Rapid Prototyping of Semantically Enhanced Score Music..... | 381 |
| <i>Julian Rubisch, Jakob Doppler, and Hannes Raffaseder.</i> | |
| [PS2 - 8] Foley Sounds Vs. Real Sounds..... | 388 |
| <i>Stefano Trento and Amalia de Götzen.</i> | |
| [PS2 - 9] Robotic Piano Player Making Pianos Talk..... | 394 |
| <i>Winfried Ritsch.</i> | |
| [PS2 - 10] Sound Spheres: a Design Study of the Articulatory of a Non-Contact Finger Tracking Virtual Musical Instrument..... | 400 |
| <i>Craig Hughes, Michel Wermelinger, and Simon Holland.</i> | |
| [PS2 - 11] Prioritized Contig Combining to Segregate Voices in Polyphonic Music..... | 408 |
| <i>Asako Ishigaki, Masaki Matsubara, and Hiroaki Saito.</i> | |
| [PS2 - 12] Rencon Workshop 2011 (SMC-Rencon): Performance Rendering Contest for Computer Systems..... | 415 |
| <i>Mitsuyo Hashida, Keiji Hirata, and Haruhiro Katayose.</i> | |
| [PS2 - 13] Comparing Inertial and Optical Mocap Technologies for Synthesis Control..... | 421 |
| <i>Ståle A. Skogstad, Kristian Nymoén, and Mats Høvin.</i> | |
| [PS2 - 14] A Toolbox for Storing and Streaming Music-Related Data..... | 427 |
| <i>Kristian Nymoén and Alexander Refsum Jensenius.</i> | |
| [PS2 - 15] Automatic Creation of Mood Playlists in the Thayer Plane: a Methodology and a Comparative Study..... | 431 |
| <i>Renato Panda and Rui Pedro Paiva.</i> | |

Poster Session 3

| | |
|---|-----|
| [PS3 - 1] Towards a Personalized Technical Ear Training Program: an Investigation of the Effect of Adaptive Feedback..... | 439 |
| <i>Teruaki Kaniwa, Sungyoung Kim, Hiroko Terasawa, Masahiro Ikeda, Takeshi Yamada, and Shoji Makino.</i> | |
| [PS3 - 2] Extraction of Sound Localization Cue utilizing Pitch Cue for Modelling Auditory System..... | 444 |
| <i>Takatoshi Okuno, Thomas M. McGinnity, and Liam P. Maguire.</i> | |
| [PS3 - 3] Support for Learning Synthesiser Programming..... | 450 |
| <i>Mateusz Dykiert and Nicolas Gold.</i> | |

| | |
|--|-----|
| [PS3 - 4] Leech: Bittorrent and Music Piracy Sonification..... <i>Curtis McKinney and Alain Renaud.</i> | 457 |
| [PS3 - 5] Sonik Spring..... <i>Tomás Henriques.</i> | 464 |
| [PS3 - 6] Isomorphic Tessellations for Musical Keyboards..... <i>Steven Maupin, David Gerhard, and Brett Park.</i> | 471 |
| [PS3 - 7] Improving the Efficiency of Open Sound Control with Compressed Address Strings..... <i>Jari Kleimola and Patrick J. McGlynn.</i> | 479 |
| [PS3 - 8] Dynamic Intermediate Models for Audiographic Synthesis..... <i>Vincent Goudard, Hugues Genevois, Émilien Ghomi, and Boris Doval.</i> | 486 |
| [PS3 - 9] From Snow [to Space to Movement] to Sound..... <i>Alexandros Kontogeorgakopoulos, Olivia Kotsifa, and Matthias Erichsen.</i> | 492 |
| [PS3 - 10] A Bayesian Approach to Drum Tracking..... <i>Andrew N. Robertson.</i> | 498 |
| [PS3 - 11] Towards a Generative Electronica: Human-Informed Machine Transcription and Analysis in MaxMSP <i>Arne Eigenfeldt and Philippe Pasquier.</i> | 504 |
| [PS3 - 12] The Closure-Based Cueing Model: Cognitively Inspired Learning and Generation of Musical Sequences..... <i>James Maxwell, Philippe Pasquier, and Arne Eigenfeldt.</i> | 510 |
| [PS3 – 13] Evaluation of Sensor Technologies for the Rulers, a Kalimba-Like Digital Musical Instrument..... <i>Carolina Brum Medeiros and Marcelo M. Wanderley.</i> | 518 |
| [PS3 14] BeatLED - The Social Gaming Partyshirt..... <i>Tom De Nies, Thomas Vervust, Michiel Demey, Rik Van de Walle, Jan Vanfleteren, and Marc Leman.</i> | 526 |
| Author Index..... | 533 |

INTRODUCTION

Dear fellow SMC Researchers,

I am glad to welcome you to the the 8th edition of the Sound and Music Computing Conference in Padova, and to present the Book of Proceedings of the Technical Program.

When in 2009 we applied for hosting SMC2011, we were driven by two main motivations. First, we wanted to contribute to the growth of our Conference. SMC was born in 2004 from a joint effort by the Italian and French Music Informatics Associations and is still a young event. In a few years its international dimension has grown considerably, thanks to the involvement of other national Music Informatics Associations, and through the establishment of a permanent Steering Committee. The quality of scientific contributions has also improved steadily, thanks to the efforts of previous Organizers.

Over the years, SMC has defined its own identity: a compact and selective conference, which aims at representing the whole spectrum of Sound and Music Computing research, looks for participation especially from young and emerging researchers, believes in interdisciplinary exchanges, grants open access to its contents.

One of the milestones in the evolution of the SMC format has been the convergence between the Conference and the Summer School. The SMC Summer School was born in 2005 as an outcome of the EU Project S2S² (Sound-to Sense, Sense-to-Sound), which also originated the SMC Research Roadmap and the smcnetwork.org portal. Starting with the 2009 edition in Porto, the Summer School takes place just before the Conference, with about 20 PhD students and young researchers attending lectures and hands-on projects. Following the decision taken by the SMC Steering Committee at the beginning of 2011, this is now the official format of SMC.

This year the Summer School focus is around the topic of interaction and embodiment in sound and music. Complementing the Summer School, and marking the beginning of the Conference, the “SMC-Rencon” Workshop is a special event of this year's SMC. Rencon is a contest for computer systems generating expressive musical performances, which are subjectively evaluated by the attending audience. Evaluation by contests, where various systems gather and compete against one another, stimulates scientific progress. Rencon was started in 2002 from this perspective: each edition hosts a competition where systems have to generate performances of newly created musical pieces on site. With this year's edition, Rencon takes place for the second time in Europe, and for the first time in Italy.

The Technical Program of the Conference itself comprises 79 contributions, selected among 136 submissions. We did our best to set up a rigorous and fair peer-review process: thanks to the work of 20 members of the Technical Program Committee, and about 130 reviewers, each submission received no less than 3 independent reviews. Accepted papers are published under a Creative Commons license and are available on smcnetwork.org. Scientific (and music) submissions were sent from 37 countries and 5 continents, which shows that our conference is no longer a EU-centered event and is becoming a worldwide reference.

The general theme of SMC2011 is “Creativity Rethinks Science”. Creativity is at the core of progress and innovation mechanisms, and SMC is a discipline where several existing unconventional creative environments are found, where research and art already collaborate in a productive way injecting new ideas and concepts in both fields. These provide a fertile ground to analyze and try to understand artistic thinking as a driver of innovation, the relationship between artistic and scientific methodologies, and the processes that lead to successful artistic and/or scientific results.

We believe that this general theme applies transversally to all the conference topics. We are very pleased to have philosopher Roberto Casati as our Keynote Speaker. Being an outsider of our community, he will provide the audience with inspiring ideas, addressing this theme from a different angle.

I started this Introduction by mentioning two motivations in our candidacy to SMC2011. The second one is that we wanted to celebrate SaMPL, the Sound and Music Processing Lab created by the Conservatory and the University in 2009. Electronic and computer music research in Padova has a longstanding tradition that dates back to the 1960s, with the pioneering research by Prof. Giovanni Battista Debiasi on electronic organs, and with the electronic music school founded by Teresa Rampazzi. This fertile ground led in 1979 to the establishment of the Center of Computational Sonology (CSC). Thirty years later, SaMPL aims at becoming part of this history.

Federico Avanzini
General Chair SMC2011



KEYNOTE ADDRESS

TIME IS OF THE ESSENCE: CREATIVITY, SYMMETRY, AND COUNTERINTUITIVE SOLUTIONS

Roberto Casati

Institut Jean Nicod, Ecole Normale Suprieure, 29 rue d'Ulm, 75005 Paris, France
casati@ehess.fr

ABSTRACT

By any measure, biological evolution is astonishingly creative, but can we use its mechanism as a model for understanding cultural creativity? Creative and artistic processes are meant to generate solutions, and the success of these solutions is highly contextually constrained. According to mainstream theories, creative engines have two abstract components: a solution generator and a solution selector. Biological evolution puts all the constraints on the selection mechanism: generation is random, waste is immense, and luckily a vast amount of time provides endless biological money. Cultural evolution cannot afford to be that generous, whereby it must incorporate constraints in the generation itself. I shall discuss a number of cases in which creativity is achieved by exploring parametric solutions that are symmetric to the ones that appear intuitive.

Biographical notes

Roberto Casati (Milan, Italy, 1961) is a tenured senior researcher with the French Centre National de la Recherche Scientifique (CNRS-EHESS-ENS). Based in Paris, France, he has worked on various research projects in philosophy and the cognitive sciences, and has taught at several universities, among which the State University of New York at Buffalo, the University IUAV-Venice, the University of Turin, and Columbia University. He is the recipient of various prizes and of grants from several institutions, including CNRS, MENRT, and the EU Commission. His books, some of which have been translated in many languages, include *Holes and other superficialities* (MIT Press 1994, with Achille Varzi), *La philosophie du son* (Philosophy of Sound, with J. Dokic, 1994), *The Shadow Club* (Knopf 2002). He is currently working, with V. Girotto, at a book on *Creative Solutions*. <http://www.shadowes.org>



ORAL PRESENTATIONS

THE PLURALITY OF MELODIC SIMILARITY

Alan Marsden

Lancaster Institute for the Contemporary Arts

Lancaster University, UK

A.Marsden@lancaster.ac.uk

ABSTRACT

Melodic similarity is a much-researched topic. While there are some common paradigms and methods, there is no single emerging model. The different means by which melodic similarity has been studied are briefly surveyed and contrasts drawn between them which lead to important differences in the light of the finding that similarity is dependent on context. Models of melodic similarity based on reduction are given particular scrutiny, and the existence of multiple possible reductions proposed as a natural basis for a lack of triangle inequality. It is finally proposed that, in some situations at least, similarity is deliberately sought by maximising the similarity of interpretations. Thus melodic similarity is found to be plural on two counts (differing contexts and multiple interpretations) and furthermore to be an essentially *creative* concept. There are therefore grounds for turning research on melodic similarity on its head and using the concept as a means for studying reduction and in musical creative contexts.

1. WHAT IS MELODIC SIMILARITY?

A common theme of music-computing research in the last couple of decades has been measurement of melodic similarity. Much of this research has been in the context of query systems, with the aim of finding a way of organising and searching a database of music so as to retrieve melodies similar to a given query. The idea has been used also as a basis for segmentation, for music analysis and for research on music cognition. This growing body of research, however, shows little agreement about what melodic similarity depends on, how to measure it, or even what it really is.

1.1 Seeking a similarity metric

The simple observation that some melodies are similar while others are different, and that the similarity can be closer or more distant, seems to have led many to believe that melodic similarity is a metric space. Formally, a metric is a function from two objects (here melodies) to a quantity (distance, difference) with the following properties [1, p.38]:

Copyright: © 2011 Alan Marsden. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

- (a) non-negativity—the distance between two objects is never less than 0;
- (b) self-identity—the distance is 0 if and only if the objects are the same;
- (c) symmetry—the distance from a to b is the same as from b to a ; and
- (d) triangle inequality—the distance from a to c is never greater than the distance from a to b plus the distance from b to c .

The first two properties are rarely open to question in the case of differences between melodies, but it is not self-evident that (c) and (d) should be true. Symmetry is most obviously questioned when a short melody is compared to a long one. (A ring tone can be similar to a symphony, at least in the sense that it brings the symphony to mind when we hear it, but the symphony is unlikely to be considered similar to the ring tone.) This situation, however, rarely arises in the contexts considered below. Thus, while symmetry in melodic similarity is in need of thorough investigation, it will be assumed to apply in the remainder of this paper.

The property most commonly questioned is triangle inequality, and the common grounds for this are that melody a might be similar to melody b by virtue of property or component x , while melody b might be similar to melody c by virtue of a different property or component y . In such a situation there is no reason to expect the dissimilarity between a and c to be limited. Despite such easily imagined counter-examples, those who use systems of measurement with the property of triangle inequality have not reported failure to match human judgements of melodic similarity on the grounds that those judgements do not exhibit triangle inequality. Indeed it is not uncommon to adapt a measure precisely so that it has the property of triangle inequality (for example the development of Proportional Transportation Distance [2] from Earth Mover's Distance) with the objective of facilitating the organisation and searching of a database. (Meanwhile, others have taken the alternative path of investigating means for organising and searching databases without the need of triangle inequality [3].)

1.2 Contrasting empirical bases

Most studies have grounded their work on some kind of empirical basis, some raw 'truth' that certain melodies are similar and others are not. When we look at the detail, however, we find that very different paradigms have been

used, firstly in the source of that 'truth' and secondly in the kind of relationship tested between melodies.

Many studies ask experimental subjects, often experts, to judge the similarity between pairs of melodies or extracts of melodies on a rating scale [4–8]. This has the advantage of directly generating measures of difference which will almost certainly have the first three properties of a metric. A rating of 0 or below is not an option; subjects are not asked to compare a melody to itself; and the set-up usually discourages asymmetric judgements. There is no guarantee, however, of triangle inequality. One objection to experimental procedures like this is that they are not realistic: musicians are rarely (if ever) in a situation when they have to match the similarity between melodies to a number. Such direct measurement was avoided in another study which also used expert judgement but subjects were asked to rank a set of melodies by their similarity to a reference melody rather than to simply compare pairs of melodies [9]. A measure of difference can be derived from the relative positions of melodies in the rankings. A potential disadvantage of the method, however, is that experts' judgment of the similarity between a pair of melodies is much more likely to be influenced by the context of the other melodies they are asked to rank simultaneously. This is avoided in approaches where subjects simply compare three melodies (identifying the pair which is most alike and the pair which is least alike) [10, 11]. Indeed, this approach is the one which places the least burden on experimental subjects, and it appears to have been successful for non-expert subjects, unlike the paradigms mentioned above. On the other hand, deriving metric data from these observations requires a method such as multi-dimensional scaling, and a large quantity of observations.

Other studies have avoided direct judgment of similarity, whether by experts or naive listeners. Some have depended on categorisation of melodies either from existing musicological studies [12, 7] or on the basis of geographical origin [13]. In these cases a useful metric cannot be derived from the empirical data, since distances between melodies are all either 0 or 1 according to whether or not the melodies belong to the same category. However, the data can still be used to verify a computational metric on the grounds that the computed distance for melodies within a category should be less than the distance between melodies from different categories.

Yet other studies have attempted to judge similarity on the basis of some real musical activity. Studies aimed at producing metrics for use in query-by-humming systems have been based on asking subjects to sing a known melody [14, 15]. The subjects make mistakes, so the resulting melody is not the same as the original, but it is assumed to be more similar to the original than to other melodies. Subjects can also be asked to deliberately vary a melody [16], and once again the variations are assumed to be more similar to the original than to other melodies. (Others used a related approach of introducing artificial variations into melodies, but this was usually to generate test

materials which were then subject to expert judgement of similarity.)

1.3 Similarity and cognition

Do all these paradigms study the same thing? Certainly there are other musical phenomena whose underlying models are robust under different experimental paradigms (models of tonal perception via pitch-frequency profiles are one example), and these suggest stable underlying cognitive functions. The data on melodic similarity has been shown to be relatively consistent from one expert to another and from one occasion to another under the same paradigm, but I am not aware of evidence of consistency between different paradigms. Indeed, there is clear evidence for what one might expect from other aspects of human behaviour: that judgements of melodic similarity are dependent on context. Müllensiefen and Frieler have demonstrated that a different model is required to account for similarity judgements which use the same paradigm but in which the set of melodies to be compared is different [7].

In fact, the contexts in these various experiments have been very different. The nature of melodic materials has varied widely, and crucially the instructions and information given to the subjects have also varied. Sometimes subjects have been given no further instruction than to rate the similarity between two melodies. On other occasions they have been given guidance such as to imagine that the comparison melody is a student's attempt to reproduce a teacher's melody and to think of the similarity rating as a mark [7]. (Note that in this case the similarity judgement can no longer be assumed to be symmetric.) Sometimes subjects' attention has been drawn to particular aspects of the melody, for example by being told in advance that the experiment was concerned with contour [8].

The differences in paradigm also introduce significant issues. If data is derived from real musical behaviours which do not involve explicit similarity judgements, we can only assume that similarity is a governing factor; if data is not derived from real musical behaviours we cannot be certain that it has any real musical relevance. Even in the cases based on explicit expert judgements of similarity, there are important differences. As stated above, we cannot be certain that judgement of melodic similarity has the property of triangle inequality. Even if it does not, subjects can give answers with confidence when asked to rate the similarity between two melodies, or even to judge the most similar and least similar pairs in a triple. However, in a ranking task such as used in [9] the subjects might be in a position of having to balance competing similarity judgements, depending on how they interpret the instructions. If they consider their task to be simply to ensure that the melody ranked x is no less similar to the reference than the melody ranked $x + 1$, no competing rankings can arise. If, however, they also believe that a ranking implies that the melody ranked $x + 2$ is less similar to the one ranked x than the one ranked $x + 1$, then in

the absence of triangle inequality, a subject might find it impossible to find a ranking which meets both criteria: melodies a , b and c might have decreasing similarity to the reference, and so be ranked x , $x + 1$ and $x + 2$, but c might be more similar to a than b , implying instead the ranking x , $x + 2$ and $x + 1$.

It is not safe, therefore, to assume that these studies investigate the same phenomenon of melodic similarity. Until there is evidence that data produced under these various paradigms is compatible, and in particular evidence that melodic similarity does exhibit triangle inequality, it is probably better to consider melodic similarity to be a family of possibly related phenomena.

2. MEASURING SIMILARITY

As mentioned above, different approaches to measuring melodic similarity have arisen from different objectives. A common one has been the retrieval of melodies from a database, but there are others also. Some seek to use measures of similarity as an aid in ethnomusicological studies, for example to find variants of a folk song, or to trace the provenance of a song. Others aim to use it as a tool in music analysis. In each case, the kinds of differences one is likely to find in melodies are likely to vary, and an approach founded on behaviours should take these into account. For example, in a query-by-humming system, a similarity metric should ideally be based on the kinds of errors which singers make when trying to recall and reproduce a melody. Similarly, similarity in folk songs should take into account the kinds of changes commonly introduced in oral traditions (either accidentally or deliberately), which might vary from one culture to another. In music analysis, one is generally concerned not with mistakes or accidental changes, but with deliberate and crafted variations of musical materials. In the remainder of this paper, I will concentrate on similarity in this context.

2.1 Similarity based on reduction

It is common to regard melodies as having an underlying structure, and to consider melodies sharing the same structure to be similar (at least in one sense) even if their surface details are quite different. To account for this kind of similarity, studies have been based on comparing melodies not note-by-note, but on the basis of a reduction of the melodies (generally in a tree structure) which progressively removes decorative notes until only the main outline of the melody is left [16–19].

Rizo and colleagues [16, 17] derive the reduction of a melody by selecting one of the notes occurring in each span based on a small number of rules. The spans are determined by the metre, so that, in 4/4 for example, there is a span for each bar, at the next level down two spans for the minims (half notes), then four spans for the crotchets (quarter notes), etc., halving each span at the level above. There are also higher-level spans which group bars into pairs, etc. The result is a tree structure in

which each node corresponds to a specific time span, and the rhythm of the melody is completely defined by the tree structure. The reduction is built bottom-up by

- (a) always selecting a note in preference to a rest,
- (b) selecting a harmonic note in preference to a non-harmonic one, and
- (c) selecting the note at the head of the span if both are harmonic.

A harmonic analysis of the melody must be generated before reduction, and this is currently done by hand. A measure of similarity based on the tree edit distance between the reductions of melodies was compared with edit distance on the melodic surfaces alone. The reduction-based similarity measure proved to perform better at distinguishing variations of a melody from unrelated melodies [16].

The approach of Orio & Rodà [18] is similar, in that it generates a tree based on the metrical structure, and notes are selected within each span partly on the basis of a harmonic analysis. The selection, however, is based on a more complex set of weights using the relation of the note to the underlying harmony (fifth, third or root), the function of that harmony, and the position in the metre. Furthermore, similarity between melodies is not based on the edit distance between trees. Melodies are segmented (using pre-existing segmentation schemes) and the segmentation propagated to higher levels of the tree. The resulting melodic segments, expressed as interval patterns, are placed in a directed acyclic graph (DAG) in which parent-child relations between segments copy those relations in the reductions. The difference between two segments is then measured by the minimum path length between the segments in the DAG, and the difference between two melodies is the average difference between their component segments. This method was not tested against other measures of melodic similarity.

The reductions produced by my own system [19, 20] are intended to more closely mimic the reductions of Schenkerian analysis. Furthermore, they are based not just on melodies but on a full musical texture (generally extracts from piano pieces). The reduction process is therefore considerably more complex than those outlined above. In particular, the reduction tree does not necessarily follow the metrical structure (as indeed it does not in many Schenkerian analyses), and no prior harmonic analysis is necessary (though specification of the key and metre is). While early results matched actual analyses to a promising degree [20], an attempt to use the same system of reduction for demonstrating the similarity underlying themes and variations produced less promising results [19]. Matching themes and variations via reductions proved no better than matching on the basis of the surfaces alone.

2.2 Multiple reductions

One possible reason for the disappointing results in [19] might have been poor reductions. I did not check each reduction for accuracy (after all, in the absence of prior

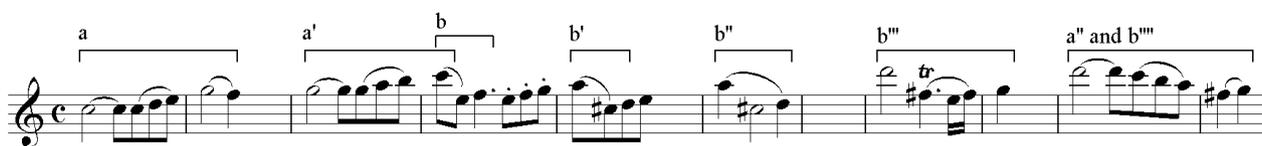


Figure 1. Extracts from Mozart's string quartet in C major, K. 465, first movement.

analyses, there is no test of accuracy other than expert analytical judgement), but I did note a number of cases where the reduction of a theme seemed incorrect. One important finding from the research on computational Schenkerian analysis is that a very large number of reductions is possible on the basis of the 'rules' inferred from writings on Schenkerian analysis alone [20]. Indeed, music analysts commonly recognise that alternative analyses of the same piece of music are possible and valid. If multiple reductions are possible, how should a similarity-measurement procedure based on reduction select which reduction to use?

It is instructive in this context to compare reduction-based similarity with edit distance, or more specifically Levenshtein distance. This measures the difference between two sequences in terms of the number of deletions, insertions and substitutions required to transform one sequence into another. Since reduction depends on selecting one of the notes of a pair (in most cases; Rizo et al. allow selection from a triple if warranted by the metre) each reduction step can be considered as equivalent to a deletion. Note that an insertion in one sequence is equivalent to a deletion in the other, and a substitution is equivalent to a deletion in both sequences. Thus if the difference between two melodies is measured as the minimum of the sum of reduction steps necessary to arrive at the same reduction for both melodies minus the number of reductions which take place in equivalent places (to account for substitutions), the difference is equivalent to the Levenshtein distance between the two melodies. (There is thus a strong correspondence between Levenshtein distance and the metric used by Orio & Rodà.) However, this assumes that the two melodies can be freely aligned in the way which allows for the minimum number of deletions, insertions and substitutions. Reduction, on the other hand, in all of the cases examined, is constrained by the rhythm, metre and other characteristics of the melody. Thus computing difference by reduction can be seen as similar to computing the Levenshtein distance with constraints on how the melodies may be aligned. I say 'similar' because it is difficult to see how the constraints of harmony and melody could be applied without actually performing the reduction. However, it might be a useful approach (especially in the light of the considerable computational complexity of reduction as performed in [20]) to use alignment constrained by metre as a means of guiding reduction.

3. SIMILARITY AND CREATIVITY

3.1 Finding similarity

Reduction is not the only approach to similarity which depends on a step which is potentially subject to multiple interpretations. A number of similarity-measurement systems depend on segmentation, which also is not an unequivocally definite process. If this is the case, we should expect similarity judgements to vary according to the degree of freedom (or inclination) that subjects have to interpret the melodies in multiple ways.

At one extreme are probably the situations when someone compares themes and variations or when a teacher assesses a student's performance. In both cases, there is a presumption that the melodies should be similar, and so listeners are likely to *seek* the interpretations which allow maximum similarity. At the other extreme are situations when listeners have to make snap judgements or when they are asked to rank melodies for similarity. In the first case there will not be time for multiple interpretations; in the second there is an inclination to find difference as much as to find similarity.

If, in some situations at least, similarity is judged on the basis of maximising the similarity between interpretations of two melodies, we should *expect* triangle inequality to be violated: that melody *b* can be interpreted in different ways to be similar to both *a* and *c* does not imply that there is any way to interpret *a* to be similar to *c* (at least not in general; this conjecture would have to be tested with respect to specific methods of interpretation, such as reduction methods).

3.2 An example

There is no direct evidence for such multiple interpretation in similarity studies I know of, but I can retrieve a candidate case from a music analysis I made some years ago [21, 22]. Figure 1 shows extracts from the first violin part of Mozart's string quartet in C major, K. 465 "Dissonance"). The allegro begins with the theme shown as **a**. This is immediately repeated a tone higher (not shown) and then, with a slight modification, as **a'**. The last note of **a'** begins a new motive **b** which appears to contrast with **a** (descending instead of rising; made up largely of shorter notes; containing a large leap instead of mostly steps). This is repeated at **b'** (reinforcing the identity of the motive) and then in rhythmic transformation some bars later at **b''** (where the recognition of similarity is aided by using exactly the same pitches). Several bars later the figure identified as **b'''** is heard, whose similarity to **b''** is aided by the equivalent durations of the second

note (though in the case of **b'''** it is decorated with a trill). Finally, beginning on the same pitch as **b'''** and ending with the same pair of pitches, a figure is heard which is also clearly similar to **a** by inversion. (Indeed, to help make this clear, the intervening music has presented several other versions of **a** without inversion.) This figure is easily recognised as similar to *both* **a** and (with the aid of the intermediate transformations) **b**.

Is it true, then, that **a** is similar to **b**, despite the fact that at first the motives seemed to be contrasted? If it is, then we must reduce **a** in *different* ways to find maximum similarity in each case. To find maximum similarity between **a** and **a'**, we must reduce **a** by removing the appoggiatura on the last note, which implies that the remaining notes are passing notes from C to F. To find maximum similarity between **a** and **b**, on the other hand, the first step must be to reduce out the quavers in **a** and regard the appoggiatura (neighbour note) as prior. It was my contention in the original analysis [22] that Mozart intended this play with our sense of the difference and similarity between these motives as a way of capturing the listeners interest.

3.3 Exploring similarity through creativity

Listening to music, or indeed any human process with music, involves interpretation, and interpretation is always a *creative* act. When musicians say two melodies are similar, the arguments above suggest that the musicians have *created* that similarity as much as recognising it. While it is now not uncommon for researchers to claim that a single measure of melodic similarity for all situations is an impossibility (e.g., [7]), this argument suggests that it is an impossibility in *any* situation. The best one can hope for is a measure which will usefully approximate human judgements of similarity in such situations.

The distinction is perhaps technical, since no researchers have claimed to derive a perfect measure of melodic similarity, but it does imply a radically different research perspective. In particular, it suggests that melodic similarity might profitably be explored in explicitly creative situations. For example, a system which aimed to allow users to compose music on the basis of arranging similar and contrasting melodic fragments might be based on competing models of similarity. Then by observing users interaction with the software (probably silently through background monitoring), data could be gathered about which model was most useful for achieving the users' artistic goals.

3.4 Using similarity to explore reduction

Another possible research direction which turns previous research on its head is to use similarity as a means for investigating reduction rather than the other way around. As mentioned above, the bases for making Schenkerian reductions are not well understood, and there are not pre-existing paradigms for their discovery. If my hypothesis that similarity, at least in some situations, is based on finding the maximally similar reductions of two melo-

dies, then melodies which are known to be similar could be used as ground truths for guiding reduction. This has the advantage over the approach taken in [20] that, instead of being based on the activities of experts directed towards either pedagogy or analytical debate, it is based on the practice of real composers and listeners. Sets of variations, in particular, provide a promising ground for such investigations.

3.5 Creativity of music information retrieval

Researchers who develop systems for measurement of musical similarity generally take a scientific approach, judging their success or failure by the degree to which results match observations. Yet they too are creative, or at least have a creative influence, not only in the general sense of making something new, but also in a musical sense. They might not make new pieces of music, but their work will certainly lead to new kinds of musical experience.

The recent past provides numerous examples of similar creative impact of scientific advances. The invention of MP3 encoding, for example, in conjunction with the internet, has created an entirely new environment in which to discover, obtain, experience and even create music, crucially creating new kinds of musical community [23]. The iPod and similar personal music devices (also dependent on the technology of MP3 and related encodings) has also radically affected common experiences of music. In contrast to previous centuries when the only way to experience music on one's own was to play it oneself, listening to music has become commonly an isolated and personal experience. Indeed, listeners commonly report using a mobile music device in order to create a 'personal space' [24], quite the opposite of the traditional necessary association of music with a social or communal space.

If the work of those who research melodic similarity leads to ubiquitous software which allows music to be rapidly retrieved on the basis of its similarity to a given model, what will be the impact on our musical culture, and on the nature of music which is created? And, since judgements of similarity are context-dependent (as discussed above), what will be the consequent effect on people's concepts of melodic similarity? Musical scientists too do not escape the uncertainty principle: in investigating melodic similarity they affect the very culture which generates the concept of melodic similarity itself.

4. CONCLUSION

Melodic similarity seems not to be a single relationship, but to be plural on at least two counts. Firstly, it differs from one context to another. Secondly, it depends of differing interpretations. The second of these is undoubtedly a creative act (though listeners do not generally regard themselves as creative). In enabling new ways of experiencing and encountering music, researchers of melodic similarity also have a creative impact on musical culture.

5. REFERENCES

- [1] M.A. Armstrong, *Basic Topology*, Springer-Verlag, 1983.
- [2] P. Giannopoulos & R.C. Veltkamp, "A pseudo-metric for weighted point sets," in *Proceedings European Conference on Computer Vision (ECCV 2002)*, LNCS 2352, Springer, 2002, pp. 715–730.
- [3] R. Typke & A.C. Walczak-Typke, "A tunnelling-vantage indexing method for non-metrics," in *Proceedings International Conference on Music Information Retrieval (ISMIR)*, Philadelphia, 2008, pp. 683–688.
- [4] T. Eerola, T. Järvinen, J. Louhivuori & P. Toiviainen, "Statistical features and perceived similarity of folk melodies," in *Music Perception*, 2001, pp. 275–296.
- [5] T. Eerola & M. Bregman, "Melodic and contextual similarity of folk song phrases," in *Musicae Scientiae*, 2007, pp. 211–233.
- [6] D. Müllensiefen & K. Frieler, "Cognitive adequacy in the measurement of melodic similarity: algorithmic vs. human judgments," in *Computing in Musicology*, 2004, 147–176.
- [7] D. Müllensiefen & K. Frieler, "Modelling experts' notions of melodic similarity," in *Musicae Scientiae*, 2007, pp. 183–210.
- [8] M.A. Schmuckler, "Melodic contour similarity using folk melodies," in *Music Perception*, 2010, pp. 169–193.
- [9] R. Typke, R. Wiering & R.C. Veltkamp, "Transportation distances and human perception of melodic similarity," in *Musicae Scientiae*, 2007, pp. 153–181.
- [10] H. Allan, D. Müllensiefen & G. Wiggins, "Methodological considerations in studies of musical similarity," in *Proceedings International Conference on Music Information Retrieval (ISMIR)*, Vienna, 2007, pp. 473–478.
- [11] A. Novello, M.M.F. McKinney & A. Kohlrausch, "Perceptual evaluation of inter-song similarity in western popular music," in *Journal of New Music Research*, 2011, pp. 1–26.
- [12] A. Volk, P. van Kranenburg, J. Garbers, F. Wiering, R.C. Veltkamp & L.P. Grijp, "A manual annotation method for melodic similarity and the study of melody feature sets," in *Proceedings International Conference on Music Information Retrieval (ISMIR)*, Philadelphia, 2008, pp. 101–106.
- [13] Z. Juhasz, "A systematic comparison of different European folk music traditions using self-organising maps," in *Journal of New Music Research*, 2006, pp. 95–112.
- [14] N. Hu, R. Dannenberg, A.L. Lewis, "A probabilistic model of melodic similarity," in *Proceedings of the International Computer Music Conference (ICMC)*, San Francisco, 2002, pp. 509–515.
- [15] B. Pardo & J. Shifrin, "Name that tune: a pilot study in finding a melody from a sung query," in *Journal of the American Society for Information Science and Technology*, 2004, pp. 283–300.
- [16] J.F. Bernabeu, J. Calera-Rubio, J.M. Iñesta & D. Rizo, "A probabilistic approach to melodic similarity," in *Proceedings of Music and Machine Learning*, 2009, pp. 48–53.
- [17] D. Rizo, *Symbolic music comparison with tree data structures*, PhD thesis, University of Alicante, 2010.
- [18] N. Orio & A. Rodà, "A measure of melodic similarity based on a graph representation of the music structure," in *Proceedings International Conference on Music Information Retrieval (ISMIR)*, Kobe, 2009, pp. 543–548.
- [19] A. Marsden, "Recognition of variations using automatic Schenkerian reduction," in *Proceedings International Conference on Music Information Retrieval (ISMIR)*, Utrecht, 2010, pp. 501–506.
- [20] A. Marsden, "Schenkerian analysis by computer: a proof of concept," in *Journal of New Music Research*, 2010, 269–289.
- [21] A. Marsden, "Listening as discovery learning," in *Contemporary Music Review*, 1989, pp. 327–340.
- [22] A. Marsden, *Analysing music as listeners' cognitive activity, a study with reference to Mozart*, PhD thesis, Cambridge University, 1987.
- [23] A. Tanaka, "Interaction, experience and the future of music," in K. O'Hara & B. Brown (eds.), *Consuming Music Together: Social and Collaborative Aspects of Music Consumption Technologies*, Springer, 2006, 271–292.
- [24] E. Nettamo, M. Nirhamo & J. Häkkinen, "A cross-cultural study of mobile music – retrieval, management and consumption," in *Proceedings 18th Australia conference on Computer-Human Interaction (OZCHI '06)*, 2006, 87–94.

REAL-TIME UNSUPERVISED MUSIC STRUCTURAL SEGMENTATION USING DYNAMIC DESCRIPTORS

André Pires

Department of Computer Science
University of São Paulo
andrespires@gmail.com

Marcelo Queiroz

Department of Computer Science
University of São Paulo
mqz@ime.usp.br

ABSTRACT

This paper presents three approaches for music structural segmentation, i.e. intertwined music segmentation and labelling, using real-time techniques based solely on dynamic sound descriptors, without any training data. The first method is based on tracking peaks of a sequence obtained from a weighted off-diagonal section of a dissimilarity matrix, and uses Gaussian models for labelling sections. The second approach is a multi-pass method using Hidden Markov Models (HMM) with Gaussian Mixture Models (GMM) in each state. The third is a novel approach based on an adaptive HMM that dynamically identifies and labels sections, and also sporadically reevaluates the segmentation and labelling, allowing redefinition of past sections based on recent and immediate past information. Finally, a method to evaluate results is presented, that allows penalization both of incorrect section boundaries and of incorrect number of detected segments, if so desired. Computational results are presented and analysed both from quantitative and qualitative points-of-view.

1. INTRODUCTION

Music structural segmentation is a task that underlies many important audio processing applications, including genre classification, audio summarization and music search [1]. Finding the temporal borders and labelling music sections according to common properties provides a means to approach such applications. Besides, real-time music structural segmentation may also be used in musical performances, for instance to control interactive processes responding to timbre variations.

In the literature we may find several methods for music segmentation. Cooper and Foote [2] propose a segmentation method based on finding peaks of a dissimilarity sequence built from a similarity matrix, whose main diagonal is traversed (and dot-multiplied) by a radial smoothing checkerboard kernel; additionally, they cluster similar sections via a statistical method. Aucouturier and Sandler [3] propose a segmentation method based on HMMs, using the sequence of observations as training data. Peeters,

la Burthe and Rodet [4] propose a multi-pass approach using an HMM which is initialized after finding the centroids of the sections using k-means. The main difference between the latter and the former method lies in the way the HMM is initialized, where HMM parameters are expected to converge to a local optimum that better represents the real music sections, avoiding over-segmentation due to fine-grained changes in spectral content, aiming for long-term structures [4].

The methods proposed in [2, 3, 4] are not specifically meant to be used in real-time, and they specifically require training data for initializing key structures used in segmentation. As opposed to that, we look specifically toward real-time unsupervised techniques, i.e. methods that operate promptly on an audio stream and do not require any *a priori* information about the stream content. In section 3.2 and 3.3 we present variations of methods found in [2, 4], and in section 3.5 we propose a novel approach based on an adaptive HMM that identifies section boundaries and labels sections in real-time, but also sporadically reevaluates the output of the segmentation, allowing a complete redefinition of past sections using more recent information, possibly information that was not yet available at the time a specific past section was identified. This allows the correction of errors imposed by the real-time output requirement, both in terms of redefining section boundaries as well as relabelling sections according to the most recent statistical models of the identified sections.

According to [9], all segmentation methods described in this paper could be categorized as *homogeneity-based* approaches, as they first depart from a *novelty-based* procedure to finally produce clusters based on the similarity between section models.

The goal of this paper is both to present a novel real-time method for the music structural segmentation problem, and to assess the results of segmentation using real-time unsupervised techniques, both from quantitative and qualitative points-of-view. We also aim to compare segmentation results using instantaneous and dynamic descriptors within such methods. Dynamic descriptors [4] provide an alternative temporal modelling for describing audio frames, combining information obtained from several audio frames, as opposed to instantaneous descriptors, which refer to a single audio frame.

The structure of this paper is as follows; in section 2 we present techniques for defining dynamic descriptors; in section 3, we develop the aforementioned real-time un-

supervised techniques for automatic music structural segmentation; and in section 4 we present a method to evaluate the results obtained by these techniques, which is used to compare their performances using instantaneous MFCC descriptors and several dynamic MFCC descriptors.

2. DYNAMIC DESCRIPTORS

Choosing the right sound features can strongly influence the segmentation's success or failure. As pointed by [4], Mel Frequency Cepstral Coefficients (MFCC) can be considered *static* features, as they represent sound restricted by an analysis frame (usually 30 to 100 ms), as opposed to *dynamic* features, that can model the temporal evolution of sound.

One way to model dynamic signal evolution is shown in [4], where temporal evolution is modelled after the spectral shape of the signal energy, divided in Mel sub-bands. Another approach for generating dynamic descriptors is based on the idea of modelling temporal evolution from any descriptors extracted from the audio signal [5], using a smoothing function that condenses several past observations into a single value. This attenuates abrupt changes in the sequence of observations at the frame-rate temporal level, which might easily confuse segmentation techniques, but hopefully it preserves timbre changes at larger-scale temporal levels.

Let $X = \{x_n\} \subset \mathbb{R}^d$, $n = 1, \dots, N$ be the observation sequence extracted from the audio signal, where n is a frame index. Let L be the number of past observations, we define a dynamic descriptor sequence $Y = \{y_n\} \subset \mathbb{R}^\alpha$, $n = L + 1, \dots, N$ as

$$y_n = D(x_{n-L}, x_{n-L+1}, \dots, x_n), \quad (1)$$

where D is a smoothing function that takes into account L observations and α is the new dimension after the transform. This smoothing function D can be defined in several ways; we considered four strategies: statistical moments, Euclidean norm, exponential decay and FFT coefficients.

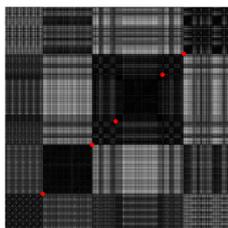


Figure 1. Similarity Matrix generated using raw MFCC descriptors. The red dots are the real transitions between sections, manually defined.

To illustrate these strategies, consider the similarity matrix of Figure 1, generated using MFCC. Figures 2a to 2h display two similarity matrices for each strategy, with dynamic descriptors using temporal memories of 1 and 10 seconds, corresponding in our experiments to $L = 20$ and $L = 200$, respectively. In those images the *red dots* are the real transitions between sections, which were manually

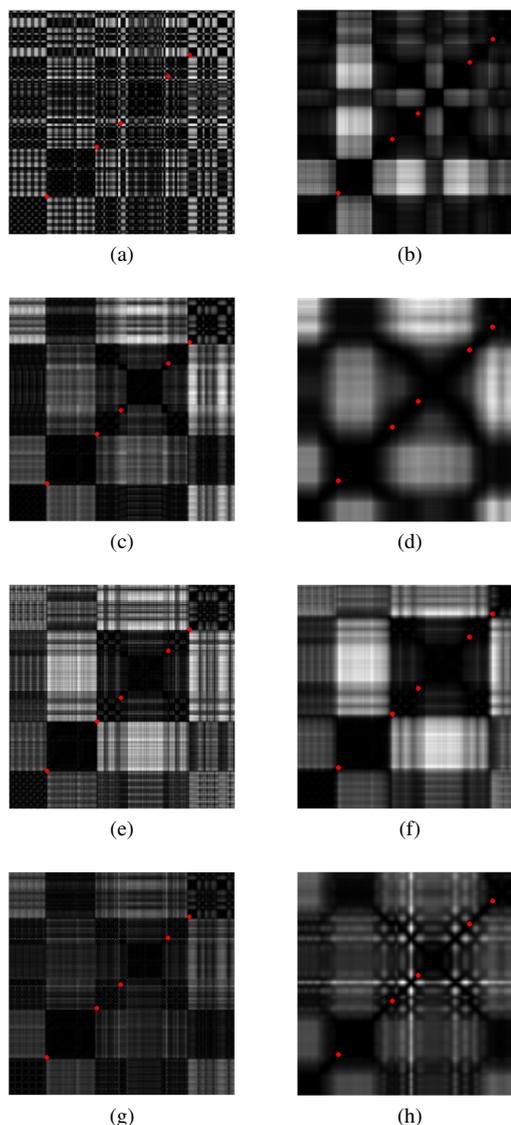


Figure 2. Similarity matrices generated using dynamic descriptors with memory parameters of 1 and 10 seconds, shown in left and right column, respectively. Figures (2a) and (2b) use *Statistical Moments*, figures (2c) and (2d) use *Euclidean Norm*, figures (2e) and (2f) use *Exponential Decay* and figures (2g) and (2h) use *FFT coefficients*. The red dots are the real transitions between sections, which were manually defined.

defined. The similarities were calculated using the cosine similarity measure.

Thus, for each $x_{k,t}$ (i.e. the dimension k of x at instant time t) we can generate the descriptors as follows.

Statistical Moments. In the dynamic descriptors generated with statistical moments, each $x_{k,t}$ provides four new values: mean, variance, skewness and kurtosis.

Euclidean Norm. This dynamic descriptor represents the temporal series corresponding to $x_{k,t}$ as the Euclidean norm of L previous values, similarly to what RMS amplitude does with respect to instantaneous amplitude values.

Exponential Decay. With this dynamic descriptor, temporal memory is represented by an exponential decay func-

tion $w(n) = e^{-2\pi n}$, which is used to attenuate past observations.

FFT coefficients. With this method, each dimension k of x_t is segmented in windows of size L , from which Fast Fourier Transform (FFT) coefficients are calculated. Only the first 7 coefficients are selected, corresponding to the slow variation of the spectrum of that descriptor.

3. REAL-TIME STRUCTURAL SEGMENTATION

To define objectively and above all suspicion what a *musical section* is might be a harsh task. Putting aside the music theoretic issues and focusing only on the properties of sound (more specifically the *polyphonic timbre* as defined by [3]), we may define a section as an audio fragment with a contiguously varying global timbre description. Pitch, rhythm and amplitude might give some clues to where section boundaries are, but in this work we aim to capture section changes only by identifying contiguous sections of global timbre stability.

In real-time unsupervised music structural segmentation there is no available training data, and the input data are flowing continuously. One issue we have to deal with is to define the right moment when the identified sections are going to be labelled. As the input data keeps flowing in, labels and section boundaries might also change, and until the music ends it is not possible to define all musical sections and labels. Despite this theoretical impossibility, it might be interesting to consider musical applications of incomplete or provisional segmentation and labelling, for instance, for switching on and off interactive sound-processing units depending on variation of global timbre. Defining an incomplete segmentation and labelling at time t as the result of those processes assuming the music had literally ended at time t is at least theoretically sound.

Algorithm 1 outlines a general solution for real-time structural segmentation. At first, it continuously reads the descriptor vector and accumulates these observations until it reaches a minimum section size allowed. The core of the algorithm is formed by the function that finds section changes (line 5) and the function that labels sections according to timbre proximity (line 17). In the following sections we will discuss the techniques used to solve these two problems. In section 3.1 we will present a first technique to automatically label the sections; alternative techniques will be presented in sections 3.3 and 3.5, which uses Viterbi's algorithm on an HMM to label the sections.

3.1 Labelling

The primary task in labelling is to identify which musical sections have the same characteristics, i.e. share a common global timbre so they may receive the same label. For this purpose we need a method to compare all sections found and to calculate some measure of section separability. This measure should determine how similar is a section model ω_i to another section model ω_j , and a threshold on this measure would tell us whether or not to label them equally. For didactic purposes let us consider only two section models: ω_i and ω_j ; this labelling method will extend to a set

Algorithm 1 Real-time structural segmentation

Require: I ; Input data

w_{\min} ; minimum window size

w_{\max} ; maximum window size

Ensure: S ; section change locations

L ; labelled sections

```

1:  $p \leftarrow 1, T \leftarrow \{\}, x \leftarrow \text{read}(I)$ 
2: while  $x \neq \text{NULL}$  do
3:    $T \leftarrow T \cup \{x\}$ 
4:   if  $|T| + 1 \geq w_{\min}$  then
5:      $relative \leftarrow \text{LocateChangePoint}(T)$ 
6:     if  $relative = \text{NULL}$  then
7:       if  $|T| + 1 > w_{\max}$  then
8:          $T \leftarrow \{t_i : t_i \in T, i \geq w_{\min}\}$ 
9:       end if
10:      else
11:         $absolute \leftarrow (p - 1) - |T| + relative$ 
12:         $T \leftarrow \{t_i : t_i \in T, i > relative\}$ 
13:         $S \leftarrow S \cup absolute$ 
14:      end if
15:       $p \leftarrow p + 1$ 
16:    end if
17:     $L \leftarrow \text{Label}(S), x \leftarrow \text{read}(I)$ 
18: end while

```

of P sections in a straightforward fashion. This method assumes that section models are normally distributed.

The Bhattacharyya distance was developed to estimate the classifier error, but it is also used as a class separability measure, and provides an upper bound to the classifier error, similarly to the Chernoff Bound [6]. This distance between section models $\omega_i = \mathcal{N}(\mu_i, \Sigma_i)$ and $\omega_j = \mathcal{N}(\mu_j, \Sigma_j)$ is given by

$$d_{ij} = \frac{1}{8}(\mu_i - \mu_j)^T \left(\frac{\Sigma_i + \Sigma_j}{2} \right)^{-1} (\mu_i - \mu_j) + \frac{1}{2} \ln \frac{\left| \frac{\Sigma_i + \Sigma_j}{2} \right|}{\sqrt{|\Sigma_i| |\Sigma_j|}}$$

Once we have calculated the distance for each pair of section models, those who have very small distances shall be clustered, sharing the same label. In order to cluster similar sections, we have used average linkage clustering [7], and the clusters are given by calculating the inconsistency coefficient ξ_i for each link i , established by the clustering algorithm, and cutting the dendrogram at any point between the two links having the largest inconsistency values [8]. The inconsistency coefficient is a measure of how similar the links below each link are, calculated as $\xi_i = \frac{z_i - \mu_{z_i}}{\sigma_{z_i}}$, where z_i is the i -th link height, μ_{z_i} is the mean of all link's heights at the same depth and level and σ_{z_i} is the standard deviation of all link's heights at the same depth and level. Figure 3a shows a dissimilarity matrix using the Bhattacharyya distance between section models before the clustering algorithm. Light shades indicate small distances between models, meaning that a cluster can possibly be formed. Depending on the threshold we might say that the second and third section models are candidates to form a cluster, as well as the sixth, seventh and eight models. Figure 3b shows the section models after the clustering algorithm. Note that the clustering algorithm may join not only temporally adjacent sections, but also temporally dis-

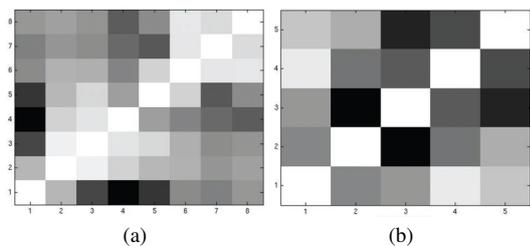


Figure 3. Dissimilarity between section model pairs, calculated before (figure 3a) and after (figure 3b) clustering, using Bhattacharyya distance.

tant sections, according to the similarity of their statistical models.

The above method is akin to the one in [2], but there a Singular Value Decomposition of the similarity matrix is used, and section models are clustered according to the Kullback-Leibler distance.

3.2 Dissimilarity Matrix Peaks

In the Dissimilarity Matrix Peaks (DISSM-PEAKS) method, transitions are detected through the rapid and abrupt changes in the temporal observation sequence. To this end, we calculate a dissimilarity sequence using the information given by the dissimilarity matrix and select the peaks of this sequence, which are the section transitions. The dissimilarity matrix is built using the cosine distance. After we locate the transitions, each section is mapped into a normal distribution and we label them according to subsection 3.1. The details of this method are as follows.

1. Let N be the number of observations. The dissimilarity sequence, defined as $\Delta = \{\delta_i : 1 \leq i \leq N\}$, is calculated by scanning a certain neighborhood of the dissimilarity matrix M from each diagonal point M_{ii} . This neighborhood can be characterized by its size $h \ll N$ and a vector ν_i , $1 \leq i \leq N$ containing the neighboring observations $\nu_i = \{M_{i+k, i-k} : 1 \leq k \leq h, 1 \leq i-k, i+k \leq N\}$. The dissimilarity is defined as the inner product $\delta_i = \langle W, \nu_i \rangle$ of the observation vector and an exponential decay vector $W = \{w_j : 1 \leq j \leq h, w_j = e^{-2\pi j(h-1)^{-1}}\}$.
2. Detect the peaks of the vector Δ using a noise tolerant peak finder algorithm¹. Peak magnitudes must be ordered in descending order, in order to select candidate transition points with highest dissimilarity. The final transition points are obtained after the parameter w_{\min} (minimum section size) is used to exclude peaks that lie too close to other candidate transition points with higher dissimilarity. Temporal indices $T = \{t_i : 1 \leq i \leq P\}$ of the P selected peaks are considered transitions points for $P+1$ sections $\{[x_{t_0}, x_{t_1}], [x_{t_1}, x_{t_2}], \dots, [x_{t_P}, x_{t_{P+1}}]\}$, where $t_0 = 1$ and $t_{P+1} = N+1$.
3. Each section $[x_{t_k}, x_{t_{k+1}}]$ is mapped to a normal distribution $\mathcal{N}(\mu_k, \Sigma_k)$, where the mean vector μ_k and the covariance matrix Σ_k are estimated from the observation data contained in $[x_{t_k}, x_{t_{k+1}}]$.

¹ In our experiments we used a MATLAB[®] code by Nathanael C. Yoder, peakfinder.m

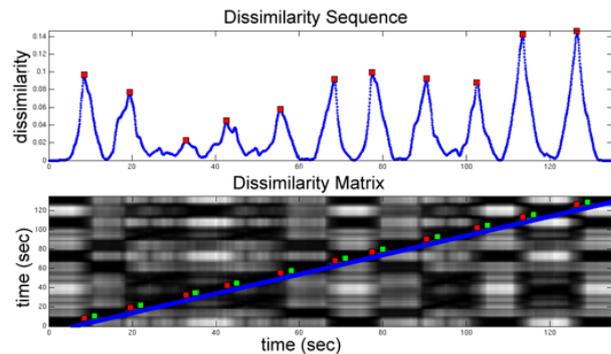


Figure 4. Transitions between sections found with the Dissimilarity Matrix Peaks method. The red dots in dissimilarity matrix are the peaks found before the adjustment, the green dots are the peaks after the adjustment, and the blue strip is the neighborhood used to build the dissimilarity sequence.

4. Label the sections using the method described in subsection 3.1.

When the dissimilarity matrix M is generated using dynamic descriptors, the observation sequence gets smoother as we increase the temporal memory L , and so we need to navigate farther away from the diagonal of the dissimilarity matrix to be able to detect peaks. Thus, it is convenient to change the neighborhood definition so that it starts with an offset from the diagonal, as $\nu_i = \{M_{i+k, i-k} : 1 + \rho \leq k \leq h + \rho, 1 \leq i - k \leq i + k \leq N\}$, where the parameter $\rho = L/2$ ensures that the method will exclude smoothed observations that are affected by $M_{i,i}$.

Furthermore, also due to temporal smoothing, temporal transitions T found by the method are anticipated of $L/2$ observations, which means that we also need to adjust all points to $\hat{t}_i = t_i + L/2$. In our experiments we have used a 1 second neighborhood, corresponding to $h = 20$, which fits very well for our dataset, but it must be estimated according to a given musical context/genre.

Figure 4 shows the dissimilarity sequence and the transition points found calculated using exponential decay dynamic descriptors, where L corresponds to 5 seconds.

3.3 Multi-pass GMM-HMM

The Multi-Pass GMM-HMM (MPS-GHMM) method is based on the multi-pass algorithm by [4]; the motivation for this variant is modelling the states (potential, initial and final) as mixtures of Gaussian distributions. Using our database set in another running tests, this variation improved the method described in [4], leading to an error of 6.8% against 15.25%. The method can be summarized in the following steps:

Build potential states. First, we identify the boundaries of each section. This is accomplished by the method described in subsection 3.2. To build potential states, we select the observations within each section and estimate the parameters of the Gaussian Mixture Models (GMM) using the Expectation Maximization (EM) algorithm [10]; the potential states are modelled as $s_i(x) =$

$\sum_{m=1}^M c_{im} p_{im}(x)$, where i is the i -th state, c_{im} is the m -th mixture coefficient, and the m -th pdf is $p_{im}(x) = \frac{1}{2\pi^{d/2} |\Sigma_{im}|^{1/2}} e^{-0.5*(x-\mu_{im})^T \Sigma_{im}^{-1} (x-\mu_{im})}$ with mean vector μ_{im} and covariance matrix Σ_{im} . See Figure 5a.

Reduce potential states. The number of potential states is reduced in a similar way to what is done in [4], i.e. potential states having similarity $> .99$ are grouped together. In [4], the potential states are represented by the mean observation vector within each section, and the cosine distance is used to measure similarity. In our case, the potential states are GMMs having m mean vectors and m covariance matrices, so we used the Bhattacharyya distance to measure similarity and summing up the distances for each Gaussian of the mixture. *Initial states* are generated after grouping potential states; initial states are also modelled as GMMs, whose parameters are estimated using the EM algorithm and the observations of each potential state. In other words, if there is a group $\{s_a, s_b\}$, then GMM parameters of each initial state are estimated through the observations $X_a \subset X$ and $X_b \subset X$ restricted to each potential state section. At the end of this step, K initial states are built. See Figure 5b.

Introduce time constraints. The timing constraints are applied similarly as in [4], except that we can use a mixture of normal distributions instead of a simple normal distribution. The K initial states are used to initialize a continuous ergodic HMM [10]. The parameters of the HMM are re-estimated using the Baum-Welch algorithm with the training data set X , and so we obtain the *final states*. The final music labels are obtained using Viterbi's algorithm, given the HMM and the descriptor vectors X . See Figure 5c.

3.4 Bayesian Information Criterion

The Bayesian Information Criterion (BIC) is a very well-known hypothesis test method in statistics, but to our knowledge it has not yet been used specifically for Music Information Retrieval. Nonetheless, there are works [11, 12] using BIC to detect acoustic changes in an audio signal, e.g. to detect changes from one speaker to another.

BIC is based on the log likelihood ratio between two Gaussian models, and in our case is used to compare two competing hypotheses: either an audio fragment has (at least) one possible splitting point defining (at least) two sections, or else the whole fragment belongs to a single section. Thus, let $X_{i,j} = x_i, \dots, x_j$ be the observation sequence. We'd like to test the hypothesis of $X_{i,j}$ being adequately modelled by a single normal distribution $\mathcal{N}(\mu, \Sigma)$,

$$H_0 : x_i, \dots, x_j \sim \mathcal{N}(\mu, \Sigma) \quad (2)$$

versus the hypothesis of $X_{i,j}$ being split in two parts adequately modelled by two different normal distributions

$$H_1 : \begin{aligned} x_i, \dots, x_k &\sim \mathcal{N}(\mu_1, \Sigma_1) \\ x_{k+1}, \dots, x_j &\sim \mathcal{N}(\mu_2, \Sigma_2). \end{aligned} \quad (3)$$

Parameters for each normal distribution are computed from the corresponding observation sequence, and depend on the splitting point k . The log-likelihood ratio is given by $R(k) = N \log |\Sigma| - N_1 \log |\Sigma_1| - N_2 \log |\Sigma_2|$ where $N =$

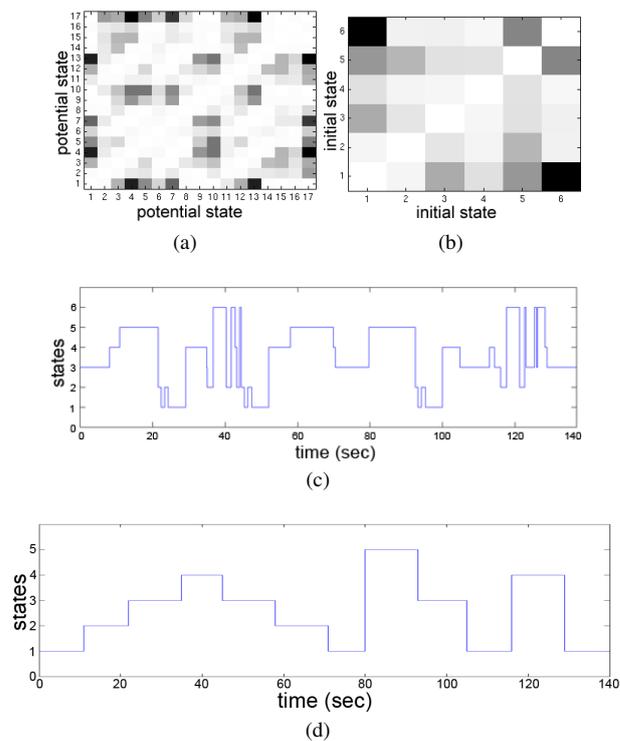


Figure 5. Potential states found using the dissimilarity sequence method – Figure 5a. Initial states after grouping – Figure 5b. Final states and labelling after Viterbi's algorithm – Figure 5c. Reference Segmentation, for comparison – Figure 5d.

$|X_{i,j}|$, $N_1 = |X_{i,k}|$ and $N_2 = |X_{k+1,j}|$, i.e. the number of observations within each set. The BIC values for each index k is

$$BIC(k) = R(k) - \lambda \Pi \quad (4)$$

which is a function of the maximum likelihood ratio and a penalty factor $\Pi = \frac{1}{2}(d + \frac{1}{2}d(d+1)) \log N$, depending on the number of random variables d in the observation vector X , weighted by λ (in our experiments we have used $\lambda = 1$). The hypothesis H_1 with two different Gaussian models offers a better model (compared to H_0) whenever the result of this equation is positive. Thus the transition point that provides the best splitting point (corresponding to the maximum likelihood ratio) is given by

$$k = \underset{k}{\operatorname{argmax}} \{ BIC(k) \mid BIC(k) > 0 \}. \quad (5)$$

In real-time segmentation with BIC, after locating the section boundaries, constrained to w_{min} (see algorithm 1) in order to prevent over-segmentation, the labelling is done using the method described in subsection 3.1.

3.5 Adaptive HMM

The Adaptive HMM (AHMM) is a real-time method which aims at labelling sections as each transition point between sections is identified, and not afterwards, as in the BIC method. In other words, labelling is done simultaneously with real-time segmentation. The algorithm works in two stages: a cache stage and a reevaluate stage.

3.5.1 Cache Stage

Each transition point t_k found provides a new section $T = [x_{t_{k-1}}, x_{t_k})$, and triggers the labelling routine. Let $B = \{b_j\}$ be the set of existing section labels of the real-time routine. The model of the new section is compared to each model corresponding to $b_j \in B$, and the model having higher similarity (mean log-likelihood) is updated with the observations of the new section T , provided that this similarity is higher than a minimum threshold α ; if this is not the case, a new label is added to B , with a model built from T . Note that each state is modelled as a mixture of Gaussian distributions

$$b_j(x) = \sum_{m=1}^M c_{jm} p_{jm}(x), \quad (6)$$

where M is the number of Gaussians per state and $p_{jm}(x)$ is the Gaussian density function (see Section 3.3).

The details of the first stage of this method are as follows:

1. Initialize HMM $\lambda(A, B, \pi)$ with 0 states, where $A = \{a_{ij}\}$ is the state transition probability matrix and $\pi = \{\pi_j\}$ is the initial state probability distribution.
2. Find a transition point t_k defining a section $T = [t_{k-1}, t_k)$, as described in algorithm 1 (section 3). For instance, using the BIC method 3.4.
3. Find a label b_j with model closest to T :

$$j = \operatorname{argmax}_i \left\{ \theta_i = \frac{1}{|T|} \sum_{t_k \in T} \log b_i(t_k) \right\} \quad (7)$$

If $\theta_j \leq \alpha$ then a new section model b_{S+1} is added to B , with parameters estimated by the EM algorithm.

If $\theta_j > \alpha$ then the observations in T are added to the model of b_j , whose parameters are re-estimated using the EM algorithm.

4. Update the state transition matrix A , reinforcing the transition between the previously detected state model and b_j (or b_{S+1} , if $\theta_j \leq \alpha$). Return to step 2.

Instead of simply applying the label with higher similarity to the observations of each section found, observations are labelled according to λ , using Viterbi's algorithm.

3.5.2 Reevaluate Stage

As time passes, the HMM model represented by λ accumulates errors due to the requirement of real-time operation, in the sense that models that were defined and made sense for segmentation at time t are carried on and keep influencing labelling at times $t + 1$, $t + 2$ and so on. This shows the need for a model refinement stage, where up-to-date knowledge about the input is used to reevaluate all previously defined sections and labels. The triggering of this refinement routine can be done by several different means. For instance, refinement may be triggered by user intervention, by a threshold on the number of HMM states allowed, or it might be automatically triggered every K iterations for a pre-defined K . The refinement stage of this method is described as follows.

1. Cluster similar section models using hierarchical cluster and Bhattacharyya distance as similarity measure (section 3.1).
2. Update the state transition matrix A by summing up the grouped state transitions.
3. Reestimate the HMM parameters $\hat{\lambda}(\hat{A}, \hat{B}, \pi)$, using the Baum-Welch algorithm. The corresponding state sequence for the observations X is obtained by Viterbi's algorithm.

Figure 6a shows all states (labels) found in a musical piece just before the refinement stage. Figure 6b shows the same piece after the refinement stage, where we can see a reduction from 26 to 8 states.

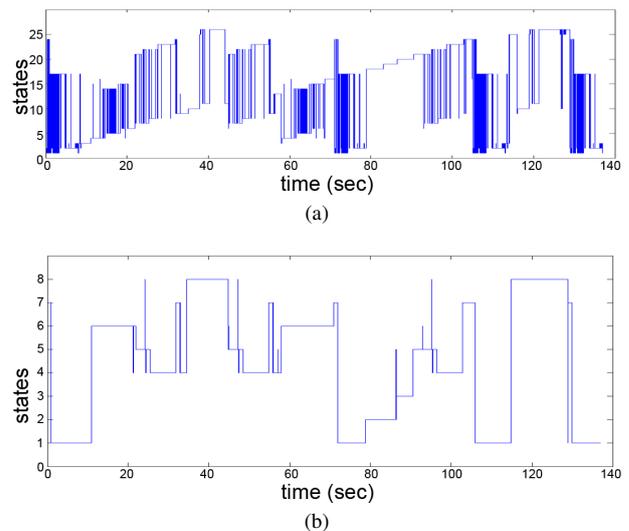


Figure 6. State sequence before (6a) and after (6b) the clustering step in Adaptive HMM. Refer to Figure 5d for reference segmentation.

This method is sensitive to the α parameter, which must be estimated. In our experiments we under-estimated its value (for the record, $\alpha = -0.016138$), so the algorithm tended to produce a larger number of states, which would hopefully be fixed by the second stage. An alternative is to let the user decide which level of similarity is suitable for a given musical context.

4. RESULTS

Having defined the segmentation and labelling techniques, we would like to compare the probability error of applying these methods to a set of musical pieces. We built a musical database containing 41 pieces of different genres with up to 7 different musical parts each, manually segmented and labelled. For each piece, we extracted MFCC descriptors and four groups of dynamic descriptors as in section 2: *moments*, *Euclidean*, *exponential decay* and *FFT coefficients*. Each group was computed using four different values for temporal memory: 0.5, 1, 5 and 10 seconds, totalling 697 executions for each method described in the previously section.

To evaluate segmentation and labelling performances, a measure of probability error is defined in subsection 4.1, which corresponds to a lower bound on the number of temporal positions wrongly labelled. Results are collected in groups of dynamic descriptors, from which we compare the performance of different values of the temporal memory parameter.

4.1 Evaluation

The segmentation and labelling evaluation depends on two factors: the labelling error probability (i.e. a measure of whether the labels in the test sequence are consistent with the labels in the reference sequence), and the segmentation error probability (i.e. a measure of the temporal errors in detected section boundaries). There are many metrics for evaluating the segmentation results, some of which do not consider the temporal order of observations, but the overall frames [9]. We propose here a measure that takes as input the temporal sequence of manually-assigned (correct) labels for each analysis frame, and the corresponding sequence produced by the algorithm. Thus, temporal discrepancies in the localization of boundaries will be translated into the (relative) number of analysis frames wrongly labelled.

Let N be the number of observations in X , p the number of labels manually defined in the reference sequence $f(t)$, such that $f(t) : \mathbb{N} \rightarrow \{1, \dots, p\}$, and let q be the number of labels found in the test sequence $g(t)$, such that $g(t) : \mathbb{N} \rightarrow \{1, \dots, q\}$. We wish to find a relabelling function $m : \{1, \dots, q\} \rightarrow \{1, \dots, p\}$, corresponding to a translation of the algorithmically produced labels to the manually defined labels, in such a way that the relabelled sequence $h(t) = m(g(t))$ is closest to the reference sequence in the sense of minimizing the error ratio

$$\epsilon(m) = \frac{1}{N} \sum_{t=1}^N \delta(f(t), h(t)), \quad (8)$$

where δ is the Kronecker Delta. This is necessary due to the fact that the labelling method does not necessarily provide the same symbols adopted by the reference sequence.

Although this relabelling function can be constrained in different ways according to different interpretations, in this work we will adopt a measure that focuses on the temporal accuracy of section boundaries, and only implicitly penalizes differences in the number of reference labels p and produced labels q . Thus we will consider as candidate relabelling functions all p^q sequences of size q generated by the elements $\{1, \dots, p\}$. Instead of a brute force search, we used an association matrix $s_{f(t),g(t)}$, from which the optimal labelling function according to $\epsilon(m)$ is given by the Hungarian algorithm [13].

One interesting aspect of this measure is the fact that it does not over-penalize a segmentation strategy that would subdivide reference sections because of minor timbre variations, but it does penalize incorrectly detected section boundaries. Theoretically, this measure would not penalize a degenerate labelling that associated a different label to every frame, but no honest structural segmentation method

would produce such a degenerate solution. An alternative set of relabelling functions that do not suffer from this theoretical problem is the set of injections from $\{1, \dots, q\}$ to $\{1, \dots, p\}$, where the second set is enlarged with artificial reference labels whenever $q > p$. This alternative would severely penalize over-segmentations but would be proportionately less stringent on sloppy section boundary detection.

4.2 Comparative Results

To evaluate the impact of the temporal memory parameter on the segmentation and labelling methods, we grouped the results by dynamic descriptor type. Besides executing all methods described previously, we added the K-Nearest Neighbors (K-NN) method for the sake of comparison. As a well-known supervised classification technique, meaning it has access to training data prior to classification, K-NN is expected to surpass every unsupervised competitor, but the obtained error values may be used as a baseline. For the sake of simplicity, we have used a 1-NN configuration, and a 10-fold cross-validation for evaluation.

In our experiments we used 13 MFCC extracted from 100 ms analysis windows with a feature extraction rate of 20 Hz. Comparing the results within each dynamic descriptor type (Figure 7) we see that the segmentation errors generally increase with temporal memory, except for the MPS-GHMM technique, where we can clearly see that smaller values of temporal memory (or no memory, i.e raw MFCC) also produce large errors.

Table 1 displays the best error results for each technique separately, i.e. which dynamic descriptor (if any) worked best for each technique. We may see from this table that only the *Exponential Decay* dynamic descriptors were able to improve error bounds with respect to raw MFCC, whereas *FFT coefficients*, *Euclidean* and *moments* dynamic descriptors followed similar patterns, but with slightly worse error values.

| | Error | Descriptor |
|-------------|--------------|--------------------------|
| BIC | 14.81% | MFCC |
| AHMM | 20.11% | MFCC |
| MPS-GHMM | 5.47% | MFCC Exp. Decay, 1 sec |
| DISSM-PEAKS | 15.41% | MFCC Exp. Decay, 0.5 sec |
| K-NN | 2.57% | MFCC Exp. Decay, 1 sec |

Table 1. Minimum error rate for each technique and descriptors that provided best results.

5. CONCLUSIONS

In this paper, we have presented three unsupervised methods for real-time music structural segmentation, including a novel one, and we compared them to a fourth method [11, 12]. We also presented a method to evaluate competing hypotheses for segmenting the same piece, aiming primarily at capturing temporal location errors of section boundaries. We have discussed the generation of dynamic descriptors based on the temporal modelling of MFCC.

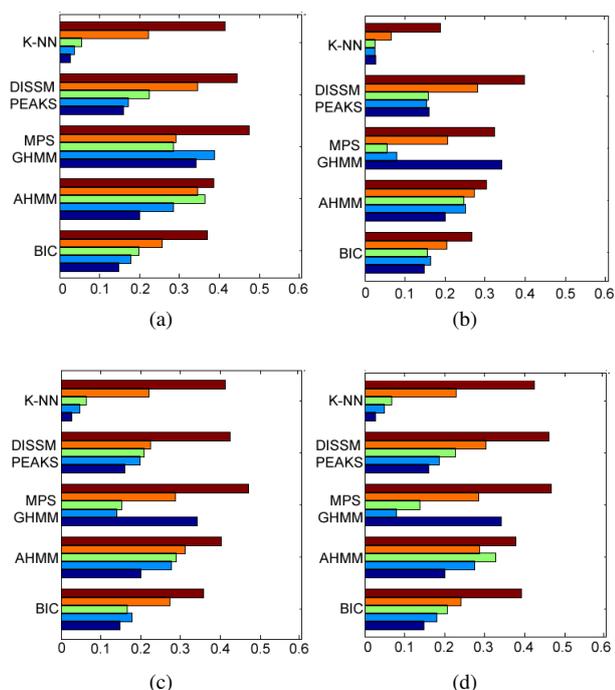


Figure 7. Error rate for each of the technique using *Moments* (Figure 7a), *Exponential Decay* (Figure 7b), *Euclidean* (Figure 7c) and *FFT coefficients* (Figure 7d) dynamic descriptors. Bar colors refer to temporal memory values for the dynamic descriptors: dark blue = raw MFCC values, light blue = 0.5 sec, green = 1 sec, orange = 5 sec and brown = 10 sec.

A number of experiments were performed to compare different methods using dynamic descriptors with different temporal memory values. The primary results are promising, showing that it is possible to perform such tasks in real-time, without any training data. The results show that temporal modelling of static music descriptors can lead to improved results, and that the best dynamic descriptors are those which do remember past information but do not overemphasize it, as is in the case of raw MFCC and Exponential Decay temporal memory.

Further work will consider using other evaluation techniques and a standard musical database, as described in [9].

Acknowledgments

We have used GMM algorithms from Netlab², HMM algorithms from PMTK3³, and the K-NN algorithm from Weka [14].

This work has been supported by CAPES, CNPq and FAPESP (grant 2008/08632-8). We would also like to thank Nina Hirata, Roberto Hirata, Airlane Alencar and Miguel Ramirez for valuable discussions, and the Brazilian company E-BIZ Solution.

² <http://www1.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/>. March 20th, 2011

³ <http://code.google.com/p/pmtk3>. March 20th, 2011

6. REFERENCES

- [1] R. Dannenberg and M. Goto, "Music structure analysis from acoustic signals," *Handbook of Signal Processing in Acoustics*, pp. 305–331, 2009.
- [2] M. Cooper and J. Foote, "Summarizing popular music via structural similarity analysis," in *Applications of Signal Processing to Audio and Acoustics, IEEE Workshop on.*, 2003, pp. 127–130.
- [3] J. Aucouturier and M. Sandler, "Segmentation of musical signals using hidden Markov models," *110th Convention of the Audio Engineering Society*, 2001.
- [4] G. Peeters, A. La Burthe, and X. Rodet, "Toward automatic music audio summary generation from signal analysis," in *Proc. International Conference on Music Information Retrieval*, 2002, pp. 94–100.
- [5] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," *Project Report*, 2004.
- [6] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*. Academic Press, 2008.
- [7] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *The Computer Journal*, vol. 26, no. 4, p. 354, 1983.
- [8] T. Korenius, J. Laurikkala, M. Juhola, and K. Jarvelin, "Hierarchical clustering of a finnish newspaper article collection with graded relevance assessments," *Information Retrieval*, vol. 9, no. 1, pp. 33–53, 2006.
- [9] J. Paulus, M. Müller, and A. Klapuri, "Audio-based music structure analysis," in *Proc. 11th International Conference on Music Information Retrieval*, 2010.
- [10] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [11] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion," *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998.
- [12] M. Omar, U. Chaudhari, and G. Ramaswamy, "Blind change detection for audio segmentation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Philadelphia, USA*, 2005.
- [13] H. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

MULTIPLE-INSTRUMENT POLYPHONIC MUSIC TRANSCRIPTION USING A CONVOLUTIVE PROBABILISTIC MODEL

Emmanouil Benetos and Simon Dixon

Centre for Digital Music, Queen Mary University of London, London E1 4NS, UK

{emmanouilb, simond}@eecs.qmul.ac.uk

ABSTRACT

In this paper, a method for automatic transcription of music signals using a convolutive probabilistic model is proposed, by extending the shift-invariant Probabilistic Latent Component Analysis method. Several note templates from multiple orchestral instruments are extracted from monophonic recordings and are used for training the transcription system. By incorporating shift-invariance into the model along with the constant-Q transform as a time-frequency representation, tuning changes and frequency modulations such as vibrato can be better supported. For postprocessing, Hidden Markov Models trained on MIDI data are employed, in order to favour temporal continuity. The system was tested on classical and jazz recordings from the RWC database, on recordings from a Disklavier piano, and a woodwind quintet recording. The proposed method, which can also be used for pitch content visualization, outperforms several state-of-the-art approaches for transcription, using a variety of error metrics.

1. INTRODUCTION

The goal of an automatic music transcription system is to convert an audio recording into a symbolic representation, such as a piano-roll, a MIDI file or a music sheet. The creation of a system able to transcribe music produced by multiple instruments with a high level of polyphony continues to be an open problem in the research community, although monophonic pitch transcription is largely considered solved. For a comprehensive overview on transcription approaches the reader is referred to [1].

Transcription or pitch tracking methods that employ probabilistic models related to the ones used in this work are detailed in Section 2. Other approaches related to this paper include the work by Poliner and Ellis [2], where piano note classification was performed using support vector machines (SVMs). In order to improve transcription performance, the classification output of the SVMs was fed as input to a hidden Markov model (HMM) [3] for postprocessing. The same note smoothing technique was also used in [4], where the main transcription algorithm consists of a

spectral distance measure modeling polyphonic sounds as a weighted sum of Gaussian spectral models.

A signal processing-based multiple-F0 estimation was proposed by Saito et al. in [5], which uses the inverse Fourier transform of the linear power spectrum with log-scale frequency, called *specmurt* (an anagram of cepstrum). The input log-frequency spectrum is considered to be generated by a convolution of a single pitch template with a pitch indicator function. The deconvolution of the spectrum by the pitch template results in the estimated pitch indicator function. Previous work by the authors which is used for comparative purposes includes a signal processing-based polyphonic transcription system [6] which is based on joint multiple-F0 estimation using a feature-based score function and note onset and offset detection.

In this work, a system for automatic transcription of polyphonic music is introduced, which is based on a proposed extension of the shift-invariant probabilistic latent component analysis (PLCA) [7] model. Contrary to the models in [7, 8], which use a single spectral template for all pitches from the same instrument source, this model is able to support the use of multiple pitch templates extracted from multiple sources. Using a log-frequency representation and frequency shifting, detection of notes that are non-ideally tuned, or that are produced by instruments that exhibit frequency modulations is made possible. Sparsity is also enforced in the model, in order to further constrain the transcription result and the instrument contribution in the production of pitches. Also, an intermediate result of the proposed model is a time-pitch representation which can be used for pitch content visualization of polyphonic music. Finally, a hidden Markov model-based note tracking method is employed in order to provide a smooth piano-roll transcription. The system was tested on recordings from the RWC database [9], the Disklavier dataset in [2], as well as the MIREX multi-F0 woodwind quintet [10]. A comparison was performed with various transcription methods using error metrics found in the literature. It is shown that the proposed system outperforms several state-of-the-art approaches for the same experiment. Also, it is indicated that a shift-invariant model can improve the detection of non-ideally tuned notes.

The outline of the paper is as follows. In Section 2, the PLCA and shift-invariant PLCA methods are presented, along with their applications in music transcription and relative pitch tracking. The proposed polyphonic music transcription system is introduced in Section 3. Finally, the employed dataset, metrics and transcription experiments

performed are described in Section 4, while conclusions are drawn in Section 5.

2. LATENT VARIABLE METHODS

2.1 PLCA

Probabilistic latent component analysis (PLCA) is a model for acoustic analysis developed by Smaragdis et al. [11]. It provides a probabilistic framework that is extensible as well as easy to interpret. Considering the spectrogram as a probability distribution $P(\omega, t)$, the asymmetric PLCA model can be formulated as:

$$P(\omega, t) = P(t) \sum_z P(\omega|z)P(z|t) \quad (1)$$

where $P(\omega|z)$ are the spectral templates corresponding to component z , $P(z|t)$ are the component activations through time, and $P(t)$ is the energy distribution of the spectrogram. For estimating $P(\omega|z)$ and $P(z|t)$, iterative update rules are employed, which are based on the Expectation-Maximization (EM) algorithm.

In [12], an extension of the PLCA model was proposed for polyphonic music transcription, supporting multiple spectral templates for each pitch and multiple instruments. The concept of *eigeninstruments* was introduced, which models instruments as mixtures of basic models. Sparsity was enforced on the transcription matrix and the source contribution matrix of the model by a tempering-based approach. For experiments, stored pitch templates from various synthesized instrument sounds were used. Experiments were performed on instrument pairs taken from the multi-track woodwind recording used in the MIREX multi-F0 development set [10], as well as on three J.S. Bach duets.

2.2 Shift-invariant PLCA

An extension of the basic PLCA algorithm was proposed in [7], in order to extract shifted structures in non-negative data. The shift-invariant PLCA method can be used in conjunction with log-frequency spectrograms (e.g. the constant-Q transform) in order to extract pitch. This is feasible since in log-frequency spectra the inter-harmonic spacings are the same for any periodic sounds. The shift-invariant PLCA model is defined as:

$$\begin{aligned} P(\omega, t) &= \sum_z P(z)P(\omega|z) *_{\omega} P(f, t|z) \\ &= \sum_z P(z) \sum_f P(\omega - f|z)P(f, t|z) \end{aligned} \quad (2)$$

where the spectral template $P(\omega|z)$ corresponding to component z is convolved with the pitch impulse distribution $P(f, t|z)$. $P(z)$ is the prior distribution of the components. Again, in order to estimate $P(z)$, $P(\omega|z)$, and $P(f, t|z)$, the EM algorithm can be utilized.

The shift-invariant PLCA model was used in [8] for multiple-instrument relative pitch tracking, where one pitch template is attributed to each instrument source and is shifted across log-frequency. The constant-Q transform (CQT) was used as a time/frequency representation. Since the

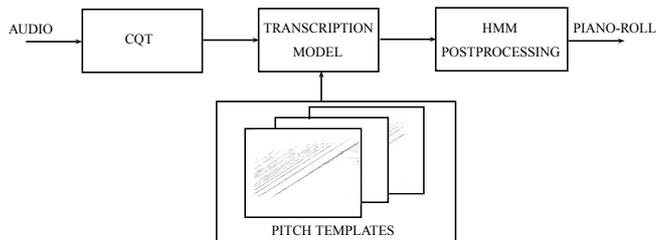


Figure 1. Diagram for the proposed polyphonic transcription system.

problem was unsupervised, additional constraints were imposed on eq. (2). Firstly, a sliding Gaussian Dirichlet prior distribution was used in the computation of $P(f, t|z)$ in order to eliminate any octave errors. In addition, in order to enforce temporal continuity, a Kalman filter type smoothing is applied to $P(f, t|z)$ at each iteration step. The method was tested on the MIREX [10] woodwind quintet using mixtures of two instruments at a time.

3. PROPOSED METHOD

The goal of the proposed transcription system is to provide a framework that supports multiple templates per pitch, in contrast to the relative pitch tracking method in [8], as well as multiple templates per musical instrument. In addition, the contribution of each instrument source is not constant for the whole recording as in [8], but is time-dependent. Also, its goal is to exploit the benefits given by a shift-invariant model coupled with a log-frequency representation, in contrast to the transcription method in [12], for detecting notes that exhibit frequency modulations and tuning changes.

In subsection 3.1, the extraction of pitch templates for various instruments is presented. The main transcription model is presented in subsection 3.2, while the HMM post-processing step is described in subsection 3.3. A diagram of the proposed transcription system is depicted in Fig. 1.

3.1 Extracting Pitch Templates

Firstly, spectral templates are extracted for various instruments, for each note, using their whole note range. Isolated note samples from three different piano types were extracted from the MAPS dataset [13] and templates from other orchestral instruments were extracted from monophonic recordings from the RWC database [9]. For extracting the note templates, the constant-Q transform (CQT) was computed [14] with spectral resolution of 120 bins per octave. Afterwards, the PLCA model of eq. (1) using only one component z was employed in order to extract the spectral template $P(\omega|z)$. In Table 1, the pitch range of each instrument used for template extraction is shown.

3.2 Transcription Model

Utilizing the extracted instrument templates and by extending the shift-invariant PLCA algorithm, a model is proposed which supports the use of multiple pitch and instrument templates in a convolutive framework, thus support-

| Instrument | Lowest note | Highest note |
|-------------|-------------|--------------|
| Cello | 26 | 81 |
| Clarinet | 50 | 89 |
| Flute | 60 | 96 |
| Guitar | 40 | 76 |
| Harpsichord | 28 | 88 |
| Oboe | 58 | 91 |
| Organ | 36 | 91 |
| Piano | 21 | 108 |
| Violin | 55 | 100 |

Table 1. MIDI note range of the instrument templates used in the proposed transcription system.

ing tuning changes and frequency modulations. By considering the input CQT spectrum as a probability distribution $P(\omega, t)$, the proposed model can be formulated as:

$$P(\omega, t) = P(t) \sum_{p,s} P(\omega|s, p) *_{\omega} P(f|p, t) P(s|p, t) P(p|t) \quad (3)$$

where $P(\omega|s, p)$ is the spectral template that belongs to instrument s and MIDI pitch $p = 21, \dots, 108$, $P(f|p, t)$ is the time-dependent impulse distribution that corresponds to pitch p , $P(s|p, t)$ is the instrument contribution for each pitch in a specific time frame, and $P(p|t)$ is the pitch probability distribution for each time frame.

By removing the convolution operator, the model of (3) can be expressed as:

$$P(\omega, t) = P(t) \sum_{p,f,s} P(\omega - f|s, p) P(f|p, t) P(s|p, t) P(p|t) \quad (4)$$

In order to only utilize each template $P(\omega|s, p)$ for detecting the specific pitch p , the convolution of $P(\omega|s, p) *_{\omega} P(f|p, t)$ takes place using an area spanning one semitone around the ideal position of p . Since 120 bins per octave are used in the CQT spectrogram, f has a length of 10.

The various parameters in (3) can be estimated using iterative update rules derived from the EM algorithm. For the expectation step the update rule is:

$$P(p, f, s|\omega, t) = \frac{P(\omega - f|s, p) P(f|p, t) P(s|p, t) P(p|t)}{\sum_{p,f,s} P(\omega - f|s, p) P(f|p, t) P(s|p, t) P(p|t)} \quad (5)$$

For the maximization step, the update equations for the proposed model are:

$$P(\omega|s, p) = \frac{\sum_{f,t} P(p, f, s|\omega + f, t) P(\omega + f, t)}{\sum_{\omega,t,f} P(p, f, s|\omega + f, t) P(\omega + f, t)} \quad (6)$$

$$P(f|p, t) = \frac{\sum_{\omega,s} P(p, f, s|\omega, t) P(\omega, t)}{\sum_{f,\omega,s} P(p, f, s|\omega, t) P(\omega, t)} \quad (7)$$

$$P(s|p, t) = \frac{\sum_{\omega,f} P(p, f, s|\omega, t) P(\omega, t)}{\sum_{s,\omega,f} P(p, f, s|\omega, t) P(\omega, t)} \quad (8)$$

$$P(p|t) = \frac{\sum_{\omega,f,s} P(p, f, s|\omega, t) P(\omega, t)}{\sum_{p,\omega,f,s} P(p, f, s|\omega, t) P(\omega, t)} \quad (9)$$

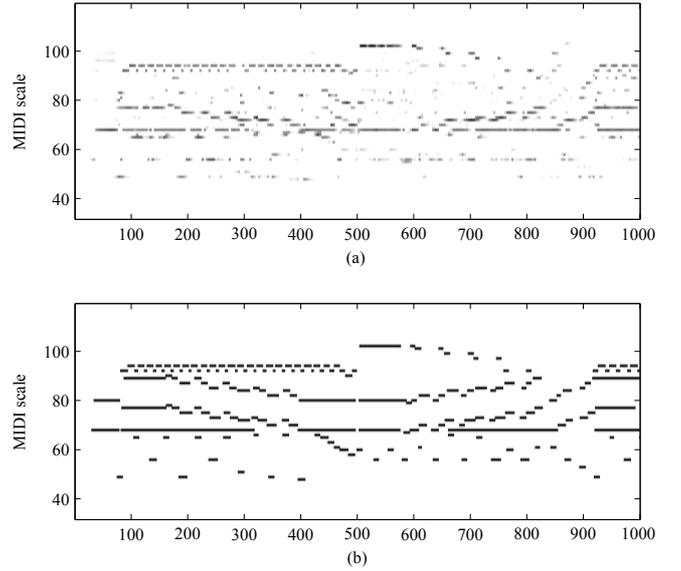


Figure 2. (a) The transcription matrix $P(p, t)$ of the first 10s of the MIREX woodwind quintet. (b) The pitch ground truth of the same recording. The abscissa corresponds to 10ms.

It should be noted that since the instrument-pitch templates have been extracted during the training stage, the update rule for the templates (6) is not used, but is included for the sake of completeness. Using these constant templates, convergence is quite fast, usually requiring 10-20 iterations. The resulting piano-roll transcription matrix and pitch matrix are respectively given by:

$$\begin{aligned} P(p, t) &= P(t) P(p|t) \\ P(f, p, t) &= P(t) P(p|t) P(f|p, t) \end{aligned} \quad (10)$$

By stacking together slices of the pitch matrix $P(f, p, t)$ for all pitch values: $P(f, t) = [P(f, 21, t) \dots P(f, 108, t)]$ we can create a time-pitch representation which can be used for visualization purposes. In $P(f, t)$, f has a length of $88 \times 10 = 880$, thus representing pitch in a 10 cent resolution. In Fig. 2, the transcription matrix $P(p, t)$ for an excerpt of the MIREX multi-F0 woodwind quintet recording can be seen, along with the corresponding pitch ground truth. Also, in Fig. 3, the time-pitch representation of an excerpt of the ‘RWC MDB-C-2001 No. 12’ (string quartet) recording can be seen, where the frequency modulations caused by vibrato are visible.

In order for the algorithm to provide as meaningful solutions as possible, sparsity is encouraged on transcription matrix $P(p|t)$, expecting that only few notes are present at a given time frame. In addition, sparsity can be enforced to matrix $P(s|p, t)$, meaning that for each pitch at a given time frame, only a few instrument sources contributes to its production. The same technique used in [12] was employed for controlling sparsity, by modifying the update equations (8) and (9):

$$P(s|p, t) = \frac{\left(\sum_{\omega,f} P(p, f, s|\omega, t) P(\omega, t) \right)^\alpha}{\sum_s \left(\sum_{\omega,f} P(p, f, s|\omega, t) P(\omega, t) \right)^\alpha} \quad (11)$$

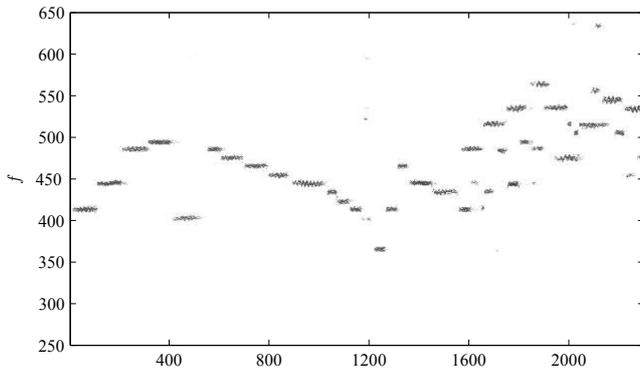


Figure 3. The time-pitch representation $P(f, t)$ of the first 23s of ‘RWC MDB-C-2001 No. 12’ (string quartet) in a 10 ms time scale.

$$P(p|t) = \frac{\left(\sum_{\omega, f, s} P(p, f, s|\omega, t)P(\omega, t)\right)^\beta}{\sum_p \left(\sum_{\omega, f, s} P(p, f, s|\omega, t)P(\omega, t)\right)^\beta} \quad (12)$$

By setting $\alpha, \beta > 1$, the entropy in matrices $P(s|p, t)$ and $P(p|t)$ is lowered and sparsity is enforced.

3.3 Postprocessing

Instead of simply thresholding $P(p, t)$ for extracting the piano-roll transcription as in [12], additional postprocessing is applied in order to perform note smoothing and tracking. Hidden Markov models (HMMs) [3] have been used in the past for note smoothing in signal processing-based transcription approaches (e.g. [2, 6]). Here, a similar approach to the HMM smoothing procedure employed in [2] is used, but modified for the probabilistic framework of the proposed transcription system.

Each pitch p is modeled by a two-state HMM, denoting pitch activity/inactivity. The hidden state sequence for each pitch is given by $Q_p = \{q_p[t]\}$. MIDI files from the RWC database [9] from the classic and jazz subgenres were employed in order to estimate the state priors $P(q_p[1])$ and the state transition matrix $P(q_p[t]|q_p[t-1])$ for each pitch p . For each pitch, the most likely state sequence is given by:

$$\hat{Q}_p = \arg \max_{q_p[t]} \prod_t P(q_p[t]|q_p[t-1])P(o_p[t]|q_p[t]) \quad (13)$$

which can be computed using the Viterbi algorithm [3]. For estimating the observation probability for each active pitch $P(o_p[t]|q_p[t] = 1)$, we use a sigmoid curve which has as input the transcription piano-roll $P(p, t)$ from the output of the transcription model:

$$P(o_p[t]|q_p[t] = 1) = \frac{1}{1 + e^{-P(p, t)}} \quad (14)$$

The result of the HMM postprocessing step is a binary piano-roll transcription which can be used for evaluation. An example of the HMM postprocessing step is given in Fig. 4, where the transcription matrix $P(p, t)$ of a piano recording from [2] is seen along with the output of the HMM smoothing.

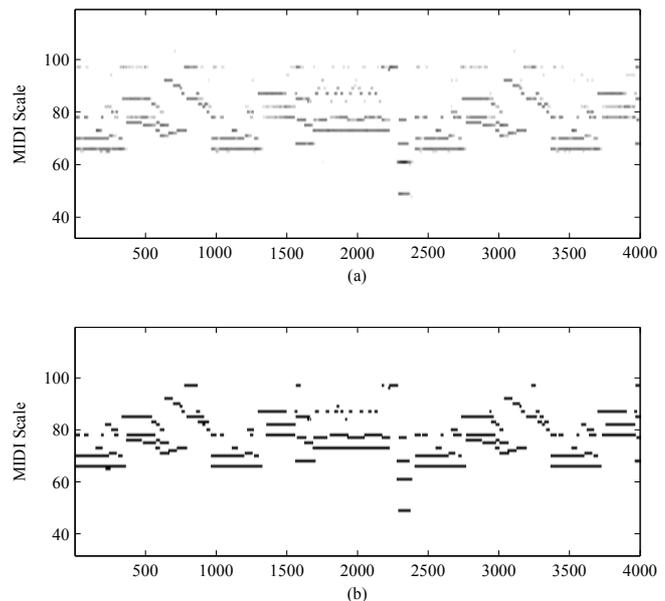


Figure 4. (a) The transcription matrix $P(p, t)$ of the first 40s of the J. Haydn Piano Sonata No.54 from the Disklavier dataset of [2] (b) The output of the HMM post-processing step (the abscissa corresponds to 10 ms).

4. EVALUATION

4.1 Datasets

For the transcription experiments, we used recordings from three different sources. Firstly, 12 excerpts from the RWC database [9] were employed, which have been widely used for evaluating transcription systems (e.g. [15, 5, 4]). The dataset contains classical and jazz music produced by piano, guitar, flute, and bowed strings, with the majority being piano. Aligned ground-truth MIDI data was created using the original non-aligned MIDI reference for the first 23 sec of each recording, using Sonic Visualiser¹.

In addition, the test dataset developed by Poliner and Ellis [2] was also used for transcription experiments. It contains 10 one-minute classical recordings from a Yamaha Disklavier grand piano, sampled at 8 kHz along with aligned MIDI ground truth. Finally, the full woodwind quintet recording from the MIREX multi-F0 development set [10] was also used for transcription experiments.

4.2 Evaluation Metrics

Several evaluation metrics are employed for the recordings used for the transcription experiments. All evaluations take place by comparing the transcribed output and the ground-truth MIDI files using a 10 ms scale, as in the MIREX multiple-F0 estimation task [10]. The first metric that is used is the overall accuracy (Acc_1) used in [2]. Also, an additional set of metrics is employed, namely the alternative accuracy measure (Acc_2), the total error (E_{tot}), the substitution error (E_{subs}), missed detection error (E_{fn}), and false alarm error (E_{fp}). Definitions for the aforementioned set of metrics can be found in [15, 5, 4].

¹ <http://www.sonivisualiser.org/>

4.3 Results

Transcription experiments using the 12 excerpts from the RWC database were performed using only piano templates, or using the full list of instrument templates shown in Table 1. Results are presented in Table 2, comparing the performance of the system with other state-of-the-art methods [6, 4, 5, 15], while in Table 3 additional metrics are used in order to compare the performance of the proposed system with the method in [6]. It can be seen that when using the piano templates, the proposed method outperforms other systems with respect to the accuracy measure Acc_2 . Also, most of the errors of the system consist of missed detections, while relatively few false alarms are detected. Concerning the signal processing-based method in [6], the improvement using Acc_2 is 0.5%, which rises to 1.0% when Acc_1 is utilized.

When using the proposed system with all instrument templates, performance is significantly lowered, although it should be stressed that the majority of the RWC recordings are produced by piano. This also indicates that having a knowledge of the instruments present can significantly improve the performance of the proposed system. It is notable that when RWC recording 10 -a string quartet- is transcribed using the all-instruments model, its Acc_2 is 82.7%, far surpassing all other methods. Concerning sparsity parameters, after experimentation, no sparsity was added to the instrument contribution matrix ($\alpha = 1$) and sparsity was only enforced on the transcription matrix ($\beta = 1.5$). It is worth mentioning that with $\beta = 1$, performance using the piano templates drops to 58.6% for Acc_2 . Also, in order to evaluate the contribution of the shift-invariant model, experiments were also performed by disabling convolution, resulting in a PLCA-based model similar to the one in [12]. In terms of Acc_2 , performance for the RWC recordings was 60.1%, which indicates that using a shift-invariant model for transcription can improve performance when non-ideally tuned recordings or when frequency modulations are considered. To the authors' knowledge, no statistical significance tests have been made for transcription, apart from the piecewise tests in the MIREX task [10]. However, given the fact that transcription evaluations actually take place using 10 ms frames, even a small accuracy change can be shown to be statistically significant, using a method like [16].

Results using the 10 piano recordings from [2] are shown in Table 4, compared with results from other approaches reported in [2] and the method in [6]. For this experiment, only piano templates were used in the proposed system. Again, it is shown that the proposed system outperforms all other methods - compared to the one in [2], improvement is 1.1% with respect to Acc_1 . It should be stressed also that the training set for the method of [2] used data from the same source as the test set. When compared with [6], the performance improvement is much larger compared to the RWC recordings (about 10.6%). This can be attributed to the much faster tempo of the pieces in [2], since the method in [6] is more suited to slower tempo due to the onset/offset detections performed, tending to accumulate transcription errors in cases of rapidly changing notes. Additional met-

| Data | Proposed (piano) | Proposed (all) | [6] | [4] | [5] | [15] |
|------|------------------|----------------|-------|-------|-------|-------|
| 1 | 64.3% | 58.3% | 60.0% | 63.5% | 59.0% | 64.2% |
| 2 | 70.5% | 61.4% | 73.6% | 72.1% | 63.9% | 62.2% |
| 3 | 70.3% | 53.8% | 62.5% | 58.6% | 51.3% | 63.8% |
| 4 | 67.0% | 63.4% | 65.2% | 79.4% | 68.1% | 77.9% |
| 5 | 66.9% | 55.4% | 53.4% | 55.6% | 67.0% | 75.2% |
| 6 | 71.7% | 73.3% | 76.1% | 70.3% | 77.5% | 81.2% |
| 7 | 67.0% | 55.9% | 68.5% | 49.3% | 57.0% | 70.9% |
| 8 | 67.7% | 51.4% | 60.1% | 64.3% | 63.6% | 63.2% |
| 9 | 51.9% | 48.8% | 50.3% | 50.6% | 44.9% | 43.2% |
| 10 | 55.3% | 82.7% | 72.4% | 55.9% | 48.9% | 48.1% |
| 11 | 57.1% | 54.2% | 56.2% | 51.1% | 37.0% | 37.6% |
| 12 | 30.4% | 26.8% | 36.6% | 38.0% | 35.8% | 27.5% |
| Mean | 61.7% | 57.1% | 61.2% | 59.1% | 56.2% | 59.6% |

Table 2. Transcription results (Acc_2) for the 12 RWC recordings compared with other approaches.

| Method | Acc_1 | Acc_2 | E_{tot} | E_{subs} | E_{fn} | E_{fp} |
|--------------|---------|---------|-----------|------------|----------|----------|
| Proposed (p) | 60.8% | 61.7% | 38.3% | 8.9% | 19.6% | 9.8% |
| Proposed (a) | 54.8% | 57.1% | 42.9% | 11.5% | 24.4% | 7.0% |
| [6] | 59.8% | 61.2% | 38.8% | 7.3% | 24.8% | 6.7% |

Table 3. Transcription error metrics for the 12 RWC recordings using piano only (p) or all templates (a), compared with the approach in [6].

rics for the recordings of [2] are included in Table 5.

Finally, results using the proposed system are shown using the MIREX multi-F0 woodwind quintet [10] in Table 6. The MIREX recording is available in 5 instrument tracks; although results using pairs of these tracks have been reported in [8, 12], to the authors' knowledge no results using the complete 5-instrument recording have been published. Using the full instrument templates matrix, performance of the proposed system is 48.1% using Acc_2 , while it is 39.0% using the system in [6]. Again, the reduced performance of [6] can be attributed to the fast tempo of the recording (a part of which is depicted in Fig. 2).

5. CONCLUSIONS

In this work, a system for automatic music transcription using a model based on shift-invariant probabilistic latent component analysis techniques was proposed. The main contribution of the paper is a transcription model that is able to support multiple instrument and pitch templates and is able to detect notes produced without ideal tuning or exhibiting frequency modulations. The system was tested on recordings from several sources, where it was shown to outperform other state-of-the-art transcription techniques using several error metrics. The system architecture makes it suitable for instrument-specific transcription applications. Also, a by-product of the system is a time-pitch representation that can also be used for pitch content visualization. Selected transcription examples are available online², along with the original excerpts for comparison.

² <http://www.eecs.qmul.ac.uk/~emmanouilb/transcription.html>

| Method | Proposed | [6] | [2] | [17] |
|---------|----------|-------|-------|-------|
| Acc_1 | 57.6% | 47.0% | 56.5% | 41.2% |

Table 4. Mean transcription results (Acc_1) for the piano recordings in [2] compared with other approaches.

| Method | Acc_1 | Acc_2 | E_{tot} | E_{subs} | E_{fn} | E_{fp} |
|----------|---------|---------|-----------|------------|----------|----------|
| Proposed | 57.6% | 56.7% | 43.3% | 10.9% | 16.9% | 15.5% |
| [6] | 47.0% | 47.2% | 52.8% | 10.7% | 33.6% | 8.5% |

Table 5. Transcription error metrics for the piano recordings in [2] compared with the approach in [6].

Since it was indicated that system performance can be improved by utilizing knowledge of the instruments present in the recording, instrument identification techniques will be incorporated in future versions of the system. Finally, future research will focus on producing templates for the attack, transient, sustain, and release states of the produced notes of each instrument and incorporate such formulation into the proposed model, in an effort to further reduce the number of missed detections.

Acknowledgments

E. Benetos is supported by a Westfield Trust PhD Studentship (QMUL). We would like to thank the late Graham Grindlay from Columbia University for providing part of the MIREX recording annotation.

6. REFERENCES

- [1] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*, 2nd ed. New York: Springer-Verlag, 2006.
- [2] G. Poliner and D. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP J. Advances in Signal Processing*, no. 8, pp. 154–162, Jan. 2007.
- [3] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [4] F. Cañadas-Quesada, N. Ruiz-Reyes, P. V. Candéas, J. J. Carabias-Orti, and S. Maldonado, "A multiple-F0 estimation approach based on Gaussian spectral modelling for polyphonic music transcription," *J. New Music Research*, vol. 39, no. 1, pp. 93–107, Apr. 2010.
- [5] S. Saito, H. Kameoka, K. Takahashi, T. Nishimoto, and S. Sagayama, "Specmurt analysis of polyphonic music signals," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 3, pp. 639–650, Mar. 2008.
- [6] E. Benetos and S. Dixon, "Polyphonic music transcription using note onset and offset detection," in *IEEE Int. Conf. Audio, Speech and Signal Processing*, Prague, Czech Republic, May 2011.
- [7] P. Smaragdis, B. Raj, and M. Shashanka, "Sparse and shift-invariant feature extraction from non-negative data," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Las Vegas, USA, Apr. 2008, pp. 2069–2072.
- [8] G. Mysore and P. Smaragdis, "Relative pitch estimation of multiple instruments," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, Apr. 2009, pp. 313–316.
- [9] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: music genre database and musical instrument sound database," in *Int. Conf. Music Information Retrieval*, Oct. 2003.
- [10] "Music Information Retrieval Evaluation eXchange (MIREX)." [Online]. Available: <http://music-ir.org/mirexwiki/>
- [11] P. Smaragdis, B. Raj, and M. Shashanka, "A probabilistic latent variable model for acoustic modeling," in *Neural Information Processing Systems Workshop*, Whistler, Canada, Dec. 2006.
- [12] G. Grindlay and D. Ellis, "A probabilistic subspace model for multi-instrument polyphonic transcription," in *11th Int. Society for Music Information Retrieval Conf.*, Utrecht, Netherlands, Aug. 2010, pp. 21–26.
- [13] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, Aug. 2010.
- [14] C. Schörkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in *7th Sound and Music Computing Conf.*, Barcelona, Spain, Jul. 2010.
- [15] H. Kameoka, T. Nishimoto, and S. Sagayama, "A multipitch analyzer based on harmonic temporal structured clustering," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 982–994, Mar. 2007.
- [16] I. Guyon, J. Makhoul, R. Schwartz, and V. Vapnik, "What size test set gives good error estimates?" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 52–64, Jan. 1998.
- [17] M. Ryyänen and A. Klapuri, "Polyphonic music transcription using note event modeling," in *2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, Oct. 2005, pp. 319–322.

| Method | Acc_1 | Acc_2 | E_{tot} | E_{subs} | E_{fn} | E_{fp} |
|----------|---------|---------|-----------|------------|----------|----------|
| Proposed | 41.9% | 48.1% | 51.9% | 23.6% | 21.9% | 6.4% |
| [6] | 33.8% | 39.0% | 61.0% | 28.1% | 26.4% | 6.5% |

Table 6. Transcription error metrics for the MIREX woodwind quintet compared with the approach in [6].

AUTOMATICALLY DETECTING KEY MODULATIONS IN J.S. BACH CHORALE RECORDINGS

Lesley Mearns, Emmanouil Benetos, and Simon Dixon

Centre for Digital Music, Queen Mary University of London, London E1 4NS, UK
{lesleym, emmanouilb, simond}@eecs.qmul.ac.uk

ABSTRACT

This paper describes experiments to automatically detect key and modulation in J.S. Bach chorale recordings. Transcribed audio is processed into vertical notegroups, and the groups are automatically assigned chord labels in accordance with Schönberg's definition of diatonic triads and sevenths for the 24 major and minor modes. For comparison, MIDI representations of the chorales are also processed. Hidden Markov Models (HMMs) are used to detect key and key change in the chord sequences, based upon two approaches to chord and key transition representations. Our initial hypothesis is that key and chord values which are derived from pre-eminent music theory will produce the most accurate models of key and modulation. The music theory models are therefore tested against models embodying Krumhansl's data resulting from perceptual experiments about chords and harmonic relations. We conclude that the music theory models produce better results than the perceptual data. The transcribed audio gives encouraging results, with the key detection outputs ranging from 79% to 97% of the MIDI ground truth results.

1. INTRODUCTION

Harmony, modulation and tonality are widely considered to be important indicators of individual composer and historical style [1]. However, harmony is not an exact science. A given chord sequence can imply more than one key, particularly in the absence of dominant harmony, and the precise moment of key change in diatonic modulation is difficult to demarcate precisely, due to the use of 'dual function' chords to smooth the transition between keys [1, 2]. Chords belonging to both the previous and new key may be reinterpreted to indicate the new key, a phenomenon referred to as 'revision' by Rorhmeier [3].

Thus there appears to be an incongruity in adopting a rigorous approach to harmony. However, a computational approach has advantages even for the experienced musicologist; the hidden or sub-conscious judgements of the analyst are rendered explicit, widely accepted facets of music theory or history may be systematically tested, and there is

a pedagogical benefit, in that music analysis is made accessible to a broader community of people. In this paper, we aim both to test the possibility of obtaining musicological information directly from audio, which if successful, has the potential to open up new opportunities for musicological research based on musical recordings, and to ascertain whether perceptual or music theory data is more effective in the modelling of harmony.

To the authors' knowledge, this is the first study which utilizes polyphonic music transcription for systematic musicology research. Although key detection could also be achieved using an audio-based chord detection system, thus skipping the transcription step, we believe that fully transcribing audio is more appropriate, as it provides a framework for extracting information from a music piece that is not limited to a specific MIR task. We consider that such collaborative work has exciting potential, both for the improvement of automatic transcription, and for computational musicology.

The outline of the paper is as follows: Section 2 of the paper describes the data and the transcription methods. Section 3 outlines the automatic chord recognition method. Section 4 describes the different HMMs. Section 5 evaluates the results, and Section 6 presents conclusions and ideas for future work. In Fig. 1, a diagram for the proposed key modulation detection system can be seen.

2. MUSIC TRANSCRIPTION

Twelve J.S. Bach chorales were selected for experiments from www.jsbchorales.net, which provides organ-synthesized recordings along with aligned MIDI reference files. The size of the dataset is appropriate for transcription experiments [4, 5]. The list of the chorales employed for the key detection experiments can be seen in Table 1. Sample excerpts of original and transcribed chorales are available online¹.

Firstly, the chorale recordings are transcribed into MIDI files using a modified version of the automatic transcription system that was proposed in [5]. The system is based on joint multiple-F0 estimation and note onset/offset detection. The constant-Q resonator time/frequency image (RTFI) [6] is employed due to its suitability for representing music signals. The number of bins per octave is set to 120, and the frequency range is set from 27.5 Hz (A0) to 12.5 kHz (the 3rd harmonic of C8). In order to suppress

Copyright: ©2011 Lesley Mearns, Emmanouil Benetos, and Simon Dixon. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

¹ <http://www.eecs.qmul.ac.uk/~emmanouilb/chorales.html>

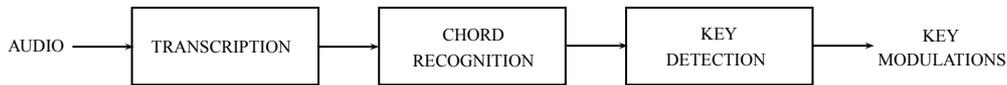


Figure 1. Key modulation detection diagram.

| | BWV | Title |
|----|-------|---|
| 1 | 1.6 | Wie schön leuchtet der Morgenstern |
| 2 | 2.6 | Ach Gott, vom Himmel sieh' darein |
| 3 | 40.6 | Schwing dich auf zu deinem Gott |
| 4 | 57.8 | Hast du denn, Liebster, dein Angesicht gänzlich verborgen |
| 5 | 85.6 | Ist Gott mein Schild und Helfersmann |
| 6 | 140.7 | Wachet auf, ruft uns die Stimme |
| 7 | 253 | Danket dem Herrn heut und allzeit |
| 8 | 271 | Herzlich tut mich verlangen |
| 9 | 359 | Werde munter, mein Gemüte |
| 10 | 360 | Werde munter, mein Gemüte |
| 11 | 414 | Danket dem Herrn, heuf und allzeit |
| 12 | 436 | Wie schön leuchtet der Morgenstern |

Table 1. The list of organ-synthesized (top) and real (bottom) chorales used for key detection experiments.

timbral information, spectral whitening is applied [4], followed by a two-stage median filtering for noise reduction.

A log-frequency pitch salience function $s[n, p]$, is extracted, along with tuning and inharmonicity parameters. Here, $p = 1, \dots, 88$ is the pitch index and n is the time frame. Onset detection is performed using a combination of a spectral flux-based and a salience function-based descriptor. For each segment defined by two consecutive onsets, multi-pitch estimation is applied in order to detect the pitches present. Pitch candidates are selected, and a pitch set score function combining several spectral and temporal features evaluates each possible pitch combination. Since the application of the transcription system concerns chorale recordings, the pitch range was limited to C2-A#6 and the maximum polyphony level was restricted to 4 voices. The pitch candidate set that maximizes the score function is selected as the pitch estimate for the current frame. Finally, note offset detection is also performed using HMMs trained on MIDI data from the RWC database [7]. Since the recordings are synthesized, tempo is constant and it can be computed using the onset detection functions from [5]. The estimated pitches in the time frames between two beats are averaged, resulting in a series of chords per beat. Transcription accuracy is 33.1% using the measure of [5], which however also takes into account note durations, hence the low value. An example of the transcription output of BWV 2.6 ‘Ach Gott, vom Himmel sieh’ darein’ is given in Fig. 2.

3. CHORD RECOGNITION

Transcribed audio, and for comparison, ground truth MIDI files, are segmented into a series of vertical notegroups according to onset times. Every new rhythmic value prompts the creation of a new vertical notegroup, so that notes which occur simultaneously or overlap in time are grouped. The pitch values within a group are converted to pitch classes 0

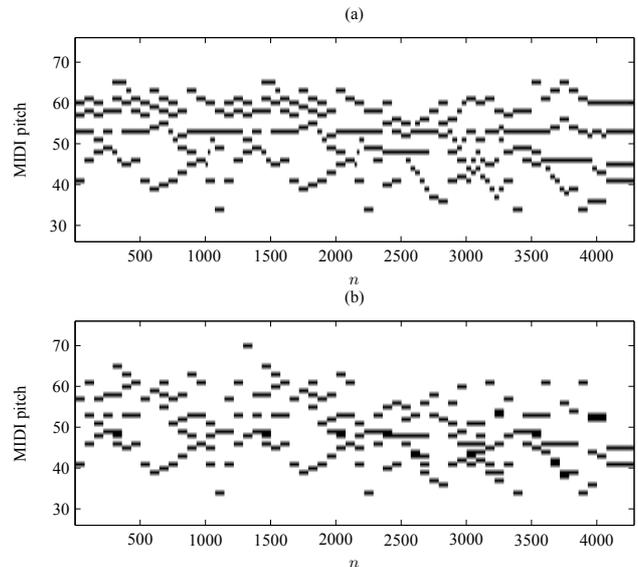


Figure 2. (a) The pitch ground-truth of BWV 2.6 ‘Ach Gott, vom Himmel sieh’ darein’. (b) The transcription output of the same recording. The abscissa corresponds to 10 ms frames.

to 11, (0=C, 1=C# etc), and repeated tones are removed in order from the bass note to create a unique ordered set. For example, MIDI pitches $\{53,57,60,65\}$, (bass, tenor, alto, soprano) would become pitch classes $\{5,9,0,5\}$ (modulo 12), which would become unique set $\{5,9,0\}$. The Bach chorales most commonly have a harmonic rhythm, (i.e. rate of harmonic change), of a crotchet beat, consequently for these experiments the vertical notegroups are organized into higher level groups which contain all of the notes present within this timing division. Thus, if the four notes of MIDI pitch $\{53,57,60,65\}$ occurred at a metrical position of 1, but the MIDI note of pitch 65 (soprano voice) gave way to the seventh on the quaver offbeat, (metrical position 1.5), to MIDI pitch 63, the complete set of pitch classes within the crotchet beat would be $\{5,9,0,3\}$.

The notegroups are classified using a chord dictionary of templates expressed as ordered sets of pitch classes, (e.g. a C Major chord is $\{0,4,7\}$). No metrical, durational, or other type of weights are attached to the tones in the notegroup. All tones, including those occurring on offbeats, are equally operative as a possible part of the harmony. The approach is deliberate in order to capture elaborated seventh chords where the seventh note is introduced on the offbeat but is still an integral part of the harmony [8].

The chord matching process undergoes a series of iterations to find the template or templates that most closely match the presented notegroup in terms of edit distance. An exact match, (edit distance 0), would be for example a root position triad (e.g. $\{0,4,7\}$). An unordered exact

match, (edit distance 0.5), would be an inverted chord (e.g. {4,7,0}). The process continues, adding 1 for each insertion or deletion, up to a maximum edit distance of 2. If a match has not been found at this stage, the offbeat notes are removed from the group, and the match process is repeated with the set of notes which occurred on the beat. Due to the requirement of the HMM for a discrete sequence of chord symbols, groups of tones returning more than one possible chord classification are reduced to a single chord choice firstly by preferring root position chords, secondly by context matching with near neighbours, (two chords in either direction), and finally by random choice.

To measure the competence of the chord labelling process, the automatically generated chord sequences are compared to hand annotated sequences. Due to the laboriousness of hand annotation, half of the files in the set have been annotated with ground truth chord sequences. Each pair of chord index values in the sequences is compared, and a basic difference measure is calculated by counting the number of matches. The final counts are normalised, resulting in a proportional measure of matched or mismatched values between the two files (Table. 2). If two index values differ, the Levenshtein distance is calculated for the two pitch class sets represented as strings, to find out the degree of difference between the pitch class sets. Many of the index value mismatches found are in fact extremely close pitch class set matches, for example, {t, 2, 5} compared to {t, 2, 5, 9}, (t=10, e=11), generating a Levenshtein difference of 1. The Levenshtein distances calculated for each file are summed and normalised by the length of sequence to produce a combined measure of accuracy and distance.

| BWV | Transcribed Audio | | Ground Truth Midi | |
|-------------|-------------------|-------------|-------------------|-------------|
| | Match | Levenshtein | Match | Levenshtein |
| 1.6: | 0.45 | 1.20 | 0.86 | 0.30 |
| 2.6: | 0.60 | 0.70 | 0.88 | 0.22 |
| 40.6: | 0.55 | 0.95 | 0.83 | 0.28 |
| 57.8: | 0.56 | 0.81 | 0.82 | 0.35 |
| 253: | 0.55 | 0.75 | 0.83 | 0.35 |
| 436: | 0.63 | 0.61 | 0.88 | 0.21 |
| Totals Avg: | 0.56 | 0.64 | 0.85 | 0.15 |

Table 2. Chord match results for transcribed audio and ground truth MIDI against hand annotations.

A greater quantity of label mismatches are found with the transcribed files than the ground truth MIDI files, depicting some of the pitch and timing errors resulting from the transcription process. Total chord mismatches between the transcribed data and the hand annotated data (i.e. where there are no pitches in common between the two pitch class sets), indicate an error in timing or quantisation. The greatest difficulty posed to the chord algorithm by the transcribed data however is the frequent presence of diads rather than triads in the groups. Resolving a diad correctly is not straightforward; if the diad is a third apart, this could imply either the upper or lower portion of a triad, equally, a diad a fifth apart could be either a major or a minor triad, a problem also encountered by Pardo [9]. The transcription algorithm

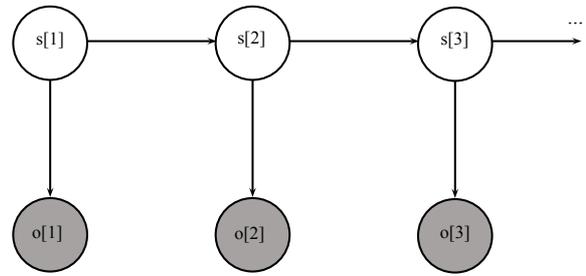


Figure 3. Graphical structure of the employed HMM for key modulation detection.

has a low false alarm error rate and a high mis-detection rate, consequently the transcription process produces output which assists the chord method where the MIDI data poses problems; groups with suspended 9th and 13th notes, or other notegroups containing complex chord tones which are not defined in the chord dictionary, are captured from the transcribed data as simple triads whereas the MIDI data may result in a ‘no chord’ value. Complex chords such as 9ths and 13ths are less adaptable to the pitch class set match approach due to the fact that internal tones must be omitted from such chords to fit with four part harmony. Overall, the average accuracy levels for the ground truth files are in the upper range of accuracy results reported by Pardo [9]. The transcribed audio achieves an average of 65% correct of the ground truth result.

4. KEY MODULATION DETECTION

4.1 Hidden Markov Models

Key change detection is performed using a set of HMMs [10]. The observation sequence $O = \{o[n]\}$, $n = 1, \dots, N$ is given by the output of the chord recognition algorithm in the previous section. The observation matrix (**B**) therefore defines the likelihood of a key given a chord. Likewise, the hidden state sequence which represents keys is given by $S = \{s[n]\}$. Each HMM has a key transition matrix $\mathbf{A} = P(s[n]|s[n-1])$. There are two dimensions, of size 24×24 , (representing the 12 major and 12 minor keys) which defines the probability of making a transition from one key to another. For a given chord sequence, the most likely key sequence is given by:

$$\hat{S} = \arg \max_{s[n]} \prod_n P(s[n]|s[n-1])P(o[n]|s[n]) \quad (1)$$

which can be estimated using the Viterbi algorithm [10]. In Fig. 3, the graphical structure of the employed HMM model is shown.

4.2 Model Definitions

Five observation matrices (**B**) and four key transition (**A**) matrices are compared in total. Three of the observation matrices are derived from music theory, and are designed to represent and test Schönberg’s theory with regard to the chord membership of the 24 major and minor modes [2]. Two further observation matrices use data from Krumhansl’s

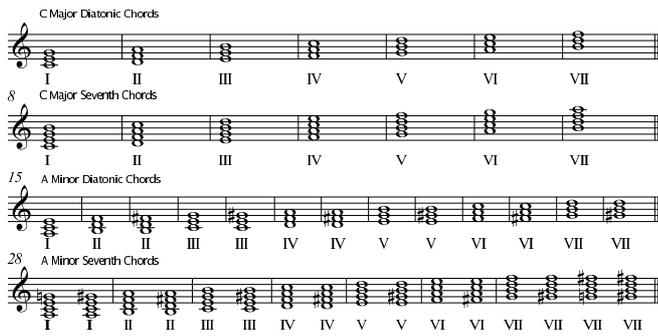


Figure 4. Diatonic chords for major and minor mode

perceptual experiments [11]. The four different versions of the key transition matrix (defined in section 4.4) are used in conjunction with all five of the observation matrices.

4.3 Observation Matrices

4.3.1 Music Theory Models

The extent to which a chord infers a key is modelled heuristically in the music theory observation matrices. The intention is to logically produce a set of musically plausible chord rankings per key across the full range of chords observed. The diatonic chords of a key are all indicative of the home key, however progressions containing chords II or IV with V or V7 are strongly indicative of the home key because they would have to be chromatically altered to imply a different key [1]. Similarly, the tonic triad, although it could be a member of several keys, tends to be prominent in the establishment of a tonal centre. Such chords may therefore be expected to rank highly compared to the lower values achieved by less characteristic chords. The relationship and interdependencies of individual tones, chords, and keys to human cognitive processing of tonality is not well understood. Consequently, to arrive at a score for a chord in relation to a key, points are given for both tone and chord properties. These include, points for each constituent tone per scale degree membership, partial points for ambiguous scale degree membership (i.e. 6th and 7th degrees in the minor key), for tonic chord status, and for being defined as a diatonic chord for the key by Schönberg. The points are then summed to give a total score for the chord in that key context.

Two of the Schönberg observation matrices symbolise the complete set of major, minor, diminished and augmented triads plus a ‘no chord’ value, resulting in a total of 49 possible chord symbols. The two matrices are weighted differently, inspired by Parncutt’s psychoacoustical work suggesting that chords are heard as having singular identities which are prior to the constituent pitches [12]. Matrix *BSchCh* therefore assigns double points to the diatonic chord as whole and gives single points for individual tones, whereas *BSchP*, therefore gives double points to constituent tones and single points for diatonic chord status. The precise rules and values used are listed in Table 3.

For example, the chord rating for a ‘C Major’ triad in the

| Feature | BSchP | BSchCh |
|--------------------------|-------|--------|
| Diatonic chord | 1 | 2 |
| Scale degree | 2 | 1 |
| Dim/aug scale degree | 1 | 0.5 |
| Ambig scaledegree | 1 | 0.5 |
| Dim/aug ambigscaledegree | 0.5 | 0.25 |
| Tonicchord | 1 | 1 |

Table 3. Rules for Schönberg observation matrix.

key of ‘C Major’ for *BSchP* would be as follows:

- C,E,G, three diatonic scale degrees = 2+2+2
- C,E,G, tonic triad = +1,
- C,E,G is listed by Schönberg as one of the diatonic triads = +1,
- Chord total = 8.

The third observation matrix *BSch7* symbolises the full set of triads and seventh chords elucidated by Schönberg [2] resulting in 22 chord definitions for the major key, and 30 chords for the minor key. (Please see Fig. 4.) The disparity in chord quantity is due to the optional raising of the 6th and 7th degree in the minor mode. A total number of 132 unique pitch class sets plus a ‘no chord’ value are therefore defined, bringing the total number of possible chord observations to 133.

The values assigned to each chord in the *BSch7* model are the same as those used for *BSchP*. In this model, the value for the dominant seventh of ‘C Major’ would be:

- G,B,D,F, four diatonic scale degrees = 2+2+2+2
- G,B,D,F, is listed by Schönberg as one of the diatonic sevenths for C Major = +1,
- Chord total = 9.

The dominant seventh chord thus is the highest signifier in the matrix for its key, satisfactorily articulating common practice in tonal harmony.

4.3.2 Music Perception Models

An HMM has been used previously to infer the overall key of a piece using Krumhansl’s perceptual data, specifically the correlations between harmonic hierarchies as a representation of key distance, and harmonic hierarchy chord ratings, which are used to populate the key transition matrix and the observation matrix respectively [13, 11]. However, many of Krumhansl’s chord ratings appear to contradict music theory. For example, in the C Major context, all of the twelve major triads, irrespective of which tone is the root, are rated as inferring the key of C Major more highly than any of the diatonic chords belonging to the key of C Major which are minor or diminished in profile. The data seems to suggest that in human perception, any major chord is more indicative of any major key, than the diatonic chords which make up that key, because it sounds major. From the perspective of music theory and common

| Chord ↓ | Key Context | | Chord ↓ | Key Context | | Chord ↓ | Key Context | |
|-----------|-------------|---------|-----------|-------------|---------|-----------|-------------|---------|
| | C Major | C Minor | | C Major | C Minor | | C Major | C Minor |
| C Maj | 6.6 (I) | 5.30 | C min | 3.75 | 5.90 | C dim | 3.27 | 3.93 |
| C#/Db Maj | 4.71 | 4.11 | C#/Db min | 2.59 | 3.08 | C#/Db dim | 2.70 | 2.84 |
| D Maj | 4.60 | 3.83 | D min | 3.12 | 3.25 | D dim | 2.59 | 3.43 |
| D#/Eb Maj | 4.31 | 4.14 | D#/Eb min | 2.18 | 3.50 | D#/Eb dim | 2.79 | 3.42 |
| E Maj | 4.64 | 3.99 | E min | 2.76 | 3.33 | E dim | 2.64 | 3.51 |
| F Maj | 5.59 | 4.41 | F min | 3.19 | 4.60 | F dim | 2.54 | 3.41 |
| F#/Gb Maj | 4.36 | 3.92 | F#/Gb min | 2.13 | 2.98 | F#/Gb dim | 3.25 | 3.91 |
| G Maj | 5.33 | 4.38 | G min | 2.68 | 3.48 | G dim | 2.58 | 3.16 |
| G#/Ab Maj | 5.01 | 4.45 | G#/Ab min | 2.61 | 3.53 | G#/Ab dim | 2.36 | 3.17 |
| A Maj | 4.64 | 3.69 | A min | 3.62 | 3.78 | A dim | 3.35 | 4.10 |
| Bb Maj | 4.73 | 4.22 | Bb min | 2.56 | 3.13 | Bb dim | 2.38 | 3.10 |
| B Maj | 4.67 | 3.85 | B min | 2.76 | 3.14 | B dim | 2.64 | 3.18 |

Table 4. Krumhansl ratings of chords in harmonic-hierarchy experiments.

compositional practice, the data is counterintuitive and one could expect inconsistent results when used with common practice musical works.

The perceptual observation matrices symbolise the same chord set as the previously described triad based Schönberg models. The four triad based models therefore process identical chord sequences, allowing a direct comparison of the models based on music theory against those based on perceptual data.

The first matrix *BKrumOrig* is formulated using Krumhansl’s chord ratings (Table 4) as per previous work by Noland [13], with the slight difference that all of Krumhansl’s chord data is used without modification. In the absence of data for augmented triads, these plus the ‘no chord’ value are given a uniform low value of 1.0. As an experiment, a second observation matrix *BKrumMod* is also created, in which the apparently contradictory values for minor chords in the major key context which are part of the key, are swapped with the major chord values which are not part of the key. For example, in the ‘C Major’ context, the values for the ‘D Major’ chord are swapped with the value for the ‘D Minor’ (chord II), ‘E Major’ with ‘E Minor’ (chord III), ‘A Major’ with ‘A Minor’ (chord VI), and ‘B Major’ with ‘B Diminished’ (chord VII). Performing this swap leads to disproportionately high values for the remaining major chords which also belie the home key without a parallel minor or diminished chord with which to exchange the rating. Such chords have 1 subtracted from their rating value to bring the data more in line with the swapped changes, for example the chord rating of 4.36 for ‘F# Major’ becomes 3.36. The values for minor chords in the minor key context in this model are left unmodified.

4.4 Key Transition Matrices

Four different versions of the key transition matrix are formalized and used for all five of the observation matrices. The first matrix *ANeutral* is neutral, so that a move to any key is equally likely. The second transition matrix *AKrum* features Krumhansl’s correlations between key profiles summed with 1 [11], similar to previous work by Noland [13]. The third and fourth matrices, referred to as *ASchEq*, and *ASchNL* respectively, are implementations of Schönberg’s table of key circles, in which seven circles of increasing key distance from a given tonic are delineated [2]. Using pitch

class set representations there are six unique circles only, the seventh containing the enharmonically equivalent keys of previous circles. Therefore the *ASchEq* subtracts an equal value of 0.25 for each key circle, commencing with an upper boundary of 2.0, and moving through the relative minor and then each successive circle, ending on the 6th circle. The *ASchNL* implementation uses an exponentially decreasing value, halving the deducted value for each circle. In *ASchNL* therefore, the numeric distance between the first circle and the sixth circle is smaller than the distance between the same two circles in the *ASchEq* matrix. For all key transition matrices except the neutral matrix, the central diagonal is slightly weighted, to give a small preference to stay in the current key. These values were determined empirically.

5. EVALUATION

5.1 Metrics

To provide a rigorous measure of accuracy of the outputs of the HMMs, each key value in the output sequences is compared to the corresponding hand-annotated key, and an error rate (*Err*), distance measure (*Dist*), measure of modulation concurrency (*Conc*), and modulation percentage (*Mods*) are calculated. Given N_{diff} the number of differences between output key and hand annotated key, N_{len} the length of the sequence, N_{cmod} the number of concurrent modulations, N_{hmod} the number of hand annotated modulations, and N_{omod} the number of modulations in the output, *Err*, *Conc* and *Mods* are defined as:

$$Err = \frac{N_{diff}}{N_{len}}, \quad Conc = \frac{N_{cmod}}{N_{hmod}}, \quad Mods = \frac{N_{omod}}{N_{hmod}} \quad (2)$$

The distance value *Dist* captures both the number of differences and the extent of each difference relative to the circle of fifths when two key values are found to conflict. For example, the distance value for a key with another key on the same circle, i.e. its dominant, subdominant, or relative minor, is 1 whereas a key difference two fifths apart on the circle of fifths (in either direction) would result in a difference value of 2, and so on. *Conc* refers to whether the HMM sequence changes key at precisely the same moment as the hand annotated sequence, regardless of whether the actual key change matches or not. Finally, *Mods* shows

| B Matrix → A Matrix ↓ | BSchP | | | | BSchCh | | | | BKrumOrig | | | | BKrumMod | | | |
|--------------------------|-------------|------|-------|-------|-------------|------|--------------|--------|-----------|------|-------|--------|----------|------|-------|-------|
| | Err | Dist | Conc | Mods | Err | Dist | Conc | Mods | Err | Dist | Conc | Mods | Err | Dist | Conc | Mods |
| Transcribed | | | | | | | | | | | | | | | | |
| ANeutral | 0.35 | 0.74 | 11.55 | 51.09 | 0.27 | 0.45 | 25.23 | 109.42 | 0.28 | 0.42 | 4.76 | 18.89 | 0.32 | 0.51 | 2.68 | 9.68 |
| AKrum | 0.26 | 0.42 | 22.31 | 78.63 | 0.30 | 0.54 | 37.58 | 132.59 | 0.30 | 0.47 | 7.82 | 52.98 | 0.31 | 0.47 | 2.68 | 33.63 |
| ASchEq | 0.26 | 0.41 | 23.54 | 87.38 | 0.31 | 0.56 | 36.07 | 124.84 | 0.31 | 0.53 | 7.82 | 53.67 | 0.30 | 0.52 | 6.50 | 34.26 |
| ASchNonLin | 0.26 | 0.39 | 28.40 | 81.72 | 0.30 | 0.47 | 33.86 | 118.57 | 0.31 | 0.53 | 7.82 | 56.31 | 0.31 | 0.54 | 5.80 | 33.00 |
| Ground Truth Midi | | | | | | | | | | | | | | | | |
| ANeutral | 0.31 | 0.45 | 9.25 | 38.26 | 0.27 | 0.45 | 24.59 | 85.26 | 0.22 | 0.37 | 8.45 | 31.72 | 0.33 | 0.53 | 5.01 | 22.87 |
| AKrum | 0.23 | 0.33 | 33.01 | 87.25 | 0.20 | 0.34 | 46.03 | 120.84 | 0.28 | 0.40 | 15.66 | 87.24 | 0.26 | 0.35 | 13.31 | 59.34 |
| ASchEq | 0.21 | 0.32 | 32.81 | 85.66 | 0.21 | 0.31 | 43.05 | 109.18 | 0.27 | 0.35 | 15.66 | 109.18 | 0.25 | 0.33 | 16.72 | 52.68 |
| ASchNonLin | 0.21 | 0.30 | 29.38 | 72.47 | 0.20 | 0.30 | 38.54 | 113.79 | 0.26 | 0.36 | 15.66 | 83.70 | 0.28 | 0.36 | 17.06 | 55.52 |

Table 5. Key data: error average, distance value for key differences average, percentage of modulation timing match, number of modulations as a percentage of hand annotated number of modulations.

| | ASchbEq / BSch7 | | | AKrum / BSch7 | | | ANeutral / BSch7 | | |
|----|-----------------|------|------|---------------|------|------|------------------|------|------|
| | BWV | Err | Dist | BWV | Err | Dist | BWV | Err | Dist |
| 1 | 1.6 | 0.18 | 0.20 | 1.6 | 0.09 | 0.09 | 1.6 | 0.11 | 0.11 |
| 2 | 414 | 0.20 | 0.25 | 414 | 0.20 | 0.28 | 414 | 0.23 | 0.32 |
| 3 | 253 | 0.23 | 0.70 | 140.7 | 0.21 | 0.23 | 359 | 0.27 | 0.50 |
| 4 | 436 | 0.25 | 0.27 | 253 | 0.23 | 0.70 | 360 | 0.28 | 0.38 |
| 5 | 140.7 | 0.27 | 0.29 | 360 | 0.23 | 0.30 | 140.7 | 0.29 | 0.31 |
| 6 | 360 | 0.33 | 0.44 | 436 | 0.27 | 0.30 | 436 | 0.33 | 0.36 |
| 7 | 359 | 0.34 | 0.39 | 359 | 0.36 | 0.50 | 253 | 0.38 | 0.78 |
| 8 | 57.8 | 0.35 | 0.46 | 57.8 | 0.39 | 0.50 | 271 | 0.41 | 0.80 |
| 9 | 271 | 0.42 | 0.88 | 271 | 0.42 | 0.77 | 57.8 | 0.44 | 0.87 |
| 10 | 85.6 | 0.45 | 0.46 | 85.6 | 0.45 | 0.48 | 2.6 | 0.45 | 0.60 |
| 11 | 2.6 | 0.55 | 0.67 | 2.6 | 0.60 | 0.67 | 85.6 | 0.48 | 0.66 |
| 12 | 40.6 | 0.78 | 1.08 | 40.6 | 0.80 | 1.14 | 40.6 | 0.69 | 1.16 |

Table 7. Chorales ordered by error rate and distance using transcribed audio and Sch7 models.

Hand annotated key and harmony labels for the mid bars of BWV 40.6:

| (Hand) | C | C | F | g | g | A | A | A | d | d | d | d | d | d | d | d |
|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| AKrum | C | C | F | g | g | G | D | D | D | d | d | d | F | F | F | g |
| ANeut | Bb | Bb | Bb | Bb | A | A | A | A | d | d | d | d | d | d | g | g |
| ASchEq | C | C | F | Bb | G | A | A | A | D | C | d | F | F | F | g | g |
| ASchNL | Bb | Bb | Bb | Bb | Bb | d | d | d | d | d | d | F | F | F | g | g |

Figure 6. Mid bars of BWV 40.6 ‘Schwing dich auf zu deinem Gott’ with HMM key outputs per transition matrix for BSch7 with hand annotated key and harmony labels.

the top of the list, with BWV 1.6 (in the key of F Major throughout), disclosing the least errors for almost every model. The three minor key chorales in the file set, BWV 85.6, 2.6, and 40.6, consistently show the greatest number of errors for all of the data sets.

The fragmentation of key sequence outputs identifies areas of harmonic complexity within the chorales. The mid section of BWV 40.6, (Fig. 6), exemplifies the difficulty of identifying a single key or exact point of key change in transitory sections; bar 9 implies several keys, and the end of bar 10 cadences in A, but is a secondary dominant of the home key of d minor.

6. CONCLUSIONS

This paper has presented an approach to key detection and key modulation using automatic chord classification of transcribed audio and ground truth MIDI data. A set of HMMs were explored using perceptual data and values calculated to represent formal music theory. Although the transcription error rate is quite high, key error rates for the audio recordings are only slightly higher compared to the key error rates for the ground-truth MIDI. Also, the key error rates are slightly higher for transcribed data using the triadic models, but the complex chord HMM exhibits remarkable alignment of results for both transcribed audio and MIDI data, suggesting that the quality of the transcribed chorales is of sufficiently high quality for the task. The music theory models were shown to outperform the perceptual data, with much of the variation between the models evincing the subtle and often ambiguous nature of musical harmony. Alignment of key boundaries is low overall with the HMM, due to the absence of phrase information, however the music theory observation matrix *BSchCh* showed a consistently better result for key change concurrence. Results are considered promising for the use of automatic transcription research in computational musicology. By combing key outputs with chord sequences, functional harmony could be obtained for the chorales measures of modulatory frequency and complexity could be derived.

Future work aims to improve the automatic chord recognition method to be able to classify complex chords and tone groups containing non-chord tones by identifying structural tones. Prior knowledge of key and harmony could also be used to improve the output of a transcription process; for example, initially transcribing the data, obtaining harmony information, and subsequently re-transcribing the data utilising this knowledge. For music research the combination of transcription and a musicology system could facilitate the analysis of large corpuses of audio data with the potential for some exciting discoveries about music.

Acknowledgments

Lesley Means is supported by an EPSRC DTA studentship. Emmanouil Benetos is supported by a Westfield Trust PhD Studentship (Queen Mary, University of London).

7. REFERENCES

- [1] W. Piston and I. Ekeland, *Harmony*. W. W. Norton & Company, 1983.
- [2] A. Schönberg, *Theory of Harmony*. University of California Press, 1911.
- [3] M. Rohrmeier, “Modelling dynamics of key induction in harmony progressions,” in *SMC 2007*, 2007.
- [4] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*, 2nd ed. New York: Springer-Verlag, 2006.
- [5] E. Benetos and S. Dixon, “Polyphonic music transcription using note onset and offset detection,” in *IEEE International Conference on Audio, Speech and Signal Processing*, Prague, Czech Republic, May 2011.
- [6] R. Zhou, “Feature extraction of musical content for automatic music transcription,” Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, Oct. 2006.
- [7] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC music database: music genre database and musical instrument sound database,” in *Int. Conf. Music Information Retrieval*, Oct. 2003.
- [8] C. H. Kitson, *Elementary Harmony*. Oxford University Press, 1920.
- [9] B. Pardo and W. Birmingham, “Algorithms for chordal analysis,” *Computer Music Journal*, vol. 26, no. 2, pp. 22–49, Summer 2002.
- [10] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [11] C. L. Krumhansl, Ed., *Cognitive Foundations of Musical Pitch*, 1st ed. Oxford University Press, 1990.
- [12] R. Parncutt, *Harmony: a psychoacoustical approach*. Springer-Verlag, 1989.
- [13] K. Noland, “Computational tonality estimation: signal processing and hidden Markov models,” Ph.D. dissertation, Queen Mary University of London, UK, Mar. 2009.

A SURVEY OF RAAGA RECOGNITION TECHNIQUES AND IMPROVEMENTS TO THE STATE-OF-THE-ART

Gopala Krishna Koduri

IIT Hyderabad

koduri@research.iiit.ac.in

Sankalp Gulati

IIT Bombay

sankalpg@ee.iitb.ac.in

Preeti Rao

IIT Bombay

prao@ee.iitb.ac.in

ABSTRACT

Raaga is the spine of Indian classical music. It is the single most crucial element of the melodic framework on which the music of the subcontinent thrives. Naturally, automatic raaga recognition is an important step in computational musicology as far as Indian music is considered. It has several applications like indexing Indian music, automatic note transcription, comparing, classifying and recommending tunes, and teaching to mention a few. Simply put, it is the first logical step in the process of creating computational methods for Indian classical music. In this work, we investigate the properties of a raaga and the natural process by which people identify the raaga. We survey the past raaga recognition techniques correlating them with human techniques, in both north Indian (Hindustani) and south Indian (Carnatic) music systems. We identify the main drawbacks and propose minor, but multiple improvements to the state-of-the-art raaga recognition technique.

1. INTRODUCTION

Geekie [1] very briefly summarizes the importance of raaga recognition for Indian music and its applications in music information retrieval in general. Raaga recognition is primarily approached as determining the scale used in composing a tune. However the raaga contains more information which is lost if it is dealt with western methods such as this. This information plays a very central role in the perception of Indian classical music.

In this work, we shortly discuss various properties of a raaga and the way the trained musicians recognize it using cues from the properties of a raaga. Further, we present a brief survey of various methods used by researchers based on such well defined rules of a raaga. We identify shortcomings in those methods and then, we present our system addressing a few of them. We discuss and compare it with the previous systems. We hope this work would be of help to Indian and non-Indian readers in understanding various properties of the raaga for computational purposes or otherwise.

Our work primarily concerns with Carnatic music, but most of the discussion applies to Hindustani music as well,

Copyright: ©2011 Gopala Krishna Koduri et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

unless mentioned otherwise.

2. PROPERTIES OF A RAAGA

Matanga, in his epic treatise Brihaddeshi, defines raaga as "that which colors the mind of good through a specific *swara*¹ and *varna* (literally color) or through a type of *dhvani* (sound)" [2]. A technically insightful definition is given by Chordia [3] and Krishnaswamy [4, 5]. It says, "Raaga is a collection of melodic atoms and a technique for developing them. These melodic atoms are sequences of notes that are inflected with various micro pitch alterations and articulated with expressive sense of timing. Longer musical phrases are built by knitting these melodic atoms together". The notion that raaga is not just a sequence of notes is important in understanding it, for developing a representation of raaga for computational purposes. For a westerner, the notion of raaga as put by Harold S Powers might be helpful in understanding what a raaga is. It says, "A raaga is not a tune, nor is it a 'modal' scale, but rather a continuum with scale and tune as its extremes" [6].

Not surprisingly, these definitions coincide in what they try to convey. Though a given raaga has characteristic melodic phrases, they are neither limited nor given. It can be understood from the fact that even after a given raaga is used to tune numerous compositions by various people, it is always possible that a new tune can be composed using that raaga. On the other hand, it is neither just a set of notes, because different raagas have same set of notes yet sound very different. This is due to various properties of raaga like the order of the notes (called *arohana/avarohana* which mean ascending/descending patterns respectively), the way they are intonated using various movements (called *gamakas*), their relative position, strength and duration of notes (i.e., various functions of swaras). We'll now see these various aspects of raaga in detail.

2.1 Arohana and Avarohana: The Ascending and Descending Progressions of a Raaga

Typically a raaga is represented using the ascending (*arohana*) and descending (*avarohana*) progressions of notes. There are certain observations (or rules) that are necessary while reciting a raaga with regards to the transitions between notes. The transitions generally occur with a note that is near to the current note in *arohana/avarohana*. There are several other heuristics characteristic of a raagas aro-

¹ Swara refers to one the seven notes in the octave.

hana and avarohana, which are not always strictly followed. We have heard multiple viewpoints about such heuristics.

2.2 Gamakas

There is a reason why Indian classical music does not have a strongly followed notation system like the western classical tradition. Consider a note, a fixed frequency value. The rapid oscillatory movement about the note is one of the several forms of movements, which are together called as gamakas. Another form of gamaka involves making a sliding movement from one note to another. Like this, there are number of ways to move around or move between the notes. There are various ways to group these movements. But the most accepted classification speaks of 15 types of gamakas [7, 8]. Apart from gamakas, there are *alankaras* (ornaments) which are patterns of note sequences which beautified and instilled some kind of feeling when listened to.

Owing to the gamakas tremendous influence on how a tune sounds, they are often considered the soul of Indian classical music. Though gamakas are used in both Carnatic and Hindustani [9], the pattern of usage is very distinct. We would like to highlight the point that gamakas are not just decorative items or embellishments, they are very essential constituents of a raaga.

2.3 Characteristic Phrases

Each raaga, just like it has a set of notes, also has few characteristic phrases. These phrases are said to be very crucial for conveying the *bhava* or the feeling of the raaga. Typically in a concert, the artist starts with singing these phrases. These are the main clues for the listeners to identify what raaga it is.

2.4 Various Roles Played by the Notes

In a given raaga, not all the swaras play the same role. As very well put by [10], just like various checkers in the game of chess, various notes in the raaga have different functions. Certain swaras are said to be important than the rest. These swaras bring out the mood of the raaga. These are called *Jeeva* swaras. The musical phrases are built around the Jeeva swaras. The note which occurs at the beginning of the melodic phrases is referred to as *Graha* swara. *Nyasa* swaras are those notes which appear at the end of such musical phrases. *Dirgha* swaras are notes that are prolonged. A swara that occurs relatively frequently is called *Amsa* swara, and that which is sparingly used is called *Alpa* swara. Though two given raagas have the same set of constituent notes, the functionality of the constituent swaras can be very different, leading to a different feeling altogether.

In addition to these above discussed properties, Hindustani classical music also emphasizes the time and season, a raaga should be used in. They seem less relevant in Carnatic music today.

That said, a raaga is an evolution phenomenon. It continually takes place over time; no existing raaga was perceived the way it is today. The properties which enhance

the characteristic nature of a raaga are retained and others are done away with. This process happens continually over decades and centuries. The raaga takes its shape and sets a unique mood depending on these properties.

Now, we'll discuss the way listener and a musician identify a raaga from a composition.

3. HOW DO PEOPLE IDENTIFY A RAAGA

Though there are no rules of thumb in identifying a raaga, usually there are two procedures by which people get to know the raaga from a composition. It normally depends on whether the person is a trained musician or a rasika, the non-trained but knowledgeable person. People who have not much knowledge of raagas cannot identify them unless they memorize the compositions and their raagas.

3.1 Non-trained Person or The *Rasika's* Way

In a nutshell, the procedure followed by a rasika typically involves correlating two tunes based on how similar they sound. Years of listening to tunes composed in various raagas gives a listener enough exposure. A new tune is juxtaposed with the known ones and is classified depending on how similar it sounds to a previous tune. This similarity can arise from a number of factors - the rules in transition between notes imposed by arohana and avarohana, characteristic phrases, usage-pattern of few notes and gamakas.

This method depends a lot on the cognitive abilities of a person. Without enough previous exposure, it is not feasible for a person to attempt identifying a raaga. There is a note worthy observation in this method. Though the people cannot express in a concrete manner what a raaga is, they are still able to identify it. This very fact hints at a possible classifier, that can be trained with enough data for each raaga.

3.2 The Trained Musician's Way

A musician tries to find few characteristic phrases of the raaga. These are called *pakads* in Hindustani music and *swara sancharas* in Carnatic music. If the musician finds these phrase(s) in the tune being played, the raaga is immediately identified. But at times these phrases might not be found or, are too vague. In this case, the musicians play the tune on an instrument (imaginary or otherwise) and identify the swaras being used. They observe the gamakas used on these swaras, locations of various notes within the music phrases and the transitions between swaras. They use these clues to arrive at a raaga.

This method seems to use almost all the characteristics a raaga has. It looks more programmatic in its structure and implementation. If the current music technology can afford to derive various low level features which can be used to identify such clues, the same procedure can be implemented computationally with almost perfect results!

These methods used by trained musicians and non-trained listeners are both important which are to be used for implementing a raaga recognition system. As we will see, the existing systems try to mimic them as much as possible.

4. AUTOMATIC RAAGA RECOGNITION

In this section, we present a survey of previous systems which dealt with raaga recognition. We discuss the different approaches, implementations and results. In the next section, we outline the shortcomings of these systems. Later we present our raaga recognition method which seeks to address some of these.

Past approaches to computer-based raaga recognition have based themselves on the properties of raga such as pitch class distributions or pitch sequence information as captured by note bi-grams or HMMs (Hidden Markov models) or swara intonations. The needed inputs are obtained by the pitch tracking of usually monophonic audio signals of an unaccompanied instrument or voice, optionally followed by a step of note segmentation and labeling.

4.1 Scale Matching

Sridhar and Geetha [11] have followed an approach where the scale used in the tune is estimated, and compared with the scales in the database. The raaga corresponding to that scale in database which matches which the estimated scale is output by the system. Their test data consisted of 30 tunes in 3 raagas sung by 4 artists. They use harmonic product spectrum algorithm [12] to extract the pitch. The tonic is manually fed. The other frequencies in the scale are marked down based on the respective ratio with the tonic. The set of notes which are used are matched against several sets of notes stored in the database for various raagas. Note that this is not the same as pitch-class profile. Here, the comparison is between the scale intervals, and not the pitch-class distribution. The results thus obtained are shown in Figure 1. A similar approach based on detecting the swaras used in arohana and avarohana to find the raaga is presented by Shetty and Achary [13].

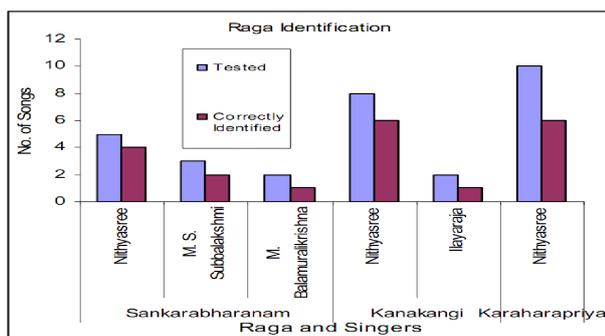


Figure 1. Results of Sridhar & Geeta’s raaga identification method [11]

4.2 Statistical Modeling and Pakad Matching

Sahasrabudde and Upadhye [14] modeled the raaga as a finite automaton based on the rules set by the properties of each raaga. This idea is used to generate a number of note sequences for a raga composition, which were technically correct and indistinguishable from human compositions. Inspired by this, Pandey et al. [15], used HMMs to

| Raaga | Test Samples | Accurately Identified | Accuracy |
|--------------|--------------|-----------------------|----------|
| Yaman Kalyan | 15 | 12 | 80% |
| Bhupali | 16 | 12 | 75% |
| Total | 31 | 24 | 77% |

Table 1. Results of Plain Raaga Identification in Tansen [15]

| Raaga | Test Samples | Accurately Identified | Accuracy |
|--------------|--------------|-----------------------|----------|
| Yaman Kalyan | 15 | 12 | 80% |
| Bhupali | 16 | 15 | 94% |
| Total | 31 | 27 | 87% |

Table 2. Results of Raaga Identification with Pakad Matching in Tansen [15]

capture the note transitions in their “Tansen” raaga recognition system. The rules to form a melodic sequence for a given raaga are well defined and the number of notes is finite. So, HMM model of a raaga proved to be good at capturing those rules in note transitions engraved by arohana and avarohana patterns of the respective raaga. They have complemented this system with scores obtained from two pakad matching modules. In one such module, pakad is identified with substring matching algorithm. In the other one, it is identified by counting the occurrences of n-grams of frequencies in the pakad.

The other important contributions of [15] include two heuristics to improve the transcription of Indian classical music - the hill peak heuristic and the note duration heuristic. Unlike western music, Indian music has a lot of micro tonal variations which makes even monophonic note transcription a challenging problem. The two heuristics try to get through these micro tonal fluctuations in attaining a better transcription. The hill peak heuristic says that a significant change in the slope or the sign reversal in slope is closely associated with the presence of a note. The note duration heuristic assumes that a note is played for at least a certain constant span of time.

Tansen is built to classify two raagas. The results are shown below. Table 1 shows the results obtained using HMM models. Table 2 shows the results obtained by complementing HMM models with pakad matching.

The central idea in this approach, which is to model a raaga as HMM, was also used in [16]. The same idea is used in an attempt to automatically generate Hindustani classical music [17], but with less success.

4.3 Pitch-class Profiles and Note Bi-grams

Chordia [3] has used the pitch class profiles and the bi-grams of pitches to classify raagas. The dataset used in his system consists of 72 minutes of monophonic instrumental (sarod) data in 17 raagas played by a single artist. The HPS algorithm is used to extract the pitch. Note onsets are detected by observing the sudden changes in phase

| Classifier | Accuracy |
|-----------------------|----------|
| Multi Variate Normal | 94% |
| FFNN | 75% |
| K-NN Classifier | 67% |
| Tree-based Classifier | 50% |

Table 3. Raaga recognition accuracies with various classifiers in Chordia’s system [3]

and amplitude in the signal spectrum. Then, the pitch-class profiles and the bi-grams are calculated. It is shown that bi-grams are useful in discriminating the raagas with the same scale. He uses several classifiers combined with dimensionality reduction techniques. Using just the pitch class profiles, the system achieves an accuracy of 75%. Using only the bi-grams of pitches, the accuracy is 82%. Best accuracy of 94% is achieved using a multivariate Normal classifier, together with principal components analysis (PCA) to reduce the feature vector size from 144 (bi-grams) + 12 (pitch profile) to 50. Performance across classifiers is shown in Table 3.

We have run a preliminary experiment to see if pitch-class distribution features (PCDs) that have been reported to be showing high accuracy in identifying Hindustani raagas [3] work in the context of Carnatic raagas. Figure 2 shows distributions of PCDs of two tunes each for two raagas. It is quite evident from the figure that there is a strong intra-raaga consistency and inter-raaga variance.

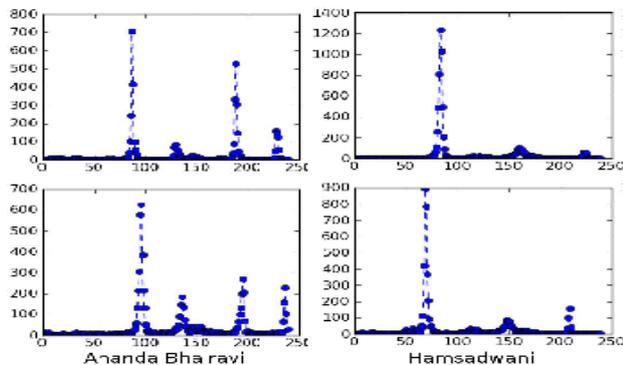


Figure 2. Pitch-class distribution for four tunes each in two raagas. Plots on the left side correspond to Ananda Bhairavi and those on the right side correspond to Hamsadwani. X-axis denotes the bin numbers. Y-axis denotes the count for the respective bin.

4.4 Swara Intonation

It is often said that, in Indian classical music, a swarasthana² does not correspond to a fixed frequency value (with its octave equivalents). It is a region [18]. So, although two raagas share the same scale, the precise intonation of specific notes can vary significantly. Belle *et al* [19] have used this

² the position held by a particular scale degree note with respect to the tonic

clue to differentiate raagas that share the same scale intervals. They evaluated the system on 10 tunes, with 4 raagas evenly distributed in 2 distinct scale groups. They showed that the use of swara intonation features improved upon the accuracies achieved with straightforward pitch class distributions.

In all the above attempts, we see that most of the approaches which we have mentioned in the beginning of the section, have been made use of. Ideally speaking those approaches should be capable of building a perfect raaga recognition system. In the the following section, we identify few problems that make this task difficult.

5. PROBLEMS THAT NEED TO BE ADDRESSED

5.1 Gamakas and Pitch Extraction for Carnatic Music

An appropriate pitch extraction module is that which can accurately represent the gamakas. It has not been a severe problem for the classification systems that were not depending on gamakas of a note for classification. If there is such a pitch extraction system in place, gamakas can be used as an additional feature to improve the accuracies of existing systems. Gamakas assume a major role when the number of raaga classes is high in the dataset.

5.2 Skipping tonic detection

The manually implemented tonic (the base frequency of the instrument/singer) identification stage needs to be eliminated if possible. Since the tonic identification itself involves some amount of error, this could adversely impact the performance of a raaga recognition system. Neither the Carnatic nor Hindustani systems adhere to any absolute tonic frequency, therefore it makes sense to build a system that can ignore the absolute location of the tonic.

5.3 Resolution of pitch-classes

Though 12 bins for pitch-class profiles look ideal to the Western eye, we hypothesize that a more continuous model can capture more relevant information related to Indian classical music. Dividing an octave into n bins where $n > 12$ can help us model the distribution with better resolution. Gamakas (the micro tonal variations) play a vital role in the perception of Indian music, and this has been confirmed by several accomplished artists. The transitions involved in a gamaka and the notes through which its trajectory passes are two factors that need to be captured. We hypothesize that this information can be obtained, at least partially, using a higher number of bins for the first-order pitch distribution.

5.4 A Comprehensive Dataset

The previous datasets which are used for testing have several problems. In Tansen, and the work by Sridhar and Geeta, the datasets had as few as 2 or 3 raagas. The dataset used by Chordia has all the data played on a single instrument by a single artist. The test datasets were constrained to some extent by the requirement of monophonic audio

(unaccompanied melodic instrument) for reliable pitch detection. In the present work, we investigate raaga recognition performances on a more comprehensive dataset with more raaga classes with significant number of tunes in each across different artists and different compositions. This should enable us to obtain better insight into the raaga identification problem.

With these issues about the raaga recognition in mind, we have implemented a system which addresses some of the challenges described. The following sections introduces our method, and presents a detailed analysis and discussion of the results.

6. OUR METHOD

As mentioned earlier, we propose to address some of the issues described in the previous section. We have taken a diverse set of tunes to include in the dataset. The use of amply available recorded music necessitates a pitch detection method that can robustly track the melody line in the presence of polyphony. The obtained sequence of pitch values converted to cents scale (100 cents = 1 semitone) constitutes the pitch contour. The pitch contour may be used as such to obtain a pitch-class distribution. On the other hand, given the heavy presence of ornamentation in Indian music, it may help to use identified stable note segments before computing the pitch-class distribution. We investigate both approaches. Finally, a similarity measure, that is insensitive to the location of the tonic note, is used to determine the best matched raaga to a given tune based on available labeled data. Each of the aforementioned steps is detailed next.

6.1 Pitch Extraction

Pitch detection is carried out at 10 ms intervals throughout the sampled audio file using a predominant pitch detection algorithm designed to be robust to pitched accompaniment [20]. The pitch detector tracks the predominant melodic voice in polyphonic audio accurately enough to preserve fast pitch modulations. This is achieved by the combination of harmonic pattern matching with dynamic programming based smoothing. Analysis parameter settings suitable to the pitch range and type of polyphony are available via a graphical user interface thus facilitating highly accurate pitch tracking with minimal manual intervention across a wide variety of audio material. Figure 3 shows the output pitch track superimposed on the signal spectrogram for a short segment of Carnatic vocal music where the instrumental accompaniment comprised violin and mridangam (percussion instrument with tonal characteristics). While the violin usually follows the melodic line, it plays held notes in this particular segment. Low amounts of reverberation were audible as well. We observe that the detected pitch track faithfully captures the vocal melody unperturbed by interference from the accompanying instruments.

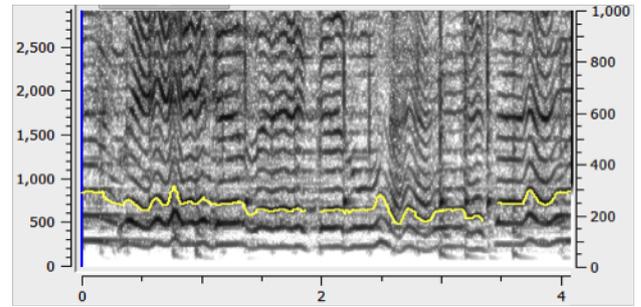


Figure 3. Screenshot from the melodic pitch extraction system of [20] showing the detected pitch superimposed on the signal spectrogram. The axis on the right indicates pitch value (Hz).

6.2 Finding the Tuning Offset

The pitch values obtained at 10 ms intervals are converted to the cents scale by assuming an equi-tempered tuning scale at 220 Hz. All the pitch values are folded into a single octave. The finely-binned histogram maximum of the deviation of the cents value from the notes of the equi-tempered 12-note grid provides us the underlying tuning offset of the audio with respect to 220 Hz. The tuning offset is applied to the pitch values to normalize the continuous pitch contour to standard 220 Hz tuning by a simple vertical shift but without any quantization to the note grid at this point.

6.3 Note Segmentation

As we observe in Figure 3, the pitch contour is continuous and marked by glides and oscillations connecting more stable pitch regions. The stable note regions too are marked by low pitch modulations. As described in Sec. 2, melodic ornamentation in Indian classical music is very diverse and elaborate. For our investigation of pitch class profiles confined to stable notes, we need to detect relatively stable note regions within the continuously varying pitch contour. The local slope of the pitch contour can be used to differentiate stable note regions from connecting glides and ornamentation.

At each time instant, the pitch value is compared with its two neighbors (i.e. 10 ms removed from it) to find the local slope in each direction. If either local slope lies below a threshold value of 15 semitones per second, the current instant is considered to belong to a stable note region. This condition is summarized by the Eq. 1.

$$(|(F(i-1) - F(i))| < \theta) \parallel (|(F(i+1) - F(i))| < \theta) \quad (1)$$

where $F(i)$ is the pitch value at the time index i and θ being the slope threshold. To put the selected threshold value in perspective, a large vibrato (spanning a 1 semitone pitch range) at 6 Hz pitch modulation frequency has a maximum slope of about 15 semitones per second. All instants where the slope does not meet this constraint are considered to belong to the ornamentation.

Finally, the pitch values in the segmented stable note regions are quantized to the nearest available note value in

the 220 Hz equi-tempered scale. This step smoothes out the minor fluctuations within intended steady notes. Figure 4 shows a continuous pitch contour with the corresponding segmented and labeled note sequence superimposed. We note several passing notes are detected which on closer examination are found to last for durations of 30 ms or more.

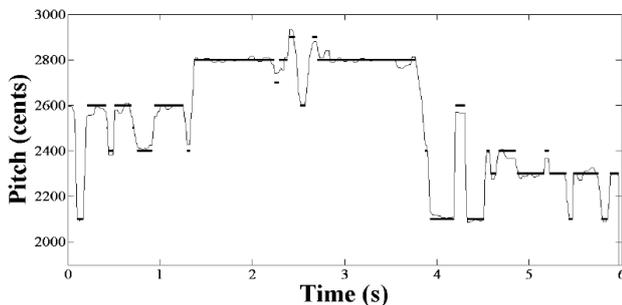


Figure 4. Note segmentation and labeling. Thin line: continuous pitch contour; Thick line: detected stable note regions.

6.4 Pitch-class Profiles

We investigate various approaches to deriving the pitch class profile. The first of two broad approaches corresponds to considering only the stable notes, segmented and labeled in the previous step. The pitch class profile is then a 12-bin histogram corresponding to the octave-folded note label values. There are two choices for weighting the note values for histogram computation. We call these P_1 and P_2 , where P_1 refers to weighting a note bin by the number of instances of the note, and P_2 refers to weighting by total duration over all instances of the note in the music piece.

A second broad approach is ignore the note segmentation step and to consider all pitches in the pitch contour irrespective of whether they correspond to stable notes or ornamentation regions. We call this P_3 . Further, the number of divisions of the octave is varied representing different levels of fineness in pitch resolution. The investigation of varying quantization intervals is motivated by the widely recognized microtonal character of Indian music.

6.5 Distance Measure

In order to compare pitch-class profiles computed from two different tunes, it is necessary that the distribution intervals are aligned in terms of the locations of corresponding scale degrees. This can be ensured by the cyclic rotation of one of the distributions to achieve alignment of its tonic note interval with that of the other distribution. Since information about the tonic note of each tune is not available a priori, we consider all possible alignments between two pitch class profiles and choose the one that matches best in terms of minimizing the distance measure. This is achieved by cyclic rotation of one of the distributions in 12 steps with computation of the distance measure at each step.

As for choosing the distance measure itself, we would like it to reflect the extent of similarity between two tunes

in terms of shared raaga characteristics. We choose the Kullback-Leibler (KL) divergence measure as a distance measure suitable for comparing distributions. Symmetry is incorporated into this measure by summing the two values as given below [19].

$$D_{KL}(P, Q) = d_{KL}(P|Q) + d_{KL}(Q|P) \quad (2)$$

$$d_{KL}(P|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad (3)$$

where i refers to the bin index in the pitch class profile, and P and Q refer to pitch class distributions of two tunes.

7. EXPERIMENT AND RESULTS

We describe a raaga classification experiment and present results on the comparative performances of the various types of pitch-class profiles for different classifier settings. A suitable dataset is constructed from commercially available CD audio recordings. To make the best use of available data, we use leave-one-out cross validation with a k-NN (k Nearest Neighbors) classifier to evaluate the performance of our system. The details of the experiment are provided next.

7.1 Dataset

There are a few observations worth mentioning in connection with the design of a test dataset for our raaga recognition system. During preliminary trials of our system, we observed a performance bias in available datasets arising from the fact that several popular compositions in Carnatic music originate in the 17th, 18th and 19th centuries. Some of these compositions sung by several artists lead to the occurrence of several sets of near identical tunes in the dataset resulting in very similar pitch profiles for supposedly different pieces of music. This prompted us to exercising due care in selecting music pieces for our test dataset. We have been careful not to include different versions of the same composition in the dataset. For instance, a tune which renders the kruti 3 nanu brOvamani chepavE is not included if a tune based on that kruti already existed in the dataset. However, since alapanas are not pre-composed, and are purely based on the artists virtuosity, we have included them. To get a bigger dataset we considered the complete Raagam-Taanam-Pallavis of various artists besides shorter krutis. This expanded the list of options from which it is possible to extract a clip to be included in the dataset. The clips were extracted from the live performances and CD recordings of 31 artists, both vocal (male and female) and instrumental (veena, violin, mandolin and saxophone) music. The dataset consisted of 170 tunes from across 10 raagas with at least 10 tunes in each raga (except Ananda Bhairavi with 9 tunes) as summarized in Table 4. The duration of each tune averages 1 minute. The tunes are converted to mono-channel, 22.05 kHz sampling rate, 16 bit PCM. The dataset can be considered very representative of the Carnatic classical music, since it includes artists spanning several decades, male and female, and all the popular instruments.

| Raaga | Total tunes | Avg. duration in seconds | Composition of Tunes |
|-----------------|-------------|--------------------------|---------------------------|
| Abheri | 11 | 61.3 | 6 vocal, 5 instrumental |
| Abhogi | 10 | 62 | 5 vocal, 5 instrumental |
| Ananda Bhairavi | 9 | 64.7 | 4 vocal, 5 instrumental |
| Arabhi | 10 | 64.9 | 8 vocal, 2 instrumental |
| Atana | 21 | 56.75 | 12 vocal, 9 instrumental |
| Begada | 17 | 61.17 | 9 vocal, 8 instrumental |
| Behag | 14 | 59.71 | 12 vocal, 2 instrumental |
| Bilahari | 13 | 61.38 | 10 vocal, 3 instrumental |
| Hamsadwani | 41 | 57.07 | 14 vocal, 27 instrumental |
| Hindolam | 24 | 60 | 15 vocal, 9 instrumental |

Table 4. Description of the dataset across 10 raagas.

| Pith-class profile | k=1 | k=3 | k=5 | k=7 |
|--|------|------|------|------|
| P_1 (12 bins, weighted by number of instances) | 55.9 | 56.5 | 57.1 | 59.4 |
| P_2 (12 bins, weighted by duration) | 71.2 | 73.5 | 76.5 | 76.5 |
| P_3 (12 bins) | 73.5 | 70 | 74.7 | 75.3 |
| P_3 (24 bins) | 72.4 | 72.9 | 75.3 | 74.1 |
| P_3 (36 bins) | 68.2 | 72.4 | 72.9 | 74.1 |
| P_3 (72 bins) | 67.7 | 68.2 | 69.4 | 68.2 |
| P_3 (240 bins) | 65.3 | 68.2 | 66.5 | 65.9 |

Table 5. Performance of weighted-k-NN classification with various pitch-class profiles

7.2 Classification Experiment

A k-NN classification framework is adopted where several values of k are tried. In a leave-one-out cross-validation experiment, each individual tune is considered a test tune in turn while all the remaining constitute the training data. The k nearest neighbors of the test tune in terms of the selected distance measure are considered to estimate the raaga label of the test tune. The distance measure used is the symmetric KL distance presented in the previous section. Since there are in all a minimum of 9 tunes per raaga, we consider values of k=1, 3, 5 and 7. Since the number of classes is high (10 raagas), it is more appropriate to consider a weighted-distance k-NN classification rather than simple voting to find the majority class. Weighted k-NN classification is described by the equations below. The chosen class is C^* ,

$$C^* = \arg \max_c \sum_i w_i \delta(c, f_i(x)) \quad (4)$$

where c is the class label (raaga identity in our case), $f_i(x)$ is the class label for the i^{th} neighbor of x and $\delta(c, f_i(x))$ is the identity function that is 1 if $f_i(x) = c$, or 0 otherwise. The weights are given by,

$$w_i = \frac{1}{d(x, y)} \quad (5)$$

where $d(x,y)$ is the symmetric KL distance between two pitch-class profiles x and y (e.g. its i^{th} neighbor).

The results in terms of percentage accuracy in raaga identification, obtained on the test dataset, appear in Table 5. Two important points emerge from the comparison of accuracies across the different types of pitch-class profiles.

For all values of k, except k=1, in the k-NN classification, we see that P_2 (the note segmented, duration weighted pitch-class profile) yields the highest accuracies. This implies that note durations play an important role in determining their relative prominence for a particular raaga realization. This is consistent with the fact that long sustained notes like dirgha swaras play a major role in characterizing a raaga than other functional notes which occur briefly in the beginning, the end or in the transitions. The benefit of note segmentation is seen in the slightly superior performance of P_2 over P_3 (12 bin). P_2 does not consider those instants that lie outside detected stable note regions. The second important point emerging from Table 5 is the decreasing classification accuracy with increasing bin resolution. Although the reverse might be expected in view of the widely held view that the specific intonation of notes within micro-intervals are a feature peculiar to a raaga, a more carefully designed, possibly unequal, division of the octave may be needed to observe this.

The overall best accuracy of 76.5%, which value is much higher than chance for the 10-way classification task, indicates the effectiveness of pitch-class profile as a feature vector for raaga identification. It is encouraging to find that a simple first order pitch distribution provides considerable information about the underlying raaga although the complete validation of this aspect can be achieved only by testing with a much larger number of raaga classes on larger dataset. Including the ornamentation regions in the pitch-class distribution did not help. As mentioned before, the gamakas play an important role in characterizing the raaga as evidenced by performance as well as listening practices followed. However, for gamakas to be effectively

exploited in automatic identification, it is necessary to represent their temporal characteristics such as the actual pitch variation with time. A first-order distribution which discards all time sequence information is quite inadequate for the task.

8. CONCLUSIONS

A brief but comprehensive introduction to the raaga and its properties is presented. Previous raaga recognition techniques are surveyed with a focus on their approach and contributions. Key aspects that need to be addressed are outlined and a method which deals with a few of them is discussed. Apart from these contributions of our work, we have also highlighted details such as the composition of the testing dataset, and provided insights into the post-processing steps involved with pitch extraction procedure for Carnatic music. This is the first work, to the best of our knowledge, that uses polyphonic audio recordings in the raaga recognition task.

The transitions in gamakas are discarded in the method explained, or are not fully utilized. A higher number of bins in the pitch distribution proved to be not necessarily useful. Future raaga recognition techniques can take into account the other properties of a raaga. Most important of these are the characteristic phrases and gamakas which suggest that temporal properties may be usefully exploited in future work.

9. REFERENCES

- [1] G. Geekie, "Carnatic ragas as music information retrieval entities," in *Proc. of ISMIR*, 2002, pp. 257–258.
- [2] P. Sharma and K. Vatsayan, *Brihaddeshi of Sri Matanga Muni*. South Asian Books, 1992.
- [3] P. Chordia and A. Rae, "Raag recognition using pitch-class and pitch-class dyad distributions," in *Proc. of ISMIR*, 2007, pp. 431–436.
- [4] A. Krishnaswamy, "Melodic atoms for transcribing carnatic music," in *Proc. of ISMIR*, 2004, pp. 345–348.
- [5] —, "Multi-Dimensional Musical Atoms in South Indian Classical Music," in *Proc. of the International Conference of Music Perception & Cognition*, 2004.
- [6] H. S. Powers, "The Background of the South Indian Raaga-System," Ph.D. dissertation, Princeton University, 1959.
- [7] S. R. Janakiraman, *Essentials of Musicology in South Indian Music*. The Indian Music Publishing House, 2008.
- [8] P. P. Narayanaswami and V. Jayaraman, 2004. [Online]. Available: <http://ibiblio.org/guruguha/ssp.htm>
- [9] Pratyush, "Analysis and Classification of Ornaments in North Indian (Hindustani) Classical Music," Master's thesis, University of Pompeu Fabra, 2010.
- [10] T. Viswanathan and M. H. Allen, *Music in South India*. Oxford University Press, 2004.
- [11] R. Sridhar and T. Geetha, "Raga identification of carnatic music for music information retrieval," *International Journal of Recent trends in Engineering*, vol. 1, no. 1, 2009, pp. 571–574.
- [12] K. Lee, "Automatic chord recognition from audio using enhanced pitch class profile," in *Proc. of the International Computer Music Conference*, 2006.
- [13] S. Shetty and K. Achary, "Raga Mining of Indian Music by Extracting Arohana-Avarohana Pattern," in *International Journal of Recent trends in Engineering*, vol. 1, no. 1. Acamey Publisher, 2009, pp. 362–366.
- [14] H. Sahasrabuddhe and R. Upadhye, "On the computational model of raag music of india," in *Workshop on AI and Music: European Conference on AI*, 1992.
- [15] G. Pandey, C. Mishra, and P. Ipe, "Tansen: A system for automatic raga identification," in *Proc. of Indian International Conference on Artificial Intelligence*, 2003, pp. 1350–1363.
- [16] M. Sinith and K. Rajeev, "Hidden Markov Model based Recognition of Musical Pattern in South Indian Classical Music," in *IEEE International Conference on Signal and Image Processing, Hubli, India*, 2006.
- [17] D. Das and M. Choudhury, "Finite State Models for Generation of Hindustani Classical Music," in *Proceedings of International Symposium on Frontiers of Research in Speech and Music*, 2005.
- [18] A. Datta, R. Sengupta, N. Dey, and D. Nag, *Experimental Analysis of Shrutis from Performances in Hindustani Music*. Scientific Research Department, ITC Sangeet Research Academy, 2006.
- [19] S. Belle, R. Joshi, and P. Rao, "Raga Identification by using Swara Intonation," *Journal of ITC Sangeet Research Academy*, 2009, vol. 23.
- [20] V. Rao and P. Rao, "Vocal melody extraction in the presence of pitched accompaniment in polyphonic music," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 8, 2010, pp. 2145–2154.

VERSION DETECTION FOR HISTORICAL MUSICAL AUTOMATA

Bernhard Niedermayer¹

¹Dept. of Computational Perception
Johannes Kepler University Linz
music@jku.at

Gerhard Widmer^{1,2}

²Austrian Research Institute for
Artificial Intelligence, Vienna
music@jku.at

Christoph Reuter³

³ Dept. of Musicology
University of Vienna
christoph.reuter@univie.ac.at

ABSTRACT

Musical automata were very popular in European homes in the pre-phonograph era, but have attracted little attention in academic research. Motivated by a specific application need, this paper proposes a first approach to the automatic detection of versions of the same piece of music played by different automata. Due to the characteristics of the instruments as well as the themes played, this task deviates considerably from cover version detection in modern pop and rock music. We therefore introduce an enhanced audio matching and comparison algorithm with two main features: (1) a new alignment cost measure – *Off-Diagonal Cost* – based on the Hough transform; and (2) a *split-and-merge strategy* that compensates for major structural differences between different versions. The system was evaluated on a test set comprising 89 recordings of historical musical automata. Results show that the new algorithm performs significantly better than the reference system based on Dynamic Time Warping and chroma features without the above-mentioned new features, and that it may work well enough to be practically useful for the intended application.

1. INTRODUCTION

Over the past 30 years, the *Phonogram Archive* of the Austrian Academy of Sciences (www.phonogrammarchiv.at) has compiled a large collection of recordings of a variety of historical musical automata (e.g., musical boxes, ‘flute clocks’, violin playing automata, barrel organs, ...).¹ Thousands of recordings have been collected, representing a large repertoire of music that was popular during certain periods of the 18th and 19th centuries – from opera arias to folk songs. Musical automata were widespread in private homes long before the invention of the phonograph. Such a collection is thus a unique source of information to study musical tastes, popular repertoire, and other musical trends during parts of the pre-phonograph era.

A major problem with the audio collection is that while the instruments themselves are relatively well documented,

¹For a report on the background of this project see http://www.phonogrammarchiv.at/Mechanical_Music/mechreal.html.

Copyright: ©2011 Bernhard Niedermayer et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the musical pieces being played are often unknown. At the same time, there is reason to believe that there will be multiple versions and interpretations of many of the songs in the collection. One way to automatically identify some of the unknown pieces would thus be to systematically search for subsets of recordings that might represent the same piece and then unify their meta-information. Given the size of the collection and the specific characteristics of the recordings (see below), this is a very difficult and tedious task.

The research described here represents a first attempt at developing audio analysis and matching technology that could help with this problem. The goal is a kind of ‘cover version’ detection: for each recording in the collection, search for other recordings that might represent (parts of) the same piece – perhaps played on an entirely different instrument, in a different key, possibly only sharing some sub-sections of the piece. Recordings of such historical music automata present new challenges to sound and music computing: apart from the sometimes extremely inharmonic sounds of the instruments (see section 3), a central problem is the often extreme ornamentation and/or arpeggiations and other asynchronies that obscure the main melody (to the extent that it is sometimes hard even for experienced listeners to recognize a song, at least at first hearing).

We present here a first pilot study that starts with a small collection of recordings of musical boxes and flute clocks, and with 3 pairs of recordings which are known to pertain to the same composition. We first experiment with fairly ‘standard’ audio matching technology (chroma features, dynamic time warping), and then, based on insights into specific problems posed by our data, develop and test an enhanced audio matching strategy (including the use of a Hough transform). Experiments show that the latter method improves results considerably and gives us reason to believe that the general problem is not unsolvable.

The remainder of this paper is organized as follows. In section 2 we give a brief overview of related work in the field of (cover) version detection. Section 3 then describes relevant characteristics of the mechanical musical instruments under consideration. The proposed version detection system is explained in Section 4, and the exact similarity measures used are defined in Section 5. An evaluation is presented in Section 6.

2. RELATED WORK

A detailed overview over current version detection systems is given in [4]. Also, an annual comparison of different al-

gorithms is carried out as part of the MIREX² contest. The evaluation is performed on two test sets, one comprising pieces of popular music, the other one consisting of performances of Chopin's mazurkas by different pianists. Best results were obtained in 2009 by [1] and [2]. Both approaches are based on chroma descriptors. [1] used cross recurrence plots based on a state space representation of two songs to rank songs according to their similarity. [2] calculated three different similarity features – two based on cross-correlation and one based on dynamic programming – in combination with three different tempo assumptions and trained a support vector machine on this data.

A completely different approach was presented in [3], where similarity estimation is not done based on audio features themselves, but on a discrete text representation. Strings are obtained from the feature sequences by clustering all chroma vectors and subsequently replacing each individual vector by the hash value assigned to the cluster it belongs to. The actual query is then performed by exploiting the open-source search engine *Lucene*³.

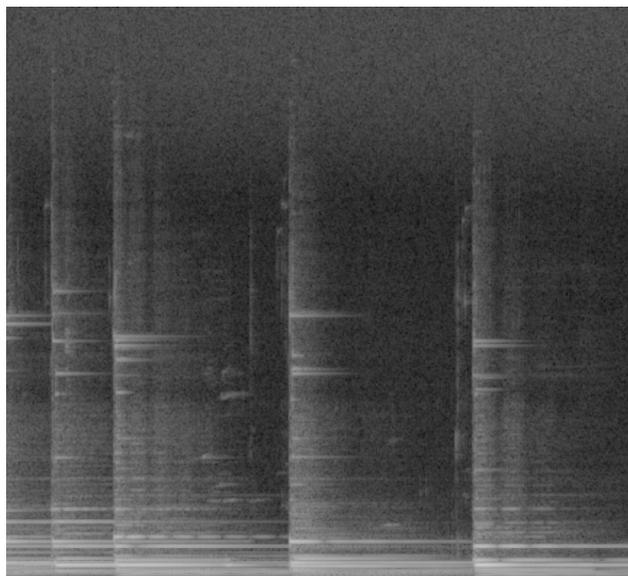
Though not directly focusing on cover version detection, also [5] is relevant to our work. Here, instead of identifying cover versions, the plausibility of audio-to-midi alignments is assessed. Several metrics were investigated, including a *relative path cost* measure that will also be employed in our matching algorithm (see Section 5).

3. ACOUSTIC CHARACTERISTICS OF MUSICAL AUTOMATA

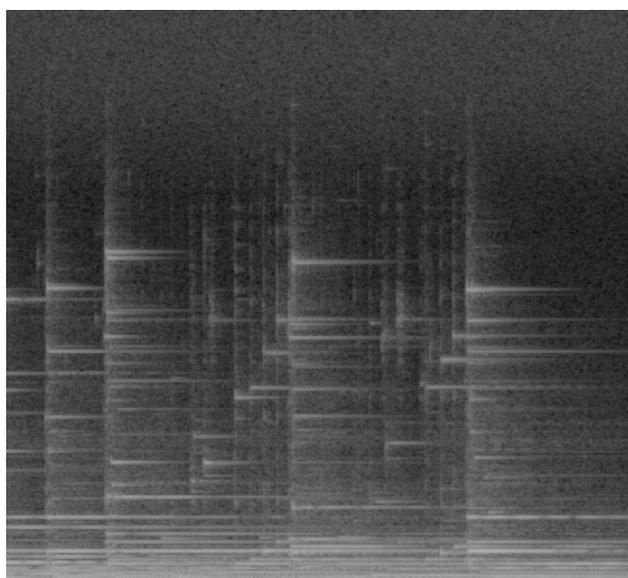
Musical automata come in wide varieties. Computer controlled pianos and similar modern instruments aside, there are musical boxes, flute clocks, violin playing automata of various kinds, barrel organs, etc. – each of them revealing individual characteristics. While flute clocks, for example, are largely consistent with the spectral envelopes one would expect from wind instruments – i.e., harmonics at the odd integer multiples of the fundamental frequency – musical boxes, where metal plates are struck, are highly inharmonic.

Another issue are different arrangements of a same theme for different automata. Besides transpositions into different keys, additional ornamentations or arpeggiations can alter the sound impression of a piece significantly. Figure 1 shows spectrograms of the ending of a theme from Mendelssohn's oratorio *Elias* as played by two different musical boxes. While in (a) only the main melody notes are played, (b) features luscious ornamentations that make it hard even for human listeners to hear the relation to the main theme.

A third challenge for a version detection system are major changes in the structure of the piece. As described in more detail in section 6 (see Table 2), when two recordings relate to the same piece, this does not necessarily mean that they both comprise the same musical sections. In our data there are samples where one musical box plays only



(a)



(b)

Figure 1. The ending of the same theme from Mendelssohn's oratorio *Elias* played by two different musical boxes.

the second half of what another one is playing. Moreover, some recordings are *potpourris* (*medleys*) of several themes (e.g., popular melodies from an opera); pairs of such medleys pertaining to the same set of themes will match only in part.

4. VERSION DETECTION

The proposed system works in two steps. First, features are extracted from the audio signals of the individual recordings. Then, a compact similarity measure is obtained, such that pairs of audio files can be ranked accordingly. In doing so, to account for transpositions and major structural changes, each piece is split into several chunks of a fixed

² MIREX: Music Information Retrieval Evaluation eXchange (organized by the IMIRSEL at the University of Illinois at Urbana-Champaign) <http://www.music-ir.org/mirex>

³ <http://lucene.apache.org>

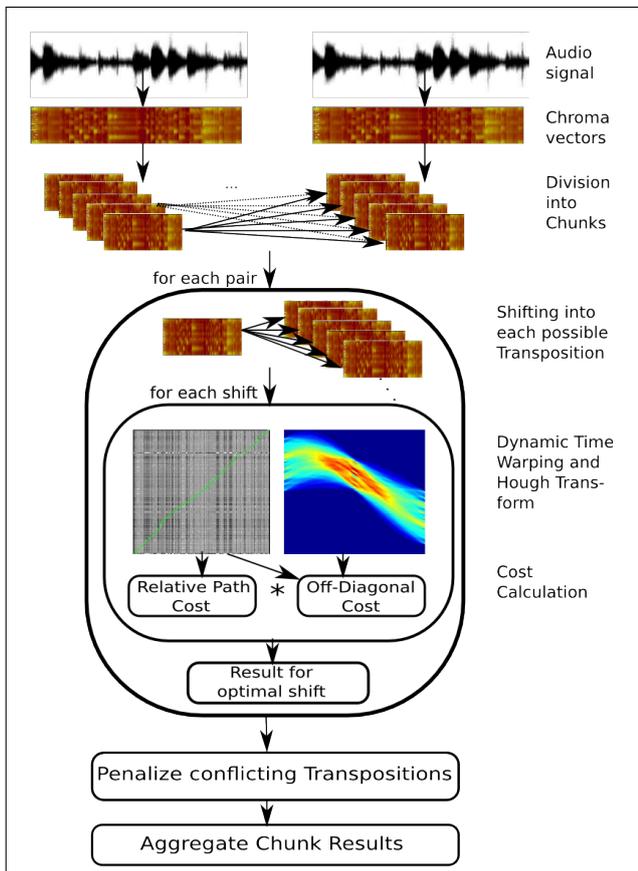


Figure 2. Calculation of the similarity measure

length which are then aligned to each other considering all possible transpositions. The final similarity measure is obtained by accumulating the fitness ratings of these alignments. An overview of the whole process is given in Figure 2.

4.1 Feature Extraction

Chroma vectors are a common feature in cover version detection or audio-to-audio alignment. They have been shown to be robust to several potentially problematic aspects, such as different instrumentation, varying degrees of polyphony, or different recording conditions. A comparative study has been presented in [6], showing that chroma vectors outperform several other features, such as MFCCs or pitch histograms, in an audio matching task.

Chroma vectors consist of a 12-dimensional vector per time frame, representing the relative energies within the individual pitch classes (i.e. C, C#, D, ...). The calculation starts by transforming the audio signal into the frequency domain. Then the energies of frequency bins corresponding to the same pitch classes are accumulated. To this end the center frequencies of all bins are folded into the same octave. The mapping onto pitch classes is then done such that a frequency can also contribute to two classes if it lies in the middle of the fundamental frequencies of two adjacent pitches. Therefore, the distance of a bin's folded center frequency \bar{f} to the fundamental f_0 of a certain pitch prototype is determined on the cent-scale as

$$d_{f_0}(\bar{f}) = 1200 * \log_2 \frac{\bar{f}}{f_0} \quad (1)$$

The energy of \bar{f} then contributes to each prototypical fundamental f_0 with a weighting of

$$w(\bar{f}, f_0) = \begin{cases} 0 & , \text{if } d_{f_0}(\bar{f}) < l \\ \cos\left(\frac{\pi}{2l} d_{f_0}(\bar{f})\right) & , \text{else} \end{cases} \quad (2)$$

In the system proposed here, l was chosen to be 75. Concerning the transform into the spectral domain, an STFT with window length 4096 and a hop size of 1024 was applied. This results in a frequency resolution fine enough to also resolve relatively low pitches and a time resolution enabling an accurate alignment between two pieces.

4.2 Transposition

A common deviation of a version of a piece from its original is a change of key. Often, the reason for these transpositions is to adapt a piece to a different lead instrument or another singer. Transpositions can also be motivated by artistic considerations, such as to give a piece a different mood.

Most similarity measures will inevitably fail if the main key has been changed between two versions of a piece. The one used here, for example, depends on the Cosine distance $d_c(A_i, B_j)$ (see equation 3 in section 4.4) that measures the error made when matching two chroma vectors A_i and B_j . The Cosine distance, however, is not robust to transpositions, i.e., shifts of one of the chroma vectors.

To compensate for possible changes of key, when computing the similarity measures between two feature sequences, each of the 12 shifts is considered. The one yielding the best result is kept and its deviation from the original key is remembered.

4.3 Segmentation

Listening to many recordings of different musical automata has revealed that the lengths of the themes played by these instruments vary significantly from less than 30 seconds up to several minutes. This can also be the case when the original piece or even the particular theme was the same. For instance, there are pairs of musical boxes where one performs only the second half of what the other one is playing.

To address this problem, the recordings were split into fragments of equal length. Instead of comparing two whole pieces, each combination of two such chunks was considered. This might seem inefficient from a computational costs point of view. However, the effects are moderate since the effort required to run the alignment algorithm (which is of complexity $O(n^2)$) and compute the similarity measure (including operations of complexity $O(n^3)$) is reduced accordingly. In addition, the processing of pairs of fragments is fully parallelizable and also, due to the fixed chunk size, the amount of memory needed is limited and independent of the actual length of the full audio recordings.

A fragment length of 25 seconds and an overlap ratio of about 50% have been found to yield good results. The overlap ratio was adapted slightly, such that the last fragments are positioned at the very end of each piece and there is no remainder left. When trying to align pairs of fragments from two pieces, all possible transpositions are considered, and different fragments from the same piece are allowed to be transposed by different numbers of semitones. Conflicts arising from this are handled later when the results from the individual fragment pairs are merged (see Section 5.3).

4.4 Alignment

To calculate a compact similarity measure, two sequences of features have to be aligned first. A well known method to perform this task is Dynamic Time Warping (DTW). Here, one starts by defining a cost function, measuring the error made when aligning a frame A_i within one feature sequence A to a frame B_j within another feature sequence B . Preliminary experiments have shown that the cosine distance d_c between two chroma vectors, defined as

$$d_c(A_i, B_j) = \frac{\langle A_i, B_j \rangle}{|A_i| * |B_j|} \quad (3)$$

yields good results.

Given the cost function, the next step is to calculate the dissimilarity matrix D , where each element $D_{i,j}$ equals $d_c(A_i, B_j)$. From this data an accumulated cost matrix C can be computed efficiently. Here, each cell $C_{i,j}$ gives the minimum cost of an alignment between the two subsequences $A_0 - A_i$ and $B_0 - B_j$. Starting from $C_{0,0} = D_{0,0}$, it is calculated iteratively by

$$C_{i,j} = \min \begin{cases} C_{i-1,j-1} + D_{i,j} \\ C_{i-1,j} + D_{i,j} \\ C_{i,j-1} + D_{i,j} \end{cases} \quad (4)$$

The alignment is finally determined by the path through D that leads to the optimal global alignment cost given by $C_{N-1,M-1}$. Backtracking the path can easily be done by remembering which of the three options in equation 4 was used in each step of the calculation of C . For a more detailed description of the Dynamic Time Warping algorithm and its properties or a possible refinement strategy, we refer the interested reader to [7] and [5] respectively.

5. SIMILARITY MEASUREMENT

Given the alignment, an intuitive similarity measure for two pieces of music would be the average cost along the alignment path. However, preliminary experiments have shown that this measure is too simple. One the one hand, it allows for insertions or deletions of notes and thus a change of melody up to certain amount, while on the other hand, it penalizes a change in the structure of a piece. In addition, although chroma vectors are relatively robust to these effects, higher average alignment costs can still be caused by changes in instrumentation or accompaniment.

5.1 Relative Path Cost

One approach to account for differences in instrumentation or accompaniment is to calculate the average cost along the alignment path *in relation to the overall average cost* \bar{D} over all $D_{i,j}$. Assuming that the two pieces of music under consideration share some similar sections, the pitch classes in the chroma feature will have similar underlying distributions. In such cases the overall average cost \bar{D} is a good baseline, estimating the average path cost of a random alignment. It can account for specifics of the two audio signals that the chroma feature is not invariant to, such as changed recording conditions, varying levels of noise, or different arrangements. In [5] a similar metric is proposed and shown to outperform others that fully rely on absolute costs along the alignment path.

5.2 Off-Diagonal Cost

An inherent property of the DTW algorithm is that it is robust to small deviations of one piece compared to another one. This is necessary in order to compensate for different performance styles or playing errors. However, the DTW algorithm's flexibility can also result in relatively low alignment costs when two melodies are compared that are really to be considered different – especially in cases where one performance contains a lot of additional ornamentation. The algorithm may decide to delete the main melody notes while matching the auxiliary notes, producing a good-looking alignment of what are really different melodies.

This undesired behavior can be detected by investigating the shape of the alignment path. Matching melodies, although with varying ornamentations, are likely to result in approximately linear paths along the main diagonal. On the other hand, different melodies which still yield low global alignment costs are characterized by major stretches and compressions of notes. This corresponds to significant horizontal or vertical segments within the alignment path. To measure this effect, we define the *Off-Diagonal Cost* as the deviation of the optimal alignment path from the best strictly linear alignment path.

A method to retrieve linear segments, known from the field of image processing, is the Hough transform [8, 9]. It transforms points $(i, j)^T$ from the image domain – in our case, the element-wise inverse of the dissimilarity matrix D – into the Hough space H , where each point $(\rho, \theta)^T$ represents a line given by θ – the angle with respect to the image domain's positive x-axis – and ρ – the distance from the origin, such that

$$\rho - i \cos \theta - j \sin \theta = 0 \quad (5)$$

A single point $(i, j)^T$ of the image domain lies on infinitely many lines. Thus, its Hough transform is a function $r_{i,j}$, following from equation 5 as

$$r_{i,j}(\theta) = i \cos \theta + j \sin \theta \quad (6)$$

$r_{i,j}$ is a sinusoid of period 2π having a magnitude of $|(i, j)^T|$ and a phase of $\arctan j/i$.

In the discrete case the Hough space is sampled and represented by an accumulator array \hat{H} of size $N \times M$. The function $r_{i,j}$ then becomes a set of corresponding cells, defined as

$$R_{i,j} = \{\hat{H}_{t,P^{-1}(h_{i,j}(\Theta(t)))} : t \in [0, N - 1]\} \quad (7)$$

where $\Theta(t)$ is the angle θ corresponding to index t and $P^{-1}(\rho)$ is the sampling index ρ resolves to.

When applying the Hough transform to the dissimilarity matrix D , for each element $(i, j)^T$, all accumulator cells in $R_{i,j}$ are increased by $D_{i,j}^{-1}$. High values within the resulting \hat{H} indicate prominent lines in the image domain. Figure 3 shows the accumulator arrays \hat{H} that result from comparing two versions of the same piece of music and two independent recordings, respectively.

In the proposed system, the size of the accumulator array was set to the size of the dissimilarity matrix D . Doing so yields fine resolutions if D is large and coarser resolutions if input data is scarce. In addition, the angle θ was restricted to have a maximum deviation from the main diagonal of $\pm 30^\circ$. Since the dominant line is assumed to be the best linear alignment path, this restricts the slopes – i.e., the tempo deviations – to reasonable values.

In summary, the Hough transform is used as a line detector. Applied on the inverse alignment costs, it finds linear segments within the dissimilarity matrix D along which the two pieces under consideration match relatively well. Such alignments allow only for an offset between the beginnings and a constant tempo change. The highest value of the accumulator array \hat{H} represents the best alignment under these constraints.

To finally calculate the Off-Diagonal cost, for each point along the alignment path as computed by the DTW algorithm, the shortest distance to the linear one is measured. These distances are then squared and averaged to obtain the final cost measure. In doing so, the first and last 5% of the paths are disregarded, so as not to penalize different offsets.

5.3 Data Merging

So far, we have proposed splitting the audio recordings into chunks to compensate for structural changes between versions of the same piece of music, described a basic method to align two such chunks, and introduced two similarity measures that indicate whether the alignment has indeed matched notes of a same melody. To finally obtain a compact similarity measure these pieces of information need to be integrated.

First, the two similarity measures need to be combined. The Relative Path cost describes the difference between feature vectors along the alignment path in comparison to a specific baseline influenced by changes in instrumentation or recording conditions between the two audio recordings. The Off-Diagonal cost, on the other hand, measures the severity of changes in rhythm or local tempo. Preliminary experiments have shown that simply taking the product of these two measures results in a meaningful aggregated matching cost.

Next, conflicts between evidence for transpositions of segments by different intervals need to be resolved. To this end, a majority vote is taken from the n most similar pairs of segments. To determine this number n , the lengths of valid alignment paths given different scenarios are considered. Let one piece be split into a chunks and the one it is compared to into b fragments, then the minimum number n_{min} of pairs needed to fully reflect an alignment path over the whole recordings along the diagonal is $\max(a, b)$. On the other hand, the maximum number n_{max} of pairs needed to cover an alignment path in the worst case – if it consists of many horizontal and vertical segments – is $a+b-1$. Therefore a reasonable number of pairs of chunks to take into consideration is chosen as $n = \alpha \max(a, b)$ with $1 \leq \alpha < 2$, depending on how much deviation from the main diagonal an overall alignment path should be allowed to exhibit.

The main idea of splitting pieces into chunks was to compensate for major structural changes, e.g. the insertion or deletion of a prominent section of a piece. A large difference in performance time would be a cue for such a modification. Therefore, instead of forcing two whole pieces to be aligned, parts of the longer recordings are allowed to be left out. To this end, the length of the shorter recording was chosen to be the determining factor, resulting in $\tilde{n} = \alpha \min(a, b)$. We set α to 1.5, to still give consideration to deviations in tempo. Experiments have shown that the number \tilde{n} of pairs taken into account outperforms the original n .

Once the main transposition interval has been obtained via voting among these \tilde{n} selected pairs of segments, deviating transposition intervals of individual pairs are penalized by multiplying the respective matching costs by a factor β . In the context of our data, $\beta = 2$ is sufficient to prevent low matching costs as a result of arbitrarily many different transpositions. The final matching cost is then obtained by averaging over the costs of the \tilde{n} most similar pairs of fragments.

6. EVALUATION

6.1 Data Corpus

The data corpus used for the evaluation comprises recordings of 89 mechanical music instruments, collected by the *Phonogram Archive* of the Austrian Academy of Sciences. About half of the pieces are played by flute clocks while the other half is performed by musical boxes. As described in Section 3, there are also significant differences in performance style and accompaniment. While some instruments only play the main melody notes, others make use of rich ornamentations.

Amongst the test data are three pairs of recordings pertaining to the same underlying piece (all of them performed by music boxes). They comprise (several) themes from Auber's opera *Fra Diavolo*, Mendelssohn's oratorio *Elias*, and Haydn's oratorio *The Creation*, respectively. These are the 'cover versions' we wish to discover in the experiment.

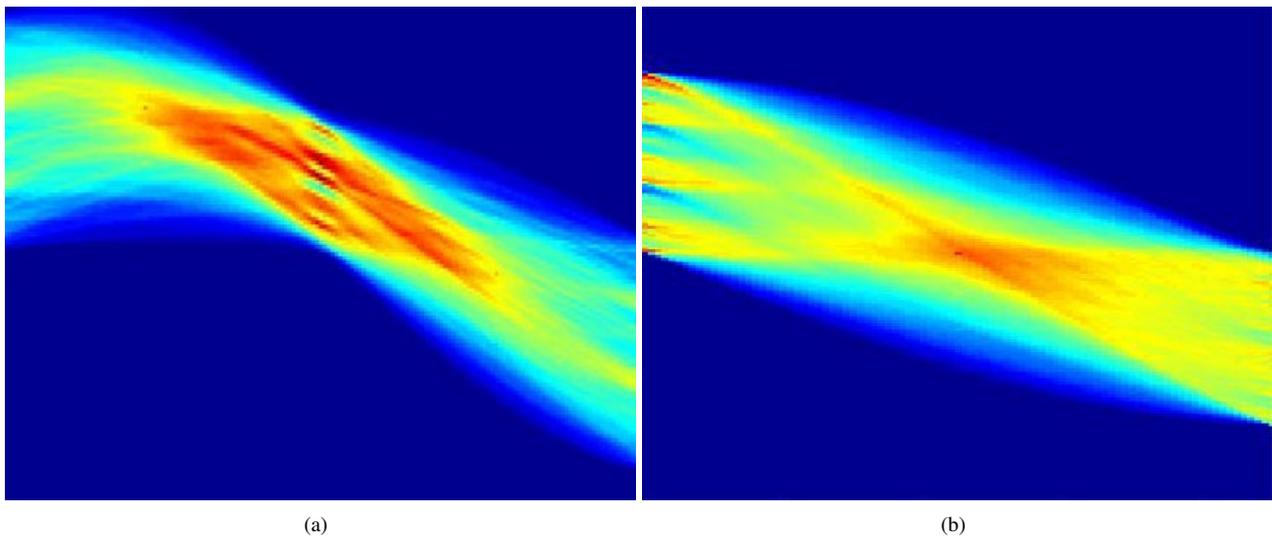


Figure 3. Hough transform of two versions of the same piece of music (a) and two independent recordings (b).

| Query Piece | Proposed System | | Standard Alg | |
|--------------|-----------------|--------|--------------|--------|
| | Ver. 1 | Ver. 2 | Ver. 1 | Ver. 2 |
| Fra Diavolo | 1 | 3 | 5 | 8 |
| Elias | 4 | 3 | 12 | 6 |
| The Creation | 5 | 9 | 8 | 11 |

Table 1. Rank of the corresponding version of the same piece within a list of 88 candidates, i.e., rank of version 2 when the query was version 1 and the other way around. ‘Standard Alg’ refers to a DTW-based matching algorithm without our two new extensions *split-and-merge* and *off-diagonal cost* (but with *relative path cost*, which was already proposed by [5]).

6.2 Results

In analogy to the MIREX audio cover song detection task, each of the 3×2 test recordings is in turn used as the query file. The respective ranking of the 88 remaining candidates is then examined and given in Table 1.

Although there is only one perfect match, we consider the results promising, given the nature of the data. In comparison to popular music, a different version of a piece is not only played on a different instrument, but, as can be presumed from the durations in Table 2, there are significant differences in which subset of the underlying piece is performed at all.

Table 1 also shows that the proposed system significantly outperforms a reference system – a ‘standard’ audio matching algorithm without the *split-and-merge* approach and the *off-diagonal cost* (but with *relative path cost*, which was already proposed by [5]). Looking at the mean rank of the corresponding version of the 6 query recordings, the two systems achieve values of 4.2 and 8.3 respectively.

Clearly, our algorithm is not precise enough for the fully automatic identification of matching recordings in a music collection as difficult as the one we are targeting. (In

| Query Piece | Ver. 1 | Ver. 2 |
|--------------|--------|--------|
| Fra Diavolo | 2:13 | 3:25 |
| Elias | 0:58 | 1:53 |
| The Creation | 0:58 | 1:55 |

Table 2. Performance times of different versions of same piece of music.

fact, neither are other state-of-the-art cover version detection algorithms in their domains of pop and rock music.) However, it may be useful as a component in an interactive search process. Also, we do have some ideas for possible improvements via quasi-transcription and higher-level representations (see below).

7. CONCLUSIONS

The paper has presented a first approach towards the automatic detection of versions of the same piece of music played by different musical automata. We have described the difficulties arising from the characteristics of these kinds of musical instruments and the different ways of arranging pieces for them. The proposed system is designed to be robust to the degrees of freedom instrument makers have, such as implementation of different subsections of the same theme, transpositions, or slight variations in tempo. To this end, each piece is split into several chunks which are compared separately, allowing each possible transposition. The comparison is based on an alignment obtained by the DTW algorithm and is evaluated via a similarity measure that combines match quality along the alignment path and plausibility of this path itself. Results from individual pairs of chunks are then combined to a final judgment about the similarity between two recordings.

The data set used for testing comprised 89 pieces including 3 pairs of recordings which share the same original. (Finding these three pairs of matching recordings in the

collection involved quite some effort.) That leaves us with only a small number of possible test setups. Future work will focus on extending the data set, including ‘ground truth’ concerning subsets of recordings that relate to the same piece.

Generally, historical mechanical instruments are limited in various ways – for instance, they generally have a rather restricted tonal range, little freedom or variation in terms of how tones are produced or modulated, etc. That might make it possible to perform some kind of automatic transcription, or at least a mapping onto a high-level representation (e.g., a list of played pitches), which again would facilitate a comparison at a higher level. On the other hand, each instrument has different tonal characteristics. Therefore, for each piece, individual tone models would need to be learned in an unsupervised manner. Given that the length of many recordings is less than 30 seconds, this is error prone as well. Still, the idea of introducing a higher-level representation of the audio signals is intriguing and will be investigated in future work.

Acknowledgments

This research is supported by the Austrian Research Fund (FWF) under grants TRP109-N23 and Z159. We would like to thank Helmut Kowar (Phonogrammarchiv of the Austrian Academy of Sciences) for providing the audio recordings for our experiments.

8. REFERENCES

- [1] J. Serrà, X. Serra, and R. G. Andrezejak: “Cross Recurrence Quantification for Cover Song Identification”, *New Journal of Physics*, Vol. 11, Issue 9, 093017, 2009.
- [2] S. Ravuri and D. P. W. Ellis: “Cover Song Detection: From High Scores to General Classification”, *Proceedings of the IEEE International Conference on Audio, Speech, and Signal Processing (ICASSP)*, pp. 65–68, Dallas, 2010.
- [3] E. Di Buccio, N. Montecchio, and N. Orio: “A scalable cover identification engine”, *Proceedings of the International Conference on Multimedia (MM’10)*, Firenze, Italy, 2010.
- [4] J. Serrà, E. Gómez, and P. Herrera: “Audio Cover Song Identification and Similarity: Background, Approaches, Evaluation and Beyond”, *Advances in Music Information Retrieval (Z. W. Ras and A. A. Wic-zorkowska Eds.)*, *Studies in Computational Intelligence*, Vol. 247, pp. 307–332, Springer, Berlin, 2010.
- [5] R. J. Turetsky and D. P. W. Ellis: “Ground-Truth Transcriptions of Real Music from Force-Aligned MIDI Syntheses”, *Proceedings of the 4th International Symposium of Music Information Retrieval (ISMIR)* Baltimore, MD, 2003.
- [6] N. Hu, R. B. Dannenberg, and G. Tzanetakis: “Polyphonic Audio Matching and Alignment for Music Retrieval”, *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, 2003.
- [7] L. R. Rabiner and B. H. Juang: “Fundamentals of speech recognition”. Prentice Hall, Englewood Cliffs, NJ, 1993.
- [8] J. Princen, J. Illingworth, and J. Kittler: “A formal definition of the Hough transform: Properties and relationships”, *Journal of Mathematical Imaging and Vision*, Vol. 1, Issue 2, pp. 153–168, Springer Netherlands, 1992.
- [9] M. Atiquzzaman and M. W. Akhtar: “Complete line segment description using the Hough transform”, *Image and Vision Computing*, Vol. 12, Issue 5, pp. 267–273, 199.

DEMETRIO STRATOS RETHINKS VOCAL TECHNIQUES: A HISTORICAL INVESTIGATION AT ISTC IN PADOVA

Elena Ceolin
Università di Padova
Dipartimento di Storia delle Arti
Visive e della Musica
ceolin.elena@libero.it

Graziano Tisato
ISTC (Istituto di Scienze e Tecnologie della
Cognizione), CNR, Padova
tisato@pd.istc.cnr.it

Laura Zattra
Università di Padova
Dipartimento di Storia delle Arti
Visive e della Musica
laura.zattra@unipd.it

ABSTRACT

Demetrio Stratos (1945-1979) was a singer known for his creative use of vocal techniques such as diplophony, bintonality and diphony (overtone singing). His need to know the scientific explanation for such vocal behaviors, drove him to visit the ISTC in Padova (Institute of Cognitive Sciences and Technologies) in the late Seventies. ISTC technical resources and the collaboration with Franco Ferrero and Lucio Croatto (phonetics and phoniatric experts), allowed him to analyze his own phono-articulatory system and the effects he was able to produce.

This paper presents the results of a broad historical survey of Stratos' research at the ISTC. The historic investigation is made possible by textual criticism and interpretation based on different sources, digital and audio sources, sketches, various bibliographical references (published or unpublished) and oral communications.

Sonograms of Stratos' exercises (made at the time and recently redone) show that various abilities existed side by side in the same performer, which is rare to find. This marks his uniqueness in the avant-gard and popular music scene of the time.

The ultimate aim of this study was to produce a digital archive for the preservation and conservation of the sources related to this period.

1. INTRODUCTION

Efstratios Demetriou (April 22, 1945 – June 13, 1979), better known as Demetrio Stratos, was a multi-instrumentalist, music researcher and singer. He is known for his activity with the Italian progressive rock group Area, as well as for his collaborations with other artists and his solo activity.

His interests in ethnomusicology and extra-European traditions, the complete mastery of a wide range of vocal techniques and the awareness of the spoken language constraints, were his background. That induced him to free the voice from the linguistic superstructures and to explore the underlying sonic substance. Among the most

impressive results of his research on what he called the instrument-voice [1], a series of unbelievable performances must be mentioned, mainly in the whistle register, producing two or three inharmonic partials at the same time, in a frequency region that could reach the 8,000 Hz.

During the late Seventies, Stratos visited several times the ISTC of CNR (Consiglio Nazionale delle Ricerche) in Padova (<http://www.pd.istc.cnr.it/>). Here, he worked with the physicist Franco Ferrero, who was an expert in phonetics, and with Lucio Croatto, phoniatric expert, to explore this vocal effects by the means of ISTC technical resources: a spectrograph of the VoiceIdentification and an electroglottograph (or EGG, or laryngograph) Elettro-Glottograph EG 830 by the F-J Electronics [9].

Unfortunately, Stratos' premature death at age 34 put an end to his research activity, which would have provided other results and assertions to his original view and definitely future pedagogical and scientific outcomes.

Why, thirty years after his death, the myth of Demetrio Stratos' voice is still alive and growing? His unusual extension of vocal techniques, the musical use of his vocal features and his penchant for scientific research, show the emergence of a figure in the musical scene who struggles against established vocal techniques and monody, but also against the established music industry.

What was Stratos looking for from phonetics, physicists, and phoniatric experts? In what way did he try to study his ability to obtain such complicated techniques? This paper is an attempt to reply to this double question.

The investigation starts from two premises. The first one is the necessity to study the Stratos' scientific experience through the sources, an aspect which has not yet been considered by literature. Literature normally has a target audience from the rock ambience. Stratos should be considered more broadly for his scientific contribution. His Paduan period is significant for this reason. It is a particular case where creativity rethinks science, as happened worldwide in the musical and scientific research centers (San Diego, Stanford Universities, Ircam, CSC in Padova, etc.). The Seventies drove musicians and scientists to collaborate in order to understand how voice and instruments physically worked (think e.g. the spectral analysis and physical models research, etc.). In this case, the performer himself decides to study his own voice and be more acquainted with his own capabilities.

The second premise emphasizes the problem of music conservation and preservation. The ISTC shares the same problem musical archives or musical institutions have. Oblivion or inaccurate preservation exist because of scientists and musicians continued acquisition of knowledge and the urge of experimentation, that brought to postpone, and often to forget, the organization and preservation of their musical materials [2, 3]. Together with the reconstruction of Stratos' experience, this research has been based on the philological method, in which the researcher follows different steps in his investigation. First of all the philologist aims to the complete *recension* and description of extant sources (even oral witnesses). Then he proceeds listing all different sources and name them with abbreviations derived from their content or origin. XX century music is characterized by heterogeneous sources (audio and video sources, sketches, digital sources, spectrograms and/or digital scores or description, oral witnesses) therefore he must consider all of them. The accurate description of the sources is the third moment of the investigation [4].

Section 2 gives an overview of the historical experience. Section 3 describes Ferrero and Stratos' work during the recording and the analysis of the vocal effects. Section 4 describes sources, methodological problems and the organization of the new archive with Stratos-related materials at the ISTC [5].

2. STRATOS AT THE ISTC (1976-1978)

This section tells the story of Stratos' presence at the ISTC, as derived from the sources mentioned in chapter 4.2.

2.1A necessity

Many are the reasons why Stratos had found the necessity to analyze his own voice compelling. First of all, as it is stressed by all Area members, Stratos' background was a melting pot of Greek, Egyptian and Balcanic musical traditions. Another field which certainly caused this interest were infant and newborn voices. Daniela Ronconi, Stratos' wife, told that Stratos was fascinated by his daughter Anastasia's voice, especially during the lallation phase;¹ he kept asking himself the reason why people lose this interesting capability while they grow up [6].

Another fact happened in 1974, when Stratos performed the John Cage piece *Sixty-two Mesostics Re Merce Cunningham* (1971) for voice unaccompanied using a microphone, a score that demands the performer a great independence and liberation. The occasion to meet Tran Quang Hai, a renowned interpreter of Eastern musical tradition and harmonic chant, was also important and allowed Stratos to learn this way of singing and its philosophical implication.

Finally, Nicola Bernardini had a relevant part in Stratos' experience. Bernardini (at the time a member of Prima Materia) met Demetrio Stratos and exchanged discussions. They used to perform overtone singing compe-

titions («we travelled together a lot at the time, and practicing the overtone singing was the best pastimes!» [17]).

2.2At the ISTC

The musical experimentation was not enough for Demetrio Stratos. He needed to give a scientific explanation to the phono-articulatory phenomena. This is why he firstly asked Pino Sambataro, his reliable otorhinolaryngologist in Milan to study his voice. But because of his inability to understand how Stratos' voice worked, this doctor decided to contact a colleague in Padova, Maurizio Accordi, who was in his opinion a specialist in this field. Accordi and Lucio Croatto, director of the ISTC (at the time Centro per le Ricerche di Fonetica) examined Stratos' phono-articulatory system accurately, and found nothing unusual. No special instrumentation was used, but only a laryngeal little mirror, for the videolaringostroboscopy did not exist yet.

Then they thought to examine Stratos' voice also from an acoustical point of view, and take him to the ISTC [6]. This happened during Fall 1976, which is also confirmed by [5] and the researcher and physicist Kyriaki Vagges, who worked at the time with Ferrero at the ISTC, and is the sole witness alive of that meeting [8].

During their meetings, Ferrero and Stratos recorded several improvisations, which were analyzed to observe their spectral content (several papers were published under the names of Accordi, Croatto and Ferrero).

Unfortunately, it is difficult to date those meetings, and yet material sources (audio sonograms and paper sources) do not give any help. Nonetheless it is certain that Stratos went to the ISTC during the years 1976, 1977 and 1978 (daa-SSN, daa-D,² and [12]) and in 1999 Ferrero declared to Janete El Haouli that he worked with Stratos at the ISTC for 4 or 5 times [1: 129].

Audio sources and sonograms demonstrate that most of the vocal material was recorded at the ISTC. Only a small number of audio recordings were made at Stratos' home [6]. Since evidently no electroglottograph tracks exist of those materials, it is easy to establish which vocal effects were recorded at ISTC under monitored conditions.

2.3Recording Demetrio Stratos

Speech and glottic sources were recorded by Ferrero team in the ISTC silent room, respectively on a Revox A77 tape recorder, and an Electro-Glottograph EG 830 by F-J Electronics [8]. The equipment also included an oscilloscope for the real time visualization and the signal analysis [8].

The vocal and glottal signals were recorded on two separate synchronous tracks. The speech signal was captured from a microphone at 10 cm from the mouth. The glottic source was acquired by means of two electrodes, attached to both side of the neck, in correspondence of the larynx. This allowed to pick up the rough signal of the vocal cords, not filtered by the resonances of the vocal tract. The absence of articulatory effects takes away all

¹ The infant baby's gibberish (from Latin *lallāre*: to sing lullaby, a verb containing the concept of producing alliterative sounds). In phonetics it means more generally a defect of speech (replacement of L for R).

² The sources are listed in chapter 4.2.

intelligibility and all human characteristics from the sound, and makes it like a buzzing of a reed instrument.

The subsequent analyses were made on a massive Voiceidentification Spectrograph 700 (same machine used in forensic application) by Franca Zecchin, who did the very first study of those materials [9]. The maximum frequency band the machine could capture was 8 KHz, which explains why Ferrero-Zecchin analysis established that Stratos could perform some extraordinary bitonality effects reaching 7000 Hz [12]. However, today analysis (made in 2002) has allowed to determine that Stratos' maximum was much higher, of about 8000 Hz (Fig. 3).

3. VOCAL EXERCISES AND ANALYSIS

3.1 Stratos' original vocal technique

Stratos' ability allowed him to produce diplophony, bitonality and diphony (overtone singing) [10].³ Diplophony is the ability to make two sounds at the same time. Vocal cords vibrate asymmetrically, and produce a waveform period with normal amplitude followed by a feebler one. One cannot always perceive two separate sounds: a normal period followed by an anomalous period means that the perceived frequency is one octave lower (for psychoacoustic reasons it is sometimes difficult to distinguish). If two normal periods are followed by one abnormal, the pitch of the perceived sound results an octave and a fifth lower [10]. What is heard, consequently, is a 'dirty' and scratched voice, because two frequency components fall in the same critical band and because of the masking of the lower partials above the higher. If this phenomenon can be sometimes accidental in the pathological voices and sometimes also in singing, Stratos made it intentionally.

Bitonality is the unusual capability to produce two different sounds which are sometimes not in a harmonic relation. In normal conditions, the vocal folds produce a sound with harmonic spectral components, i.e. the frequencies of partials are multiple of a fundamental, or separated by the same frequency interval. Sometimes the contraction of false vocal cords provokes a second sound due to the low frequency modulation. Some other times, strong false vocal cords contractions trigger very high whistles. In the case of bitonality, the adduction of vocal cords is so strong that it generates two independent non-harmonic sounds, as it happens when you touch with a finger the string on a musical instrument and the original sound splits in two. In some of Stratos' effects, the EGG demonstrated the absence of the vocal cords vibration: in this case, the perceived pure high frequency whistles are due to the reduced dimension of pharyngeal resonators [10].

Overtone singing is the extraordinary way to split the harmonic partials of a vocal sound, normally fused in a single one, in two distinct sonic images: one in the usual

vocal range of the singer, the other in an high or very high register; this pure and flute-like sound corresponds to one of the harmonic partials reinforced by the resonances or formants of the spectral envelope. In enhancing the harmonic partial, one can create real melodies. An overtone singer can 'play' these harmonic pitches in a scale which is a natural pentatonic scale (see Zarlino) [10].

3.2 Analysis and Sonograms

Analyses of Stratos voice are mentioned in the following sources: dav-L, dc-VAL82, dc-RIV80, dc-RIV80/cp, dc-T/BATT, dc-T/COP, dc-CE/CNR, dc-SON, dc-T/ZECC (see Table 2).

Franca Zecchin's graduation thesis is the first study dedicated to Stratos' vocalizations. It is a very significant source because it was made during the period in which Stratos came at the ISTC [9, 11], and because it allows establishing the origin and reliability of the audio sources. In this way it is possible to verify that nearly all Stratos' vocal effects were recorded at ISTC in controlled conditions, and cannot be the result of a (fraudulent) mixing, filtering, etc. From the entire series, Ferrero's team selected a set of 22 vocalizations to be the most representatives.

The thesis reports that examples from n.1 to n.18 were recorded at the CNR during the fall 1976. Vocalizations nn. 19, 20, 21 were brought by Stratos in September 1977, pre-recorded on a tape. Vocalization n. 22 was recorded during the Fall 1976 [9]. All those examples were subjected to analysis, even if some of them (nn. 1-6 and 12, 15, 16) do not appear in Zecchin's thesis nor in the published article [9] (Zecchin also does not mention vocalizations nn. 7 e 11).

The electrographic track of vocalizations nn. 8-10, 13, 14, 17, 18 is completely flat because of the absence of the vocal folds vibration.

The thesis also includes a brief description of Stratos' phono-articulatory attitude during the vocalization, but not the way it was deduced. Unfortunately, after 30 years, Zecchin does not remember the methodology they adopted. She makes two assumptions: the described vocalization mechanism could be a deduction made *a posteriori*, through the study of the sonograms and the formant positions and movements. A more likely explanation could be that Ferrero discussed with Stratos about what he felt inside his phono-articulatory system, and compared this 'sensations' with the sonographic results [11].

Figure 1 shows the analysis of fragment n. 18, as it is shown in the article made in 1980 [12]. It begins with 2 whistles of 3700 Hz (and 2nd harmonic, small triangle) and 5000 Hz (empty triangle).

³ Vocal analysis is one of Graziano Tisato's research topic, started during the Seventies. Soon after Stratos Paduan period, Tisato met Ferrero, and decided to determinate a precise terminology for the vocal effects. These were published in [14]. In 1989 Tisato realized the first model synthesis for the overtone singing [18].

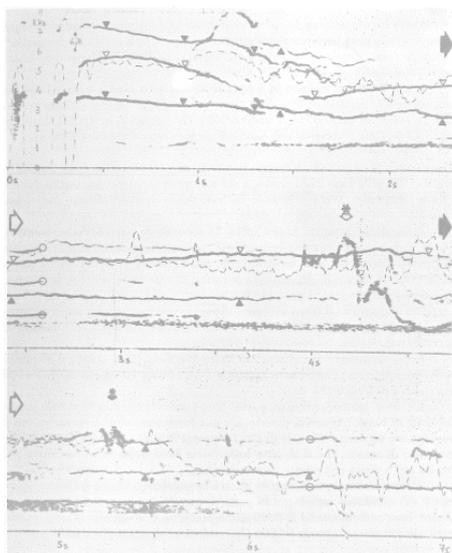


Figure 1. Vocalization n. 18 [12: 253].

In a graduation thesis made in 2005, Copiello showed new analyses of the 22 examples, which had been made by Tisato [16]. They considered sonograms, pitches and intensities contours [16]. The numeric representation shows unquestionable advantages and it solves those resolution and frequency limits of the Sonograph 700, thanks to the possibility to choose the proper frequency and time scale, and to obtain the parameter values straight from the data analysis. Table 1 shows the comparison [16]. Fig. 2 shows the same vocalization of Fig. 1 in an arbitrary time scale. The Fig. 3 shows a sonogram with an example of tritonicity.

| | | |
|--|----------------|--|
| V O C A L I Z E N o 1 8 | 1977 THESIS | -Phono articulatory attitude: the same as in nn. 8, 9 and 13. Spectrographic Recording -Cue: two non harmonic whistled notes, one at 3700 Hz + 2°, the other as pure sound beyond 5000 Hz. Their frequency is decreasing. -1st s: transition phase after which they continue as pure sound with non harmonic fluctuations. Whistled bitonal sound. -2nd - 3rd s: a whistle overlaps for three times with a fundamental frequency at 1660 Hz + 2nd and 3rd, with the result of a three-partite sound. -4th s: the main whistle with a lower frequency disappears. A flat changeable whistle overlaps the whistled note, stabilizing around 1500 Hz: bitonal sound, like "bird singing". -5th s: pitch at 1500 Hz + 2nd and flat inflected whistle (like bird singing) between 4000 and 5000 Hz. Electroglottographic recording: flat during the whole length. (vocalize 18 was recorded in 1976 at CNR. The tape include the verbal vocalize but not the glottic vocalize) |
| | 1980 STUDY | (annotations are made only if the analysis is different from the one made in 1977) Spectrographic Recording -During the whole vocalize there is a noise band around 1200 Hz Electroglottographic recording: Not quoted |
| | 2005 STUDY | (annotations where different) Spectrographic Recording: -Whistled at ~3555 Hz (La7) and 2nd harmonic; and beyond 5500 Hz pure sound. -1,5 s: the lower whistle loses the 2nd harmonic. -2-3 s: whistle at ~1660 Hz (Sol6) |

Table 1. Vocalize n. 18: comparison between [12: 253] and the new analyze with modern sonograms [16].

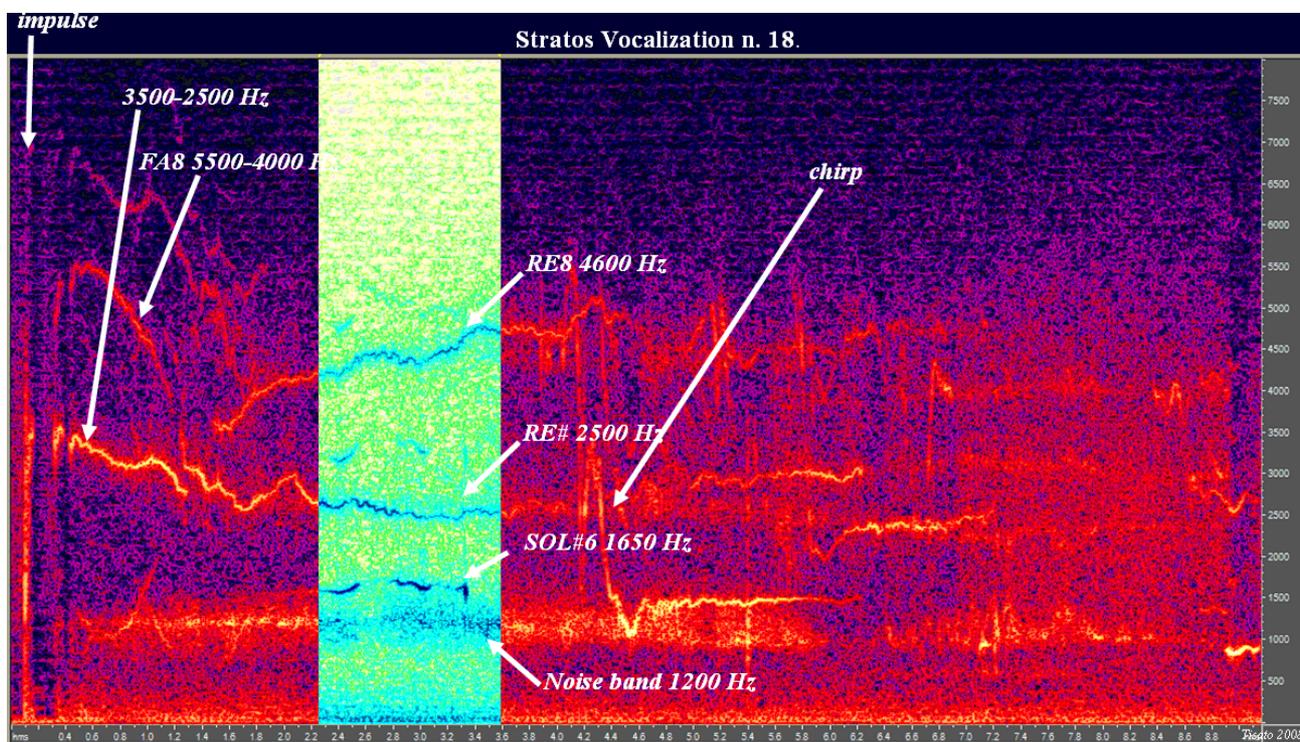


Figure 2. The vocalization starts with a guttural impulsive sound, then it proceeds until 2.2 s with two inharmonic whistles (bitonality case): the lower at 3500-2500 Hz presents a second harmonic, the higher slopes down from 5500 to about 4000 Hz. At this point until 3.3 s, a new inharmonic component appears at 1650 Hz to form the tritonicity. Around 1000-1200 Hz a very narrow noise band can be heard.

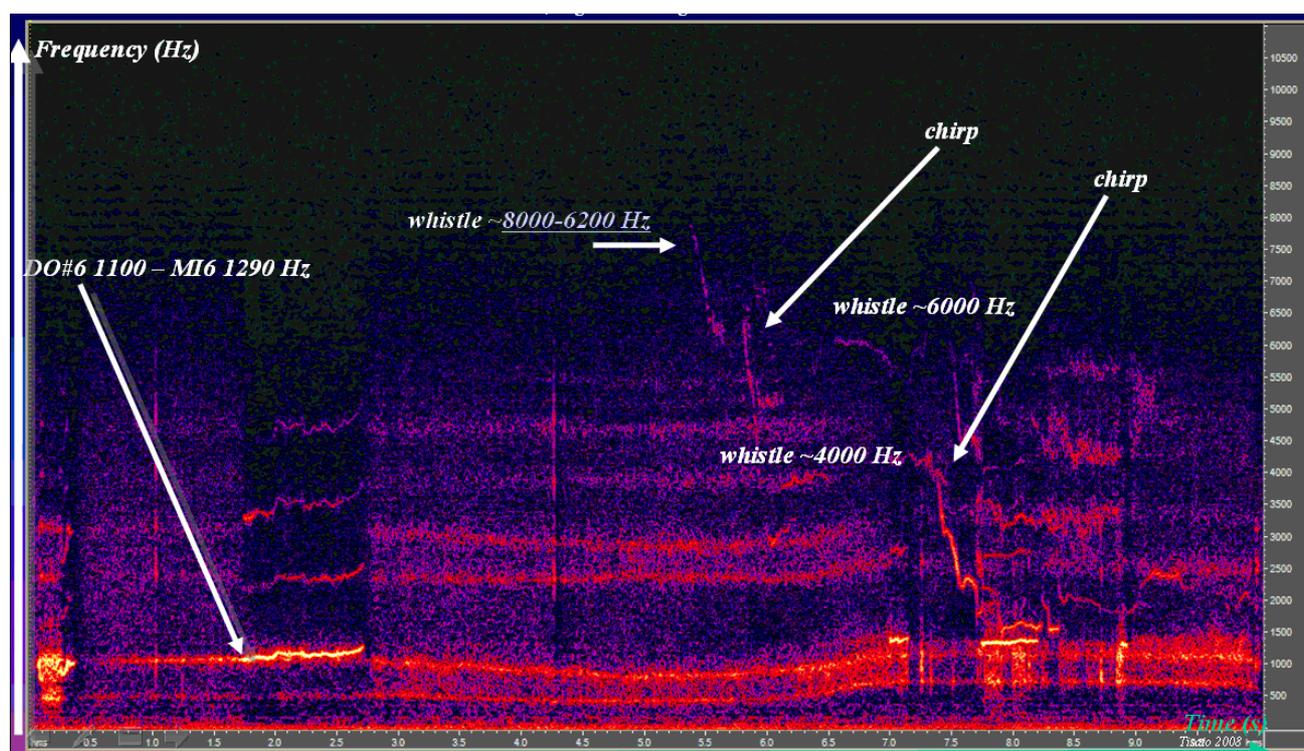


Figure 3. The vocalization starts with a breathy-vocalized sound of 300 ms, then 600 ms of an expiratory noise of low amplitude. At ~1.4 s, an high pitched cry with a set of 4 harmonics of 1 s can be heard, which raises from ~1100 (DO#6) to 1289 Hz (MI6-). Between 5-8 s, a chirp go down from ~7900 Hz to ~6200 Hz. Between 6-7 s a whistle appears at 4000 Hz and then at 6000 Hz. There is also a colored noise with 2 formants at 500 and 1000 Hz. At 7 s a chirp falling from ~4400 to 2250 Hz.

4. SOURCES

4.1 New texts and supports in philology

During the past half-century music and musical research (in both popular music and art music) have been made involving new ways of producing and analyzing sounds. As a result, sources that document information of this activity are heterogeneous and very different from the traditional status of ‘paper material’ [13]. They are not necessarily a visible or symbolic trace and can indifferently be: 1) the audio and video source (analogue, CD, the mini disc, the memory of the computer); 2) printed, handwritten sources; 4) traditional scores; 5) different sketches; 6) articles; 7) oral witnesses (oral communications find justification in the contemporary context where collaboration is important); 8) visual representations such as sonograms [3, 4]. Stratos’ experience is a good example of a XX century historical event that must be studied in detail– in terms of material and oral documentation. Chapter 4.2 shows the list of sources. Oral communications have been indispensable in reconstructing the historical facts and are included in [5].

4.2 Sources at the ISTC and elsewhere

Table 2 lists all materials collected during this research project. General Categories of sources are: Analogue audio document (daa), Visual analogue document (dav), Paper document (dc), Audiovisual digital document (dda). For specific description, see [5].

| SOURCE | Abbreviation |
|---|---------------------|
| “ <i>Cantare la Voce</i> ” (long play disc) | daa-CV |
| “ <i>Copia disco dimostrazione Titze</i> ”, (magnetic tape 1/8”, SONY HF-ES60, compact cassette) | daa-CT |
| “ <i>Demetrio</i> ” (magnetic tape 1/8”, BASF Chromdioxid 90, compact cassette) | daa-D |
| “ <i>Lo strumento voce , demo Lecce 14/4/94</i> ” (magnetic tape 1/8”, SONY Metal-xr 100, compact cassette) | daa-SVL |
| “ <i>Metrodora</i> ” (long play disc) | daa-M |
| “ <i>Napoli</i> ” (magnetic tape 1/8” BASF Chromdioxid 60, compact cassette) | daa-N |
| “ <i>Nastro cantanti 75/78</i> ” (magnetic tape, BASF, plastic flange) | daa-NC75/78 |
| “ <i>Nastro pre-Tesi , copia nastro Tesi, Stratos vocalizzi riversati</i> ” (magnetic tape 1/8”, BASF Chromdioxid 60, compact cassette) | daa-NPT |
| “ <i>Nastro Tesi su Demetrio Stratos</i> ” (magnetic tape 1/8”, BASF Cromdioxid 90, compact cassette) | daa-NTD |
| “ <i>Spectrograph7 (Stratos)</i> ” (magnetic tape, BASF Scotch, plastic flange) | daa-S7 |
| “ <i>Spectrograph 20</i> ” (magnetic tape, BASF, aluminium flange) | daa-S20 |
| “ <i>Stratos Ferrara</i> ” (magnetic tape 1/”, BASF Chromdioxid 90, compact cassette) | daa-SF |
| “ <i>Stratos Milano</i> ” (magnetic tape 1/8”, TDK, SF60, compact cassette) | daa-SM |
| “ <i>Stratos suo nastro</i> ” (magnetic tape 1/8”, TDK KR C60, compact cassette) | daa-SSN |
| Slides (mixed), ISTC archive, Padua | dav-D |
| “ <i>Lo strumento voce , demo Lecce 14/4/94</i> ”, transparencies, ISTC archive, Padua | dav-LSV |
| Transparencies, no date, ISTC archive, Padua (two parcels) | dav-L (two parcels) |

| SOURCE | Abbreviation | SOURCE | Abbreviation |
|--|--|---|--|
| | dav-L1 (first parcel) dav-L2 (second parcel) | <i>vocalizzazioni di Stratos Demetriou</i> attached to the first parcel of transparencies), without date, ISTC archive, Padua | |
| Accordi M., Croatto L., Ferrero F. E., "Analisi spettrografica di alcuni vocalizzi di Demetrio Stratos". In "Il Valsalva, bollettino italiano di audiologia e foniatra, Vol. V - N. 1", January - April 1982, pp. 2-8 | dc-VAL82 | Fortunato Roberto, "Rinascita la ricercata etichetta Cramps". In "Il mattino", Tuesday 11th July 1989, pp. 41 | dc-R.CRAMPS/M at |
| Accordi M., Croatto L., Ferrero F. E., "Descrizione elettroacustica di alcuni tipi di vocalizzo di Demetrio Stratos". In "Rivista Italiana di Acustica, Vol. IV - N. 3" 1980, pp. 229-258 | dc-RIV80 | Gatti Roberto, "In alto la voce". In "L'Espresso", 4th of June 1989, pp. 135-136 | dc-IAV/Esp |
| Accordi M., Croatto L., Ferrero F. E., "Descrizione elettroacustica di alcuni tipi di vocalizzo di Demetrio Stratos". In "Rivista Italiana di Acustica, Vol. IV - N. 3" 1980, pp. 229-258, copy of the original typewritten publication | dc-RIV80/cp | Kemp Alan, Linsley Geoff, Verhoeven Jo, "Practical Phonetics", Edinburgh University Linguistics Department, pp. 1-8, without date | dc-PP |
| Accordi M., Ferrero F., Ricci Maccarini A., Tisato G., "Il canto difonico, un esempio delle possibilità del tratto vocale", comunicazione presentata al "XVIIth Congress of Union of the European Phoniatrists, Salsomaggiore, 10-14 Ottobre 1990". In "Quaderni del centro di studio per le ricerche di fonetica, vol. IX, 1990", pp. 574-613 | dc-CDIF | "La musique religieuse du Thiber", Bulletin du Groupe d'Acoustique Musicale, 58, Université Paris VI, 1972 | dc-MR/Th |
| Baroni Vittore, "Cometa Rossa: la Musica è un Gioco Rischioso", pp. 31-33, no date | dc-COM | "Lo strumento voce, demo Lecce 14/4/94", sheets with notes | dc-SV/app |
| Battain Valeria, "Un archivio di documenti sonori non convenzionale: il fondo Demetrio Stratos dell'ISTC (Istituto di Scienze e Tecnologie della Cognizione, ex Istituto di Fonetica e Dialettologia) del CNR di Padova". Thesis of the Academic year 2006-2007, Università degli studi di Udine, Supervisor: Professor Sergio Canazza Targon | dc-T/BATT | Transparencies, paper copy, without date, ISTC archive, Padua, (two parcels) | dc-Lc (two parcels) dc-L1 (first parcel) dc-L2 (second parcel) |
| "Cantare la voce", poster, congress programme and brochure, Monday 29th e Tuesday 30th May 1989, ISTC archive, Padua | dc-CV/L (poster) dc-CV/P (congress programme) dc-CV/O (brochure) | Mangiarotti Marco, "Stratos: la musica è gioia e rivoluzione". In "Il giorno", Sunday 25th June 1989 | dc-MGR/Gio |
| Copiello Laura "Demetrio Stratos, una vocalità riscoperta". Thesis of the Academic year 2004-2005, Università degli studi di Venezia, Supervisors: Professori Giovanni Morell, Graziano Tisato e Domenico Stanzial | dc-T/COP | Mattarelli Luca, "Demetrio Stratos, una nuova vocalità". Thesis of the Academic year 1994-1995, Università degli studi di Bologna, Supervisor: Professor Gino Stefani | dc-T/Matt |
| Fariselli Loretta, Fariselli Patrizio, "Demetrio Stratos - Area, dieci anni di musica ed impegno", programme of the demonstration on the 4th of July at 9.00 pm in Piazza Mercato, Marghera, unpublished work | dc-FAR/09 | "Nastro cantanti 75/78", cover with notes, without date, ISTC archive, Padua | dc-NC75/78 |
| Ferrero E. Franco, "Attività di studi e ricerca sulla voce cantata", study presented at "Seminario CIRM, 5 Febbraio 1997", unpublished work | dc-CIRM | Ricci Maccarini Andrea, "Il canto difonico". Thesis of the Academic Year 1989-90, Università degli studi di Padova, Supervisor: Professor Maurizio Accordi, Co-Relatore: Dott. Franco Emilio Ferrero | dc-T/RIC |
| Ferrero E. Franco, "Caratteristiche elettroacustiche di alcune singole vocalizzazioni di Stratos Demetriou", Centre for the study of phonetic researches (CNR) in Padua, unpublished work | dc-CE/CNR | Receipt of payment to Franco Ferrero for the conference he held on the 29th of September in the auditorium San Rocco - Vocal Music Festival called "Caratteristiche elettroacustiche di alcuni tipi di vocalizzo di Demetrio Stratos" | dc-RIC |
| Ferrero E. Franco, "Elenco delle pubblicazioni", 30 Settembre 1997, ISTC archive, Padua | dc-EP | Sonagrams (mix), ISTC archive, Padua | DC-SON |
| Ferrero F., Ricci Maccarini A., Tisato G., "I suoni multifonici nella voce umana", article presented at "XIX Convegno Nazionale 10-12 Aprile 1991, Napoli", pp. 415-422 | dc-SM | "Spectrograph 7 (Stratos)", white cardboard with notes and writings, without date, ISTC archive, Padua | DC-SP7 |
| Ferrero Franco, "Elementi di Fonetica", publication n. 85 from the list of publications, pp. 1-43 | dc-EF | "Spectrograph 20", sheet with notes, without date, ISTC archive, Padua | dc-SP20 |
| Ferrero Franco (hypothesis of the autor based on the hand writing), "Fonografia e costo vocale", study, without date, ISTC archive, Padua, unpublished work | dc-FON | "Vocalizzi di Stratos (pre-tesi)", sheet with notes, without date, ISTC archive, Padua | dc-VOC/pt |
| Ferrero Franco, "La fonetica strumentale in funzione della diagnostica foniatrica e della riabilitazione logopedica", notepad bound by hand, handwriting by Ferrero, without date, ISTC archive, Padua | dc-FS/lib | "Vocalizzi Tesi", sheet with notes, without date, ISTC archive, Padua | dc-VOC/t |
| Ferrero Franco, "Lo strumento voce", study, without date, ISTC archive, Padua | dc-SV | Zecchin Franca, "Studio elettroacustico di alcuni vocalizzi di Demetrio Stratos". Thesis of the Academic Year 1977-1978, Università degli studi di Padova, Supervisor: Professor Franco Ferrero | dc-T/ZECC |
| Ferrero Franco, three lists of thesis (supervisor F. Ferrero), ordered by location, Academic year and alphabetical order of the titles, without date, ISTC archive, Padua | dc-T/el | "Copia disco dimostrazione Titze", conservative copy CIF0001 of the ISTC archive (DVD-data: 2+2 tracks, WAV, 96kHz-24bit) | DDA-1 |
| Tissue paper sheets, without date, ISTC archive, Padua | dc-FCV | "Demetrio", conservative copy CIF0002 of the ISTC archive (DVD-data: 2 tracks, WAV, 96kHz-24bit) | DDA-2 |
| Sheet of paper with the description of vocalizes 18 and 6 (copy of a page "Caratteristiche elettroacustiche di alcune singole") | dc-L1 | "Nastro cantanti 75/78", conservative copy CIF0008_CCIR of the ISTC archive (DVD-data: 2 tracks, WAV, 96kHz-24bit) | DDA-8/C |
| | | "Nastro cantanti 75/78", conservative copy CIF0008_NAB of the ISTC archive (DVD-dati: 2 tracks, WAV, 96kHz-24bit) | DDA-8/N |
| | | "Nastro cantanti 75/78", conservative copy CIF0008_V of the ISTC archive (DVD- data: 1 track, MOV) | DDA-8/V |
| | | "Nastro tesi su Demetrio Stratos", conservative copy CIF0006 of the ISTC archive (DVD- data: 2 tracks, WAV, 96kHz-24bit) | DDA-6 |
| | | "Spectrograph7 (Stratos)", conservative copy CIF0007 of the ISTC archive (DVD- data: 1 track, WAV, 96kHz-24bit) | DDA-7 |
| | | "Spectrograph7 (Stratos)", conservative copy CIF0007_V of the ISTC archive (DVD- data: 1 track, MOV) | dda-7/V |
| | | "Spectrograph 20", conservative copy CIF0009_CCIR of the ISTC archive (DVD- data: 2 tracks, WAV, 96kHz-24bit) | DDA-9/C |
| | | "Spectrograph 20", conservative copy CIF0009_NAB of the | DDA-9/N |

| SOURCE | Abbreviation |
|---|--------------|
| ISTC archive (DVD- data: 2 tracks, WAV, 96kHz-24bit) | |
| “Spectrograph 20”, conservative copy CIF0009_V of the ISTC archive (DVD- data: 1 track, MOV) | DDA-9/V |
| “Stratos Ferrara”, conservative copy CIF0004 of the ISTC archive (DVD- data: 2 tracks, WAV, 96kHz-24bit) | DDA-4 |
| “Stratos Milano”, conservative copy CIF0003 of the ISTC archive (DVD- data: 2 tracks, WAV, 96kHz-24bit) | DDA-3 |
| “Stratos suo nastro”, conservative copy CIF0005 of the ISTC archive (DVD- data: 2 tracks, WAV, 96kHz-24bit) | DDA-5 |

Table 2. Sources for the study of Demetrio Stratos at the ISTC.

Several additional sources are not listed here: these are articles, dissertations, projection papers, sleeves/sheets and annotations related to the audio material. The existence of documentations scattered over an extended period of time— from the Seventies up to now— tells not only the interest towards Stratos, it also ensure the importance of his musical research.

4.3 Conservation and Preservation

This project ultimate aim has been to preserve the entire documentation from obsolescence and deterioration; that is why, following the typology of sources, a digital archive has been made, which is now available at the ISTC. The digital archive purpose is: 1) to preserve a specific order (folders – e.g. digitization of paper materials – are maintained in the same sequence of the original sources), so that the sources cannot get lost in different places; 2) to preserve the chronology of their creation; 3) to guarantee the accessibility to whoever is interested; 4) to avoid damages to the authentic sources that could be caused by an incorrect use.

The archive refines and benefits of a previous research. In 2007 Valeria Battain created digital conservative copies of the entire documentation related to Demetrio Stratos at the ISTC [15]. The storage in digital format included three magnetic open tape reels (daa-NC75/78, daa-S7, daa-S20) and six compact cassettes (daa-CT, daa-D, daa-NTD, daa-SF, daa-SM, daa-SSN). Battain’s study provides a descriptive paper and picture for each original source; it also includes other papers with technical information related to the process. It was however incomplete.

The new digital archive is divided into two parts: the first one is labeled “ARCHIVIO *Demetrio Stratos*, ISTC (Istituto di Scienze e Tecnologie della Cognizione), CNR, Padova”; it relates to a selected space inside one of the shelves at the ISTC; it correspond to ‘tangible’ documents (e.g. thesis, tapes, compact-cassettes, DVD with recordings made by the eng. Sergio Canazza, etc.); the second part is an external HD USB with the whole sources that have been digitized during this work. For the moment, the archive can be accessed only locally through the computers of the Institute (ISTC). The structure of the archive reflects the choice made in organizing the whole sources. It simply systematizes them in 4 macro-categories: analogue audio document (daa), visual analogue document (dav), paper document (dc), audiovisual digital document (dda). Each folder is a container for the digitalized documents. These are: 14 ‘analogue’ audio

documents, 14 audiovisual documents (digital), 4 visual documents (‘analogue’), 40 paper documents. The documents are identifiable through the abbreviated text extension given during this research, instead of verbose and long names (see Table 2).

5. CONCLUSIONS

The reconstruction of Stratos’ experience in Padova has been made possible by the collection, the description, the analysis and the comparison of sources. Yet, this study is needless to say the first step toward investigating Stratos’ contribute to XX century vocal research. Just as an example, Stratos’ position in the avant-garde vocal research is almost completely unknown, but many aspects of his life are critical. Area member Paolo Tofani recalls the precise moment when Stratos realized the novelty of the overtone singing [19]. It was in 1976/77, when a journalist brought to the group an audiocassette with sounds sung by Mongolians: Demetrio tried and tried until he finally succeeded, because he had been influenced by John Cage; they also went to meet Cathy Berberian repeatedly and took lessons from Tran Quang Hai [19]. Also, his friendship with Nicola Bernardini, member of Prima Materia Group, is crucial. And needless to say, the milieu and historical period in which Stratos’ research took place are again crucial: the Seventies, when the youth movement reached Italy and the children of the second world war began their protest against the established culture and lifestyle. It is easy to state that this social rebellion reflects in the complete nonconformity and subversion of Stratos’ technique, an aspect that affected Stratos’ position in the musical industry, since his voice did not respect standardized vocal production, did not consider language’s rules and, on the contrary, tried to get loose from the detention of the communicative act. Future studies need to take into account all aspects of this issue.

The investigation of Stratos’ vocal effects may develop in two directions. The first one is related to the systematic investigation of the complete series of Stratos’ vocalizes, which had not been analyzed so far (as said in 3.4). The second one could help in deducing Stratos’ phonatory attitude: Voice Quality (VQ) methodologies and glottal source modelization techniques should be applied to the existing glottal tracks. In addition to the traditional parameters (Shimmer, i.e. the amplitude perturbations of the wave form; Jitter, i.e. the pitch perturbations; the Waveform Matching Coefficient, i.e. the cross-correlation between near periods; the Harmonics-to-Noise Ratio, i.e. the energy ratio between the harmonic partials and the noise components; etc.), it is in fact possible to extract more meaningful information: for example, the Glottal to Noise Excitation Ratio (GNE), which is used to discriminate among normal and pathological voices [20]. The available glottic tracks could give precious information about the glottal flow, in term of Open Quotient $Oq = T_e / T_0$, i.e. the ratio between the maximum excitation instant T_e and the sound period T_0 , and the Return phase quotient Q_a , the ratio between the return phase (in which the glottal flow reaches zero) and the closed phase of the vocal folds. The Return phase quotient proved to be the

most effective index of VQ, for it determines the sound Spectral tilt, i.e. the slope of the frequency envelope [21].

Another development is the one mentioned before. A web page of the archive would also be desirable (possibly at: www.pd.istc.cnr.it/Stratos), with the audio vocal material and the PDF documents (of course in agreement with the authors). Accessing the stock of documents via metadata would be imperative. The web access would guarantee a more large accessibility to the sources; this also should adhere to Stratos' personal interest in the study and dissemination of his own personal research.

Acknowledgments

We would like to thank Piero Cosi, Alberto Benin, Nicola Bernardini, Ferdinando Bersani, Maurizio Accordi, Oskar Schindler, Francesco Avanzini, "Area" members Paolo Tofani, Ares Tavolazzi, Patrizio Fariselli and Claudio Rocchi, Valeria Battain, Kyriaki Vaggas and Franca Zecchin, Sergio Canazza Targon and Daniela Ronconi Demetriou. This research would not have been possible without their help and advice.

6. REFERENCES

- [1] J. El Haouli (1999), "*Demetrio Stratos, alla ricerca della voce-musica*", Milano, Casanova e Chianura edizioni, 1999/2009.
- [2] N. Bernardini, A. Vidolin, "Sustainable live electro-acoustic music", Cdrom proceedings Sound and Music Computing 2005 – XV, CIM - Nov. 24-26 2005, Salerno, Italy, 2005.
- [3] L. Zattra, *Studiare la computer music. Definizioni, analisi, fonti*, (collana biblioteca contemporanea), libreriauniversitaria.it, 2011, ISBN 8862921152.
- [4] L. Zattra "Sources and philological problems in the study of Computer Music", in *Elektroakustische Musik: Technologie, Aesthetik und Theorie als Herausforderung an die Musikwissenschaft*, T. Boehme-Mehner, K. Mehner, M. Wolf eds., Essen, Die Blaue Eule Verlag, 2008, pp. 109-118.
- [5] E. Ceolin, "*Demetrio Stratos a Padova: la storia, le fonti, l'archivio*", Graduation Thesis, DAMS, Supervisor: S. Durante, L. Zattra, March 2011.
- [6] D. Ronconi Demetriou, interview given to Elena Ceolin, June 25, 2010.
- [7] M. Accordi, interview given to Elena Ceolin, October 5, 2010.
- [8] K. Vaggas, interview given to Elena Ceolin, February 18, 2011.
- [9] F. Zecchin, "*Studio elettroacustico di alcuni vocalizzi di Demetrio Stratos*", graduation thesis, 1977-1978, Università degli studi Padova, Supervisor: Prof. Franco Ferrero, 1978.
- [10] G. Tisato, interview given to Elena Ceolin, February 7, 2011.
- [11] F. Zecchin, interview given to Elena Ceolin, February 17, 2011.
- [12] M. Accordi, L. Croatto, F. Ferrero, "Descrizione elettroacustica di alcuni tipi di vocalizzo di Demetrio Stratos", in «Rivista Italiana di Acustica», Vol. IV – N. 3, 1980, pp. 229-258.
- [13] M. Caraci Vela, *La filologia musicale. Istituzioni, storia, strumenti critici*, Vol. 1, LIM, 2005.
- [14] F. Ferrero, A. Ricci Maccarini, G. Tisato (1991), "I suoni multifonici nella voce umana", in XIX Convegno Nazionale AIA – 10-12 Aprile 1991, Napoli, 1991, pp. 415-422.
- [15] L. Battain (2007), "*Un archivio di documenti sonori non convenzionale: il fondo Demetrio Stratos dell'ISTC (Istituto di Scienze e Tecnologie della Cognizione, ex Istituto di Fonetica e Dialettologia) del CNR di Padova*", Graduation Thesis 2006-2007, Università degli studi di Udine, Supervisor: Prof. Sergio Canazza Targon, 2007.
- [16] L. Copiello, "*Demetrio Stratos, una vocalità riscoperta*". Graduation Thesis 2004-2005, Università degli studi di Venezia, supervisors: Proff.s G. Morelli, G. Tisato e D. Stanzial, 2005.
- [17] N. Bernardini, interview given to Elena Ceolin, June 8, 2010.
- [18] G. Tisato, "Analisi e sintesi del Canto Difonico", in Proceedings VII Colloquio di Informatica Musicale (CIM), Cagliari, 1989, pp. 33-51.
- [19] P. Tofani, interview given to Elena Ceolin, March 21, 2011 (in the presence of G. Tisato and Giovanni Floreani).
- [20] D. Michaelis, M. Froehlich, H.W. Strube, "Selection and combination of acoustic features for the description of pathologic voices" in *J. of the Acoust. Soc. Am.*, 103(3), 1998, pp. 1628-39.
- [21] B. Doval, C. d'Alessandro, N. Henrich, "The spectrum of glottal flow models" in *Acta Acustica*, 92, 2006, pp. 1026-1046, http://rs2007.limsi.fr/index.php/PS:Page_2

SOUNDSCAPE: A MUSIC COMPOSITION ENVIRONMENT DESIGNED TO FACILITATE COLLABORATIVE CREATIVITY IN THE CLASSROOM

Dr. Sylvia Truman

Faculty of Business and Management

Regent's College

London, UK

trumans@regents.ac.uk

ABSTRACT

A question that has gained widespread interest is 'how can learning tasks be structured to encourage creative thinking in the classroom?' This paper adopts the stance of drawing upon theories of learning and creativity to encourage creative thinking in the classroom. A number of scholars have suggested that the processes of 'learning' and 'creativity are inextricably linked. Extending upon this, a generative framework is presented which exists as a design support tool for planning creative learning experiences. A demonstration of how this framework can be applied is made through the design of SoundScape – A music composition program designed for school children. This paper reports upon a study using SoundScape within a school with 96 children aged 11. The study focused on two objectives, firstly, identifying any differences in explicitly supporting the creative processes of 'preparation' as opposed to not, and secondly, comparing the outcomes of using real-world metaphors to create music compared to the use of abstract visual representation to specify music.

1. INTRODUCTION

The study reported in this paper focused on facilitating collaborative creativity in a music composition task. In particular, this paper draws together theoretical routes from learning and creativity theory. The study investigated the similarities between the two processes and based upon this a generate framework for creative learning is presented. This framework exists as a design support tool to assist with the design of creative learning experiences within the classroom. In this instance it is applied to the domain of a collaborative music composition task and was used to inform the design of SoundScape. SoundScape was designed to explore the research hypotheses driving the study. The hypotheses focused upon explicitly supporting the preparation phase of the creative processes using music technology and using visual metaphors to specify music. The findings hold a number of implications for the design of meaningful and

engaging learning experiences through considering aspects of the creative process.

2. THEORETICAL BACKGROUND: THE LEARNING PROCESS

2.1 Traditional Perspectives on Learning

Traditional pedagogy concerns itself with the passive absorption of knowledge, which is later tested in examination based scenarios. The underlying assumption of this approach places expectations upon the student to learn and recall knowledge. This is embodied via rote teaching methods [1]. Subsequently, students may respond in ways to meet what they perceive to be the teacher's expectations [2]. Brown *et al* assert that although learning abstract, de-contextualised concepts in the classroom equips students to pass examinations, they may encounter difficulty when applying concepts in authentic practice [3][4]. Secondly, students may rely upon particular features of the classroom context in which the task itself may have become embedded. This differentiates the task from authentic activity in the mind of the student. It is therefore emphasised here that learning should be set in a context appropriate to the concepts to be learned. This view is emphasised by the more contemporary approach of constructivism.

2.2 Social Constructivist Learning Theory

According to the constructivist approach, important aspects of learning are as follows: learning is contextual. Secondly, one needs knowledge to learn. It is not possible to assimilate new knowledge without having a previous knowledge structure. Thirdly, learning is a self-regulated process as every individual learns at a different rate depending on their prior knowledge and experience [5]. Finally, learning is viewed as an individual and social activity in which interactions with others and the external environment are conducive to learning [6][7]. Constructivism emphasizes that students learn by constructing meaning for themselves through active participation within a domain. This approach has a number of advantages. For example, by discussing their experiences with others, shared understandings can be developed [8]. This

is especially advantageous in collaborative settings. Many have argued that social interaction is paramount to cognitive development as learning occurs through interacting with others. This enhances the integration of newly acquired concepts into the mental structure of the learner.

2.3 The Constructionist Method of Learning: Learning by Building

Constructionist methods have sought to enhance the learning experience linking creative endeavours to learning. Constructionism can be regarded as an educational method based upon constructivist learning theory [9]. Whereas constructivism advocates that knowledge is constructed in the mind of the individual, constructionism extends upon this, suggesting that an effective way to learn is to build something tangible that exists in the real-world. This is thought to enhance the overall learning experience, making it more meaningful to the student. The emphasis of constructionism is the importance for students to be actively engaged in personally creating a product which is meaningful to themselves and others [9][10].

2.4 The Link between Learning and Creativity

Similarly to learning, creativity also involves the active construction of new ideas and content within the social context of other members of the field. Few scholars have suggested there exists a strong relationship between learning and creativity, however, the similarities between the two appear evidently striking[11][12]. Guilford states that creativity can be considered a sub-type of learning as expressed in the following statement: “*A creative act is an instance of learning...a comprehensive learning theory must take into account both insight and creative activity*”. The following section of this paper explores the fundamental models and perspectives on creativity theory and discussed how these models can be aligned with models of learning.

3. THEORETICAL BACKGROUND: THE CREATIVE PROCESS

3.1 Towards and understanding of the Creative Process

Creativity research originally focused upon stage models of the creative process starting with the work of Poincare in 1913. Poincare describes the creative process as commencing with conscious thought, followed by unconscious work, resulting in ‘inspiration’ [13]. Based on Poincare’s account of the creative process, Wallas formalised the four-stage model of creativity [14]. Wallas defined creativity as a linear four-stage model, progressing through the stages of preparation, incubation, illumination and verification (see figure 1). Preparation concerns immersing one’s self in a domain, and developing a curiosity about a particular problem [15]. During this stage, knowledge is consciously accumulated and influences are drawn from previous experience. During the incubation stage, conscious thought pertaining to the

problem is rested and left to the unconscious mind [16]. Illumination occurs when one experiences a sudden flash of insight [14]or sudden inspiration [13]. Finally, verification concerns forming judgments pertaining to the creative artefact produced.



Figure 1. Four stage model of creativity (Wallas, 1926)

Since the proposed model of creativity by Wallas in 1926 there have been many debates and redefinitions of the stages of creativity, however, certain points gain widespread agreement [17]. Firstly, there is a need for preparation for the creative act. Preparation can have a number of aspects, this can involve accumulating existing facts and resources and preparing mentally for the creative process. Second, time is required for the incubation of ideas. Third, the verification of creative thoughts has both a personal and social element. The new work must satisfy the aims of the individual and stand up to evaluation by a wider community.

4. THEORETICAL BACKGROUND: SPECIFYING MUSICAL REPRESENTATIONS

4.1 Children’s musical representations

Visual imagery is widely acknowledged as a crucial element of creative thinking [18], therefore it is common sense to incorporate visual imagery into the design of creative learning environments. In relation to music composition software, music has been typically specified using staff notation. However, more recent studies into children’s use of musical representations have reported that in some instances staff notation may act as an inhibitor in early music composition owing to the mis-match between the sound properties of music and the visual representation of staff notation [19][20].

Traditional music composition programs have been largely based upon the symbolic functions of traditional staff notation, thus, in some instances excluding those with little to no music experience. Scholars have suggested that alternative forms of graphical notation have proven more effective in studies where the symbols used visually reflect the properties of musical sounds [19]. Symbol systems used in this way do not rely upon retrieval of previously learned meanings as visual elements are matched to auditory elements through readily identifiable representations. This would also allow for musical tones to be represented cross-modally. For example, sounds may be described as ‘bright’, ‘dark’, ‘harsh’, ‘soft’ etc.

4.2 Children’s perceived confidence in music composition

A study conducted by Seddon indicated that children without formal music training appear to lack confidence in composition tasks if they associate their abilities to a lack of formal music training [21]. This is as opposed to higher levels of perceived confidence displayed by those with formal music training [22][23]. In terms of compositional works, results from Seddon’s study concluded that students with formal musical training indicated higher preferences for displaying musical expertise such as musical structure within their compositions. Compositions produced by those without formal music training were associated with higher preferences in terms of originality and exploration [21][24][25].

5. A GENERATIVE FRAMEWORK FOR CREATIVE LEARNING

Drawing upon insights from the background motivation, a framework has been developed which represents a distillation of creativity theory focusing upon education. The framework is presented in the form of a generative framework, which exists as a design support tool to assist with the design of lesson support materials and the design of educational technologies. The framework assists the design of creative educational experiences for the classroom by providing scaffolding for supporting materials in terms of the six white component boxes of the framework (see figure 2). Wallas’s four-stage model has been adapted as the fundamental basis for this generative framework, with the processes of preparation, generation and evaluation represented laterally across the framework. The vertical dimensions reflect individual (denoted here as personal) and social components of creativity. The social level refers to others, peers and society. Personal levels reflect explicit and tacit levels of thinking.

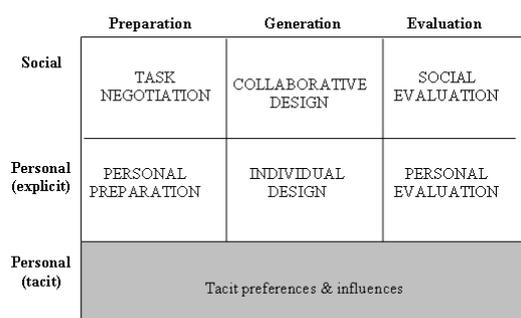


Figure 2. A generative framework for creative learning

With regard to figure 2, the lateral and vertical phases and sub-components of the generative framework are discussed within the following sub sections.

5.1 Lateral process: the preparation process

The processes of preparation, generation and evaluation are recognised herein as three integral concepts of the creative process, in that, every creative act involves the

preparation of ideas, whether in the form of tacit influences drawn from the environment, or conscious preparation for the task. Within this process, at the personal level, an individual will develop a curiosity or a desire to create. Once this desire or need has been established, information is consciously accumulated from the external environment and thoughts may be discussed with others on a ‘social’ level which the individual can reflect upon on a ‘personal’ level [15]. If working in a collaborative setting, group-wide negotiations of the task will also take place. Inevitably, the way in which an individual prepares for the task will be influenced by their past experiences which may be explicit or tacit [26].

5.2 Lateral process: the generation process

The generation process of the framework encompasses social and personal design. Within this process ideas are generated which can involve interactions and negotiations between the individual and peers in their environment. Additionally, idea generation is assisted partly by a continuous interaction occurring between levels of explicit and tacit thinking [14]. The terminology used in the creativity literature refers to these sub-conscious processes as incubation and illumination. These terms refer to the ‘incubation’ of ideas where conscious thought pertaining to the problem is rested, and ‘illumination’ is the point at which creative ideas are realised. A number of scholars suggest that influences from the environment at a ‘social’ level can trigger creative ideas to progress from tacit to more explicit thoughts at a ‘personal’ level [16]. Thus, the framework presented here acknowledges the importance of environmental factors upon the creative process, and the importance of allowing time for creative ideas to evolve.

5.3 Lateral process: the evaluation process

The evaluation process concerns reviewing early creative ideas through to evaluating the final artefact. The evaluation process may be conducted by the individual at a personal level, and by the wider community. This represents two dimensions of evaluation, a wide body of literature supports this [27][28]. Although not all creative acts culminate in historically significant acts, the creative individual may wish to verify their work with others residing within the community. This may lead to individual and or societal acceptance of the creative artefact, and in some instances, this may lead to the individual returning to earlier processes of the framework, for example for the refinement of an idea [27]. This is supported by previous studies which extend upon the work of Wallas indicating that a second incubation process may occur after initial illumination, depending on the creative idea or artefact produced [29]. Inevitably, what follows the evaluation process will differ between individuals and scenarios.

5.4 Theoretical assumptions of the framework

The generative framework for creativity attempts to explain concepts and processes involved in creativity. The creative learning process begins with social and individual preparation, and finally ends with social and individ-



Figure 5. The composition environment within SoundScape in prototypes one and two

This prototype has been designed to explicitly support the preparation phase of the creative process whilst using visual metaphors to specify music.

6.3.2 Prototype two: not explicitly supporting preparation but using visual metaphors to specify music

Prototype two does not explicitly support preparation and students enter the program at the main composition screen (generation phase of the creative process) and create a composition using pre-specified composition objects (see figure 5). The pre-specified composition objects still make use of the visual metaphors associated with the pre-selected composition theme.

6.3.3 Prototype three: explicitly supporting preparation and using abstract representation to specify music

Prototype three uses abstract representation to specify music, similar to those use in off-the-shelf music composition packages such as the E-Jay range. Preparation is explicitly supported within this prototype by allowing students to select eight music samples for use in their compositions (see figure 6). After selecting eight composition objects the students then progress to the main composition area of SoundScape and place the objects on the screen using drag-and-drop functionality (see figure 7).

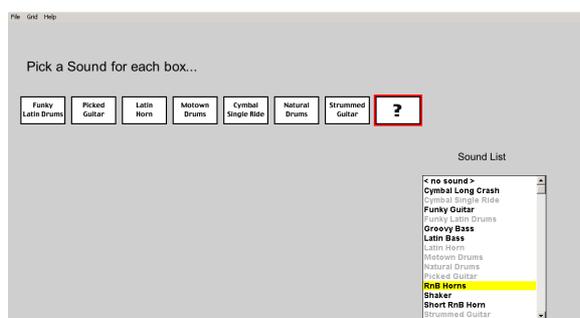


Figure 6. The composition object set up screen in prototype three

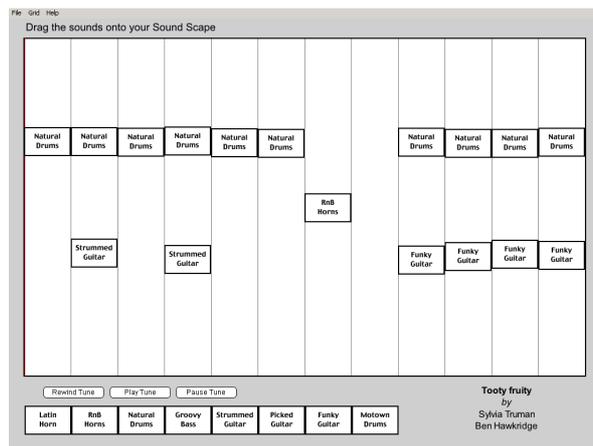


Figure 7. The composition environment within SoundScape in prototypes three and four

6.3.4 Prototype four: not explicitly supporting preparation and using abstract representations to specify music

Prototype four does not explicitly support preparation, so the students do not select composition objects to use, rather, they enter the task on the main composition screen (the generation phase of the creative process). Here they use pre-define composition objects to create their composition using the abstract representation to specify music (see figure 7).

6.4 Experimental conditions

In order to investigate the research hypotheses, each prototype was assigned to one of four experimental conditions as shown in table 1. Ninety six school children participated with this study, all eleven years of age. Twenty four participants were allocated to each condition, with twelve pairs of students in each. The study was conducted over a month and with one pair of students at a time to allow participants to work free from distraction.

6.5 Exploring student’s level of perceived confidence with the composition task

Before being introduced to the composition task, students were individually provided with a sheet of paper an A3 sheet of paper containing the question “who makes music”. This was used to elicit attitudinal responses towards the following:

1. Who the individual participant perceived to have the ability to create a piece of music.
2. What skills the individual participant perceived as necessary to create music.

This was administered prior to and after using SoundScape so that any changes in the participant’s opinions relating to the above could be compared. This was used to provide an indicator of individual student’s level of perceived confidence when approaching the task of musical composition.

6.6 Arranging groups to explore the research hypotheses using the four SoundScape prototypes

Prior to interacting with SoundScape, students were instructed: “working as a pair, create a piece of music using SoundScape. There is no right or wrong way of carrying out the task. Spend as long as you feel is necessary on your composition until you feel you have completed it.

| Con- di- tion | SS Pro- totype | Prepa- ration Support | Musical Spec | No. of Stu- dents | N o. of pa irs |
|---------------------|-------------------|-----------------------------|-----------------|-------------------------|----------------------------|
| V-P | One | Yes | Metaphor | 24 | 12 |
| V- NP | Two | No | Metaphor | 24 | 12 |
| NV- P | Three | Yes | Abstract | 24 | 12 |
| NV- NP | Four | No | Abstract | 24 | 12 |

Table 1. Experimental conditions used in the study

Data was collected during the participant’s composition session both by the program and observational behaviour analysis. The A3 sheet of paper was represented after their session with SoundScape and students were asked to add anything they felt they wanted to. Outcomes of the study are now discussed in terms of: time on task, manipulation of composition objects used, points of pairwise discussion and student’s level of perceived confidence with the composition task.

7. RESEARCH FINDINGS AND DISCUSSION

Outcomes of the study were compared across all four conditions in terms of the time spent on the composition task, the number of composition objects moved, the number of musical bars used, the number of the eight available composition objects used, the number of individual discussion points made about individual sounds, the number of individual comments made about individual pictures, the number of individual comments made about mappings (i.e. the association between the music samples and visual metaphors used), and the student’s level of perceived confidence with the task.

7.1 Time on task

Results indicate that those working within preparation conditions V-P and NV-P spent significantly longer on the task than those in non-preparation conditions V-NP and NV-NP, ($F(1, 48) = 39.734, p < 0.01$). Findings also indicate that those working with visual metaphors to specify music (i.e. groups V-P and V-NP) spent significantly longer on the task than those using abstract representations to specify music (i.e. NV-P and NV-NP), ($F(1, 48) = 4.494, p < 0.05$).

7.2 Composition object manipulations

7.2.1 Number of composition objects moved

Results indicate that those using visual metaphors moved significantly more objects than those using abstract representations to specify music ($F(1, 48) = 10.483, p < 0.05$). No significant differences were identified between preparation and non-preparation conditions.

7.2.2 Number of musical bars used

Those using visual metaphors to specify music (i.e. V-P and V-NP) used significantly more musical bars than those using abstract representations to specify music (i.e. NV-P and NV-NP), ($F(1, 48) = 10.547, p < 0.05$). No significant differences were identified when comparing preparation and non-preparation conditions.

7.2.3 Number of the available eight composition objects used

With regard to the number of the eight available composition objects used, those using visual metaphors to specify music (i.e. V-P and V-NP) used significantly more of the eight available composition objects than those using abstract representations to specify music (i.e. NV-P and NV-NP), ($F(1, 48) = 7.333, p < 0.01$). When comparing preparation and non-preparation conditions, those working within the preparation conditions used significantly more of the eight available objects than those in conditions in which preparation was not explicitly supported ($F(1, 48) = 5.794, p < 0.05$).

7.3 Pair wise discussion points

7.3.1 Discussions on individual sounds

In terms of the discussion points that took place within the pairs as students worked on their composition together, results indicate that those using abstract representations to specify music (i.e. NV-P and VN-NP) made significantly more comments about individual sounds than those using visual-metaphors to specify music (i.e. V-P and V-NP), ($F(1, 48) = 6.305, p < 0.05$). No significant differences were identified when comparing sound discussion points across preparation and non-preparation conditions.

7.3.2 Discussions on individual pictures

With regard to discussion points concerning individual pictures (for groups using visual metaphors to specify music) those within the preparation condition V-P made significantly more comments about individual pictures than participants within the non-preparation conditions V-NP, ($t(22) = 2.732, p < 0.05$).

7.3.3 Discussions on individual ‘mappings’

With regard to the mapping discussion points (for groups using visual metaphors to specify music), those in the

preparation condition V-P made significantly more mapping comments than those within the non-preparation condition V-NP, ($t(22) = 3.815, p < 0.01$). The overall findings are summarized in table 2.

| Outcome | Task Support | Representation |
|----------------------------------|--|--|
| Time on Task | Those in preparation conditions spent longer on the task | Those in visual metaphor conditions spent longer on the task. |
| No. of objects moved | No significant differences. | Those in visual metaphor conditions moved more composition objects. |
| No. of bars used | No significant differences. | Those in the visual metaphor conditions used more of the musical bars. |
| No. of eight objects used | Those in preparation conditions used more of the eight objects | Those in visual metaphor conditions used more of the eight objects. |
| No. of sound discussion points | No significant differences | Those in abstract representation conditions made more comments about sounds. |
| No. of picture discussion points | Those in the preparation condition made more picture comments. | N/A |
| No. of mapping discussion points | Those in the preparation conditions made more mapping comments | N/A |

Table 2. A comparison of overall findings from the study

7.4 Student's perceived level of confidence with the composition task

In response to the question "who makes music" handed to participants on an A3 sheet of paper prior to their interaction with SoundScape, participants named types / groups of people who they felt made music. These types were:

- Anyone
- Composers
- Record producers
- Recording artists / pop stars
- People who can play musical instruments
- People who can read music.

The responses were analysed across all conditions. Findings indicate that at time 1, 63% of participants from V-NP stated that "anyone" could make music, V-P and NV-NP conditions made the least number of comments concerning this at 50% of participants in both conditions. A large number of comments were made overall concerning musical skills such as playing an instrument and reading music across all conditions. Participants from V-P and V-NP conditions made more comments about musical skills.

Following their interaction with SoundScape, participants were asked to add any comments they felt necessary to their A3 sheet. A change in responses was noted in the participant's opinions following participant's exposure to SoundScape, with a higher number of participants from all four conditions commenting that "anyone" can make music. Other responses made after interaction also related to the question 'who makes music', with an emphasis on musical skills decreasing across all conditions. This is especially noticeable in the changes of opinion concerning skills initially perceived as prerequisites to music composition.

8. CONCLUSIONS

In answer the original question posed by this paper "how can learning tasks be structured to encourage creative thinking in the classroom?" This study has sought to provide a solution to this via the presentation of a generative framework. The design of educational technologies, including music composition technologies can be guided by this framework. This paper has also demonstrated the application of this framework through the design of SoundScape, a children's creative music composition environment. SoundScape was used as a vehicle to test two research hypotheses, the first focusing upon the effects of providing explicit support for the preparation phase of the creative process using music technology as opposed to not, and, secondly, focusing upon the effects of using visual metaphors to specify music as opposed to abstract representations. Findings from the study indicated that preparation is a crucial element of the creative process and that support preparation in music composition can assist to encourage creative thinking in children's music composition. Outcomes also suggest that the use of visual imagery to specify music is also a useful tool for learning, especially where the imagery used is consistent with real-world artifacts.

Acknowledgments

The author would like to acknowledge and extend appreciation to Ben Hawkridge (Knowledge Media Institute The Open University, UK) for his assistance in developing the SoundScape program, and to the staff and students at Heronsgate Middle School in Milton Keynes, UK for participating with this study.

9. REFERENCES

- [1] Jarvinen, E.M (1998) The LEGO/LOGO learning environment in technology education: an experiment in a Finnish context. *Journal of Technology Education*. 9, 2. p 47 – 59.
- [2] Edwards, D & Mercer, N (1987) *Common Knowledge: The Development of Understanding in the Classroom*. London. Methuen.
- [3] Brown, Collins & Duguid (1989) *Situated Cognition and the Culture of Learning*. *Educational Researcher*. January – February. 32-43

- [4] Baccarini, D (2004) The implementation of authentic activities for learning: A case study. Proceedings of the 13th Annual Teaching Learning Forum. 9-10 February. Perth: Murdoch University, Australia.
- [5] Bandura, A (1986) Social Foundations of Thought and Action: A Social Cognitive Theory. Englewood Cliffs; NJ Prentice Hall.
- [6] Frank, C (2005) Teaching and learning theory: who needs it? College Quarterly. No. 2, Vol 8.
- [7] Dewey, J (1916) Democracy and Education. New York. Macmillan.
- [8] Stager, G (2005) Towards a pedagogy of online constructionist learning. Proceedings of the 2005 World Conference on Computers in Education. Stellenbosch, South Africa.
- [9] Papert, S (1993) The Children's Machine: Rethinking School in the Age of the Computer. New York; Basic Books.
- [10] Harel, I (1991) Children Designer's: Interdisciplinary Constructions for Learning and Knowing mathematics in a Computer Rich School. Norwood; NJ. Ablex Publishing.
- [11] Guilford, J. P (1950) Creativity. American Psychologist. 5. p 444 – 454.
- [12] Karnes, M. B, McCoy, G.F, Zehrbach, R.R, Wollersheim, J.P, Clarizio, H.F, Costin, L & Stanley, L.S (1961) Factors Associated with Overachievement of Intellectually Gifted Children. Champaign, IL.
- [13] Poincare (1913) in Leytham, G (1990) Managing Creativity. Norfolk. Peter Francis Publishers.
- [14] Wallas (1926) The Art of Thought. London; Johnathan Cape [republished in 1931].
- [15] Getzels, J.W (1964) Creative Thinking, Problem-Solving, and Instruction. In E.R. Hilgard (Ed), Theories of Learning and Instruction. Chicago; University of Chicago Press.
- [16] Claxton (1998) Hare Brain Tortoise Mind: Why Intelligence Increases When you Think Less. London. Fourth Estate Limited.
- [17] Osborn, A.F. (1953). Applied Imagination (Revised Ed.). New York; Scribners.
- [18] Gruber, H. E & Wallace, D. B (1999) The case study method and evolving systems approach for understanding unique creative people at work. In Sternberg, R. J (Ed) Handbook of Creativity. pp 93 – 115. Cambridge UK; Cambridge University Press.
- [19] Walker, R (1992) Auditory-visual perception and musical behaviour. In Colwell, R. (Ed) Handbook of Research on Music Teaching and Learning. New York; Schirmer Books.
- [20] Auh, M (2000) Effects of using graphic notations on creativity in composing music by Australian secondary school students. Proceedings of the Australian Association for Research in Education Conference. Australia 2000.
- [21] Seddon, F.A. (2002) The relationship between instrumental experience, adolescent self-perceived competence in computer-based music composition and teacher evaluation of composition. In Stevens, C., Burnham, D., McPherson, G., Scubert, E. and Renwick, J. (Eds.) Proceedings of the seventh international conference on music perception and cognition, Sydney 2002, Adelaide: Casual Productions.
- [22] Seddon, F.A. and O'Neill, S.A. (2001) Creative thinking processes in adolescent composition: an interpretation of composition strategies adopted during computer-based composition', paper presented at The Second International Research in Music Education Conference, University of Exeter, April 2001.
- [23] Folkestad, G., Hargreaves, D. J., & Lindström. (1998). Compositional strategies in computer-based music making. British Journal of Music Education, 15(1), 83-97.
- [24] O'Neill, S & Sloboda, J (1997) The effects of failure on children's ability to perform a musical test. Psychology of Music. 25, 1. pp 18 – 34.
- [25] Webster, P, Yale, C & Haefner, M (1988) Test-retest reliability of measures of creative thinking in music for children with formal music training. MENC National In-Service Meeting. Indianapolis, Indiana.
- [26] Schank, R (2002) Designing World Class E-Learning. New York. McGraw-Hill.
- [27] Amabile, T.M (1996) Creativity in Context. Boulder, CO; Westview.
- [28] Csikszentmihalyi, M (1999) Implications of a systems perspective for the study of creativity. In Sternberg, R.J (Ed) Handbook of Creativity. Cambridge; Cambridge University Press. P 313-335
- [29] Leytham, G (1990) Managing Creativity. Norfolk; Peter Francis Publishers.
- [30] Runco, M.A & Bahleda, M.D (1986) Implicit theories of artistic, scientific and everyday creativity. Journal of Creative Behaviour. 20. p 93-98.
- [31] R.J (1985) Implicit theories of intelligence, creativity and wisdom. Journal of Personality and Social Psychology. 49. p 607 – 627.

WHEN SOUND TEACHES

**Serena Zanolla, Antonio Rodà,
Filippo Romano, Francesco Scattolin,
Gian Luca Foresti**

University of Udine

{serena.zanolla, antonio.roda}@uniud.it
romano.filippo@spes.uniud.it
scattolin.francesco@gmail.com
gianluca.foresti@uniud.it

Sergio Canazza

University of Padova

canazza@dei.unipd.it

ABSTRACT

This paper presents the Stanza Logo-Motoria, a technologically augmented environment for learning and communication, which since last year we have been experimenting in a primary school; this system offers an alternative and/or additional tool to traditional ways of teaching that often do not adapt to the individual learning ability. The didactic use of interactive multimodal systems, such as the Stanza Logo-Motoria, does not replace the teacher; on the contrary this kind of technology is a resource which offers greater access to knowledge and interaction with others and the environment. This is possible by inventing systems and activities, which bring out inherent values in using technology and in its integration in learning processes. The aim of this paper is to document activities carried out by Resonant Memory, the first application of the Stanza Logo-Motoria, and the relative experimental protocol that we are implementing. In addition, we are going to introduce a new application of the system, the Fiaba Magica, for strengthening gesture intentionality in children with motor-cognitive impairments.

1. INTRODUCTION

The Stanza Logo-Motoria is an interactive space, permanently installed at a school, which allows the assimilation of content by “learning in movement”. We started to develop the Stanza Logo-Motoria in 2009 and, on the occasion of the 7th SMC2010 Conference in Barcelona, we presented the first application of the system: the Resonant Memory [1]. From then on, we a) established a validation protocol for its scientific experimentation, b) experimented the Stanza Logo-Motoria at school for a year, c) made the system more user-friendly and flexible and d) developed new applications. We are experimenting the Stanza Logo-Motoria as an innovative way of teaching, which, for scientific research purposes, has also been installed at the Engineering Information Department (DEI) of the University of Padova.

Copyright: ©2011 Serena Zanolla et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The topic of this work is inherently multidisciplinary: the Stanza Logo-Motoria, an interactive and multimodal environment for learning and communication, is used at school in order to discover and emphasize the enactive approach of learning. By means of new technologies and following the enactive approach we think that it is possible to help school teachers to deal with specific learning difficulties of pupils, from dyslexia to severe motor-cognitive disabilities. In actual fact, these children need to learn by means of alternative methodologies and tools often specifically designed for them. For the child with severe disabilities, it is important to focus on communication skills in order to enhance their quality of life, which can be measured calculating the amount of participation in everyday activities [2]. In order to improve these children’s communication skills, they should be able to choose and control their social environment. With this in mind, we have developed a new application for the Stanza Logo-Motoria, called Fiaba Magica, which we are using and enhancing at the same time, with the aim of designing a specific validation protocol.

1.1 Theoretical foundations

The main theories about a child’s cognitive development recognize that during infancy the first modality of reality representation is enaction: learning through perception-action-interaction within the environment [3]; this interaction is multimodal because it involves all the senses. There are three ways of organizing knowledge corresponding to three forms of interaction with the world: enactive, iconic and symbolic [4]. Enactive knowledge is based on motor skills: enactive representations are gained “by doing” and, in the enactive context, “doing” is the tool for learning. So enactive interaction is direct, natural and intuitive. If cognition is the process whereby a living organism, interacting with its environment, brings forth, or enacts the world in which he lives, the action is considered as a prerequisite for perception and the sensory input has meaning in relation to effectuated actions. In the enactive approach sensory inputs are used to guide actions, [5] which modify the environment, and/or the relation of the organism to its environment, and hence modify in return the sensory input. Gardner, in the theory of multiple intelligences [6], argues that teaching methods, often based only on logical-

mathematical and linguistic intelligences, disregard whoever uses other cognitive modalities. Gardner believes that schools have to use a multimodal methodological approach. This theory validates teachers' everyday experience: students think and learn in many different ways. It also provides teachers with a conceptual framework for organizing and reflecting on curriculum assessment and pedagogical practices. In turn, this reflection has led many teachers to develop new approaches that might better meet learners needs in their classrooms.

In the neurosciences field, the discovery of mirror neurons [7] confirms the motor aspect of cognition: learning is the action performed inside the environment. Many object-related actions can be recognized by their sound through audio-visual mirror neurons that code actions independently of whether actions are performed, heard or seen. These neurons [8] have a) the capacity to represent action contents and b) the auditory access to contents of human language.

Technological systems used in the educational field often do not emphasize the interactivity component that promotes involvement, social interaction and collaboration [9]. Research in the field of Interaction Design rather testifies the potentiality of multimodal interactive systems for learning [10], cooperation [11] and interactive storytelling [12].

The disciplines of music therapy and music technology are recent but in the past few years, by means of multiple-media technology, it has been possible to combine them in order to allow people with severe disabilities to gain access to real-time audiovisual interaction. Several important projects such as Care Here¹ and Mediate Project² have already used interactive multiple-media technology to improve people's motor and mental skills by producing audiovisual-tactile environments. The main aim of these projects is to develop technological systems whose quality of interaction is so high that the users are not aware that they are using the technology [13].

1.2 The Stanza Logo-Motora

The Stanza Logo-Motora is a system with which it is possible to create an Augmented Reality environment [14]; the system takes a broad empty space and, using video tracking techniques, fills the environment with sounds and images. The user's presence inside the "reactive space" triggers the playback of sounds semantically connected with the topic of the lesson.

The Stanza Logo-Motora has been developed by using the EyesWeb XMI platform³; for the overall system architecture see [1]. In the Stanza Logo-Motora (fig. 1) a webcam is used to acquire the movement of the body in space and gestures; the image stream is processed by an EyesWeb application and a number of low-level features are extracted (e.g. position, velocity and acceleration of the centre of mass) [15]. Space analysis starts from the trajectory followed by each user in the reactive space. The system allows us to define a number of regions within the space and

synchronizes the user's occupation of a zone with the playback of the audio/video content.

The arrangement of sounds in space captures the pupils' attention and allows them to spatially organize the knowledge acquired. In this way teachers can furnish students with a broad context for understanding the real world and students are more likely to comprehend and remember what they are learning. By exposing students to an experiential, explorative and authentic model of learning, the Stanza Logo-Motora, may help them shift from passive to active learning modes and thus become more successful learners.

In the following paragraphs we will explain in detail two applications of the Stanza Logo-Motora called Resonant Memory and Fiaba Magica. In particular in Sec. 2 the Resonant Memory application is briefly introduced, Sec. 2.1 describes its use while Sec. 2.2 presents the validation protocol we are following. Sec. 3 explains fully a new concrete instance of Stanza: the Fiaba Magica application, Sec. 3.1 its aims and Sec. 3.2 the first results of its experimentation. The conclusions are drawn in Sec. 4.

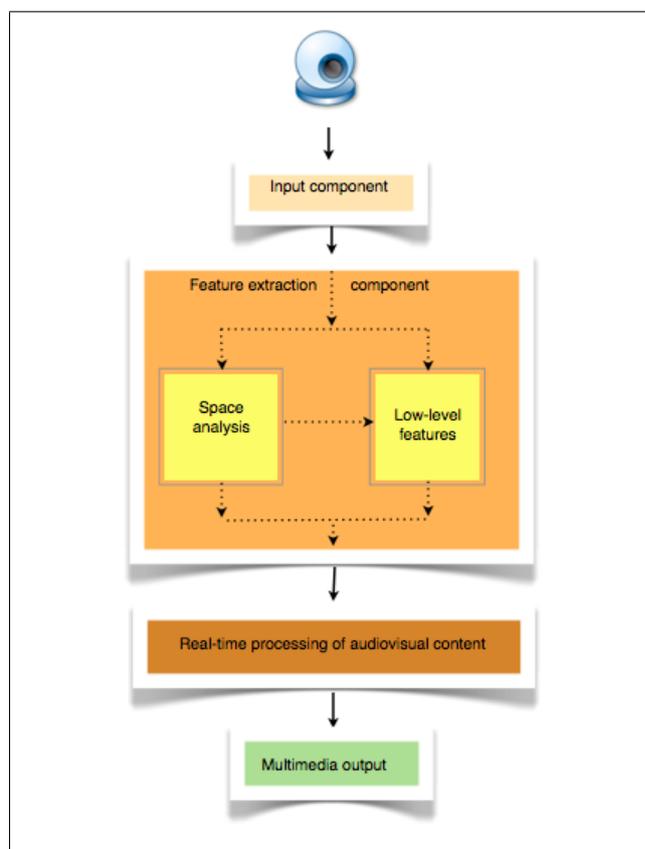


Figure 1. System architecture of the Stanza Logo-Motora.

2. THE RESONANT MEMORY APPLICATION

The Resonant Memory application allows the creation of a technologically augmented environment [14] to be used within all the subjects taught at school. To explain clearly how the Resonant Memory application works, here are the main points. The space captured by a webcam is divided into nine areas: eight of these are peripheral whereas the

¹ <http://www.bris.ac.uk/carehere>

² <http://www.port.ac.uk/research/mediate>

³ www.infomus.org

ninth is central; the number of areas may vary depending on the didactic needs. Sound information corresponds to each area. The trajectory of the user's barycentre is used to match a sound to a specific position in space. A child explores the "resonant space" in which he/she can freely move without using sensors:

- Noises, environmental sounds, and music are associated with peripheral zones and are reproduced when the child reaches and occupies a peripheral zone.
- The central area is synchronized instead with an audio reproduction of the contents to be taught that contain the elements to be connected with sounds positioned in the various peripheral areas.

The child, listening to the auditory content, enjoys searching for the sounds heard before and, at the same time, he/she creates the soundtrack of the lesson. For the Resonant Memory system architecture see [1].

2.1 Use of the Resonant Memory application

Currently, school teachers use the Resonant Memory application to manage lessons in an alternative way, e.g. as a tool to activate the written production of a tale. In this case, the teacher chooses eight sounds and collocates them in the peripheral zones of the Stanza Logo-Motoria. The child, by exploring the physical and auditory space, is encouraged to invent a tale using those sounds. Then, the tale invented by the child is converted into an audio file and located in the central zone of the Stanza Logo-Motoria. Finally the child "reads the story again", moving by himself within the sound-augmented space.

The Resonant Memory application is used also in order to study a school subject such as History: the teacher records the text of the lesson and puts it into the central area; whereas the sounds connected to the recorded text are collocates in the peripheral areas. The child listens to the content reproduced in the central area, reaches the different peripheral areas experimenting with the sounds and finally, enjoying the game, "fills the content of the lesson with sounds". Teaching contents, by becoming "physical events" which occur around the child by means of the child, activate the motor aspect of knowledge [16]. The environment has acquired, in this case, a dynamic role: it becomes a dynamic space, a space of knowledge and logic, a map where knowledge grows. The Stanza Logo-Motoria becomes a map of knowledge: the spatial map of knowledge.

We are also experimenting the use of the Resonant Memory application as a tool to develop the spatial ability on the part of students with severe visual impairments. We are checking if it is possible to use the Stanza Logo-Motoria as a means to develop orientation and mobility training for blind people. If movement is a basic element for learning, at the time that a child physically discovers his world, then learning takes place. Children with visual impairments typically need encouragement to explore the environment. To them the world may be a surprising and unpredictable place, but also non-motivating. Orientation and mobility training usually helps a blind or visually impaired child to

know where he/she is in space and where he/she wants to go (orientation). It also helps him/her to be able to carry out a plan to get there (mobility). It is important to begin to develop orientation and mobility skills in infancy, and continue during their adulthood in order to improve autonomy in moving [17]. This is the reason why we are experimenting the Stanza Logo-Motoria with pupils and adults, as a tool to develop interactive training for mobility and orientation, by working in the following way: starting from the centre of the Stanza Logo-Motoria, the user has to move in a certain direction to trigger a specific sound. He/she can move forwards, backwards, towards the right and left, diagonally forwards and backwards. The sound informs the user if he/she is following the right direction. When the user keeps to the correct direction, the sound represents an auditory reinforcement. Before carrying out this task, the child with visual impairments has to memorize the spatial localization of sound by means of a tactile map and/or the physical exploration of space.

2.2 Validation protocol of the Resonant Memory application

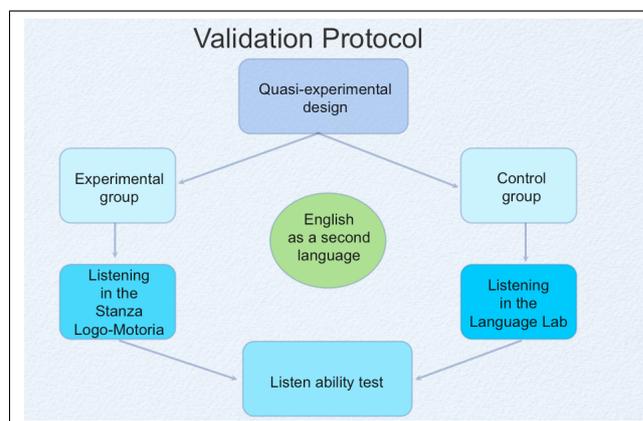


Figure 2. Validation protocol of the Stanza Logo-Motoria.

At school we have been testing (fig. 2) the Stanza Logo-Motoria in Resonant Memory modality since February, following a quasi-experimental design [18] (between subjects) with two comparable classes: two Third Classes. The quasi-experimental design requires a pre-test (February), an intermediate-test (April) and a post-test (June) for a treated (experimental) and comparison (control) group. Every test consists of two tasks: "listen and tick the right picture" and "listen, draw and color". We intend to verify (experimental hypothesis) if pupils, by using the Stanza Logo-Motoria as a listening tool for learning English as a second language, improve significantly in word recognition and language comprehension than those who use passive listening by means of loudspeakers. The dependent variable is: a significant improvement of listening comprehension ability in English as a second language. The independent variable is: use of the Stanza Logo-Motoria, in Resonant Memory modality, as listening tool. For two hours a week the control group is using the language laboratory equipped with two loudspeakers while the experimental

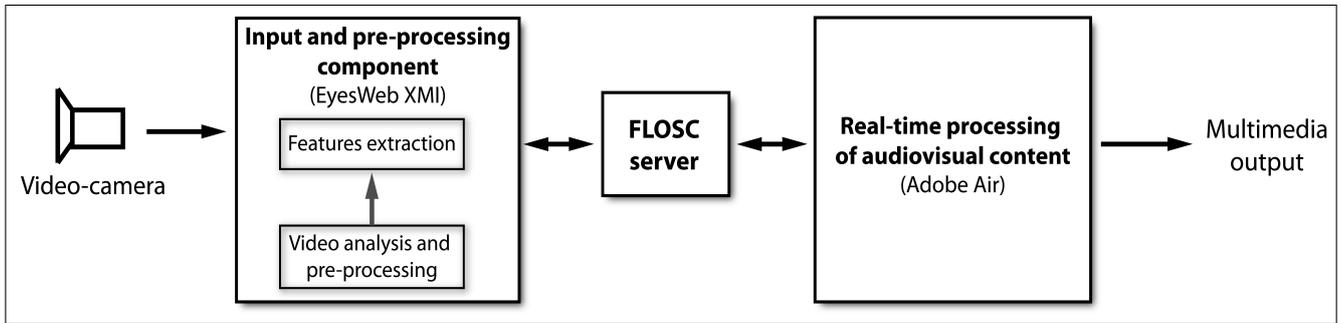


Figure 3. System Architecture of the Fiaba Magica application.

group is using the Stanza Logo-Motoria.

3. THE FIABA MAGICA APPLICATION

In Fiaba Magica, the space captured by the web-camera is divided into three areas (fig. 4); each area is synchronized with a) the audio reproduction of a sequence of a tale and b) the screen projection of the corresponding image. The video stream is processed in order to extract several low-level features related to the user's movements. Background subtraction is achieved via a statistical approach: the brightness/chromaticity distortion method [19]. Extracted features include the trajectory of the centre of mass, the Motion Index, and the Contraction Index. Using these parameters, the system recognizes the following actions performed by the user:

- raising their arms laterally one at a time;
- their body position in space.

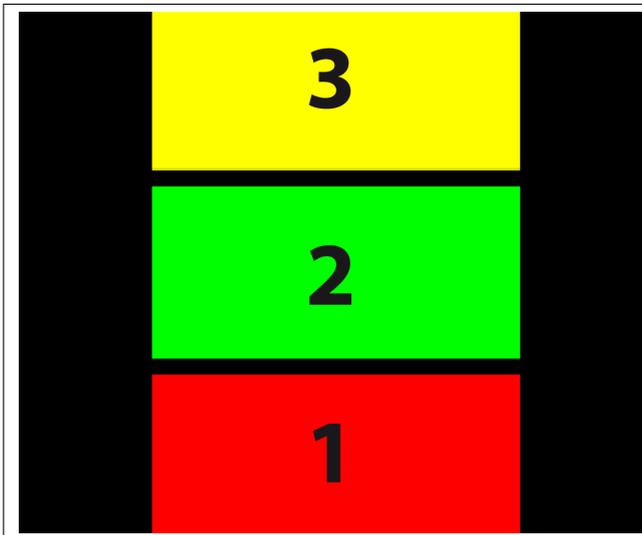


Figure 4. How the space is organized by the Fiaba Magica application.

When the user reaches the first zone, the Fiaba Magica application a) triggers the projection of two characters on the screen and b) activates the audio reproduction of the first part of the tale. During this phase no other event is triggered. After having listened to the first audio track, the

user animates the corresponding character on the screen by raising his/her arms laterally, one at a time. The graphic animation consists of two "animated talking characters" (fig. 5). Once both characters have been animated, the user reaches the second area to listen to and see the second part of the audio/video tale. Consequently, two other characters to be animated appear on the screen. The interaction modality in the second and third zones is "managed" as well as the first one.

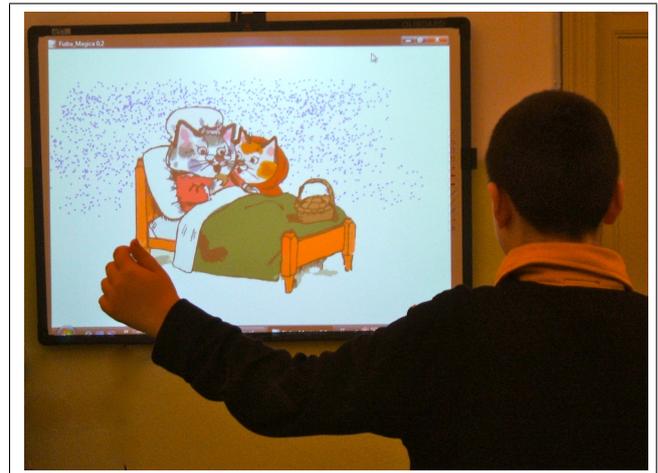


Figure 5. The user animates the corresponding character on the screen by raising his arms laterally, one at a time.

Figure 3 shows the overall system architecture of the Fiaba Magica. It consists of two major components described below.

1. The input and processing component which receives the video stream captured by the webcam observing the space; this component is responsible for:
 - processing video data (e.g. denoising and background subtraction techniques to extract the user's silhouette);
 - the motion feature extraction which enables a) the analysis of input data in order to get information about how the user occupies the space (e.g. where he/she goes; how long he/she remains in a given area) and b) the analysis of gesture features.

A software patch developed in the EyesWeb XMI environment performs the video analysis and feature extraction tasks.

2. The real-time processing of audio visual content component, which is responsible for the real-time control and processing of audio/video material, depends on the features extracted in the feature extraction stage. This task is performed by an Adobe Air application that also provides a GUI to configure the system.

A Flosc server allows bidirectional communication between EyesWeb and Adobe Air. Flosc is a communication gateway, which enables communication between Adobe Flash and any software recognizing UDP data. Flosc a) converts Actionscript data into OSC format, via XMLSocket of Actionscript, and b) sends out the OSC data in UDP format at the other end of the server.

3.1 Use of the Fiaba Magica application

We use the Fiaba Magica application to explore how augmented reality can provide positive and enjoyable leisure experiences for pupils with severe disabilities. Due to limitations in their physical and mental abilities, these children have few opportunities to engage in independent leisure activities. This lack of opportunity often leads to the development of dependent behavioral patterns and learned helplessness [2]. At school, for children suffering from severe disabilities, there are specially trained teachers who have to teach them sign language, how to walk, keep their balance, feed themselves, and exercises that improve their muscle co-ordination and their speech. Simple activities like brushing their teeth, arm movements, cutting with scissors, writing or drawing are taught to make these children self-reliant. By interacting with a multimodal environment, their self-esteem and sense of self-empowerment potentially increase. With the Fiaba Magica application, the teacher could teach spatial concepts, personal awareness of the different body parts, how to move their body in relation to others and objects and the understanding of where the body is in the environment.

The Fiaba Magica application is also used to help strengthen the gestural intentionality of children with multi-disabilities. Often these children express communicative intentionality only by means of simple gestures and vocalizations, which can be enhanced thanks to technology. Fiaba Magica “augments gestures” by synchronizing movement with visual and sound stimuli, in order to bring out the intentional feature of action [20].

Since the children suffering from severe disabilities cannot walk properly and require a wheelchair or supports, the Fiaba Magica application can cater for two users at a time, or a user in a wheelchair accompanied by a helper.

3.2 Early assessment of the Fiaba Magica application

Currently two pupils with severe impairments are experimenting the Fiaba Magica application:

1. A 6-year-old child with left-sided hemiparesis (paresis on the left side of the body) who cannot walk and

uses a wheelchair; he has learning difficulties due to a lack of development of motor skills.

2. A 12-year-old girl with cerebral palsy - a persistent, but not unchanging, disorder of posture and movement caused by a non-progressive disorder of the brain - which includes the loss of physical capabilities for walking, standing and sitting (disequilibrium syndrome), using hands and speaking abilities.

Up to now we have observed that both pupils have the opportunity to:

- be motivated to move in space;
- improve their voluntary movement of arms laterally, which - in such a situation - is disturbed by synkinetic movements (involuntary muscular movements accompanying voluntary movements) in order to develop the intentionality of gesture;
- enhance their capability to hold their head up high in order to follow the images with eyes;
- improve their balance in standing and walking;
- increase their attention span;
- interiorize the temporal concepts (before, now, after).

In particular, the child with hemiparesis, who is also in a wheelchair, has used the Fiaba Magica application with the help of his teacher; whereas the mother physically supported the girl with cerebral palsy. These users demonstrated an exceptional degree of enthusiasm, fun and enjoyment during each augmented reality experience. It will soon be possible to measure, by video analysis, the quantity and quality of upper extremity movement over a long period of time in order to check if they have had functional improvements in motion.

4. CONCLUSIONS

The Stanza Logo-Motoria, by using standard hardware and simple strategies of mapping, has sparked the interest of students and teachers in more innovative ways of learning and teaching. The system is suitable for the school environment thanks to its easy implementation; moreover, teachers are immediately involved in the design of activities due to the simplicity of mapping, which makes it instantly comprehensible. In fact, for over a year now, the use of the Stanza at school has shown that, by using the same basic scheme, it has been possible to develop in collaboration with teachers a great deal of educational activities involving several school subjects, such as English, History, Science, Music.

For children with severe disabilities, such as hemiparesis, cerebral palsy and blindness, who are placed in the Stanza Logo-Motoria, it is possible to develop particular learning paths by means of specific applications of the system, which can meet their needs of autonomy and communication with others and the environment.

These early results truly convince us that this is but the beginning of a path, which could lead to the introduction of technologically augmented learning in schools.

Acknowledgments

The authors would like to thank the InfoMus Lab of Genova University (Italy), and in particular, Antonio Camurri, Gualtiero Volpe, Corrado Canepa and Paolo Coletta who contributed to this work. We are grateful for their collaboration during preliminary development of the Stanza Logo-Motoria.

5. REFERENCES

- [1] A. Camurri, S. Canazza, C. Canepa, G. L. Foresti, A. Rodà, G. Volpe, and S. Zanolla, "The stanza logomotoria: an interactive environment for learning and communication," in *Proceedings of SMC Conference 2010, Barcelona*, 2010.
- [2] J. Sigafoos, M. Arthur-Kelly, and N. Butterfield, *Enhancing Everyday Communication for Children with Disabilities*. Baltimore: Paul H. Brookes, 2006.
- [3] J. Bruner, *Processes of cognitive growth: Infancy*. Clark University Press, Worcester, MA, 1968.
- [4] —, *Toward a theory of instruction*. Belknap Press of Harvard University Press, 1966.
- [5] H. Marturana and F. Varela, *Autopoiesis and cognition: The realization of the living*. Reidl, 1980.
- [6] H. Gardner, *Frames of Mind: The Theory of Multiple Intelligences*. Basic, 1983.
- [7] G. Rizzolatti and L. Voza, *Nella mente degli altri. Neuron specchio e comportamento sociale*. Zanichelli, 2008.
- [8] E. Kohler, C. Keysers, M. Umiltà, L. Fogassi, V. Gallese, and G. Rizzolatti, "Hearing sounds, understanding actions: Action representation in mirror neurons," *Science*, vol. 297, no. 5582, pp. 846, 848, August 2002.
- [9] E. Hornecker and J. Buur, "Getting a grip on tangible interaction: A framework on physical space and social interaction," in *CHI 2006*, 2006, pp. 437, 446.
- [10] A. Camurri, B. Mazzarino, S. Menocci, E. Rocca, I. Vallone, and G. Volpe, "Expressive gesture and multimodal interactive systems," in *Proceedings of the AISB 2004 Convention: motion, emotion and cognition*, 2004.
- [11] M. L. Guha, A. Druin, and J. A. Fails, "How children can design the future," in *HCIL - 2011 - 04*, 2011.
- [12] J. Montemayor, A. Druin, G. Chipman, A. Farber, and M. L. Guha, "Tools for children to create physical interactive storyrooms," *ACM Computer in Entertainment*, vol. 2, no. 1, January 2004.
- [13] A. Hunt, R. Kirk, and M. Neighbour, "Multiple media interfaces for music therapy," *IEEE Multimedia*, vol. 11, no. 3, pp. 50–58, July-Sept 2004.
- [14] R. Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355, 385, August 1997.
- [15] A. Camurri, B. Mazzarino, and G. Volpe, "Analysis of expressive gestures: The eyesweb expressive gesture processing library," in *Gesture-Based Communication in Human-Computer Interaction*. Springer Verlag, 2004, vol. LNAI 2915, pp. 460–467.
- [16] M. Leman, *Embodied Music Cognition and Mediation Technology*. The MIT Press, 2007.
- [17] C. Martinez and K. Moss. (1998) Orientation and mobility training: The way to go. [Online]. Available: <http://www.tsbvi.edu/Outreach/seehear/fall98/waytogo.htm>
- [18] D. Campbell and J. Stanley, *Experimental and Quasi-Experimental Designs for Research on Teaching*. In *Handbook of Research on Teaching*. Chicago: Rand McNally, 1963, ch. 5.
- [19] T. Horprasert, D. Harwood, and L. Davis, *A Robust Background Subtraction and Shadow Detection*. In 4th ACCV, Taipei, Taiwan, 2000, vol. 1.
- [20] J. Searle, *Intentionality. An essay in the philosophy of mind*. Cambridge University press, 1983.

LJUDSKRAPAN/THE SOUNDSCRAPER: SOUND EXPLORATION FOR CHILDREN WITH COMPLEX NEEDS, ACCOMMODATING HEARING AIDS AND COCHLEAR IMPLANTS

Kjetil Falkenberg Hansen

KTH Royal Institute of Technology
kjetil@kth.se

Christina Dravins

Rīga Stradiņš University
kristina.dravina@inbox.lv

Roberto Bresin

KTH Royal Institute of Technology
roberto@kth.se

ABSTRACT

This paper describes a system for accommodating active listening for persons with hearing aids or cochlear implants, with a special focus on children with complex needs, for instance at an early stage of cognitive development and with additional physical disabilities. The system is called *Ljudskrapan* (or *the Soundscrapper* in English) and consists of a software part in Pure data and a hardware part using an Arduino microcontroller with a combination of sensors. For both the software and hardware development, one of the most important aspects was to always ensure that the system was flexible enough to cater for the very different conditions that are characteristic of the intended user group.

The Soundscrapper has been tested with 25 children with good results. An increased attention span was reported, as well as surprising and positive reactions from children where the caregivers were unsure whether they could hear at all. The sound generating models, the sensors and the parameter mapping were simple, but provided a controllable and complex enough sound environment even with limited interaction.

1. INTRODUCTION

This paper describes a system for promoting listening and exploration of sounds based on playful audio interaction. The system is devised for children at an early stage of cognitive development with a focus on children with hearing impairment and complex needs.

The aim is to reduce limitations of activity due to functional disability and to encourage curiosity toward active listening by providing good conditions for exploring, investigating and playing with a collection of synthesized and recorded sounds. The system, called “Ljudskrapan” in Swedish and “the Soundscrapper” in English, consists of a software and a hardware part. The software is programmed in Pure data [1], and various sensors are used for interacting. A typical setup includes an Arduino [2] with gesture tracking sensors attached, but other input devices like game

controllers, cameras and pointing devices can be connected.

In the typical envisaged use scenario, the child interacts with the Soundscrapper through gestures in a school or a clinic, supervised by a special needs teacher or a speech therapist, and with the software operated by a second person. In a simpler use case, a user interacts with and operates the software without supervision. In both cases, the Soundscrapper can be classified as a sound exploration tool, a sound toy and musical instrument.

The Soundscrapper has ambitions to empower children with hearing impairment and complex needs in the following ways:

- make exploration of sound and music accessible,
- stimulate active listening in a playful and rewarding way,
- be inclusive by using easily adaptable sensor data for control,
- provide means for expanding orientation by listening and promoting audition as a means of orientation and communication.

The aim is also to offer novel possibilities for assessing hearing capabilities in children at an early stage of development. An important function is to improve the possibilities for the child to provide clear and unambiguous feedback to caregivers.

These are by no means modest goals, but we believe that the method may offer many possibilities to support the development of listening capabilities for children with complex needs and hearing impairment. The advantage with the proposed system is that the child can influence listening and choose which sounds are interesting, thus promoting active participation and supporting independence.

Pilot interventions and the first tests have been promising, and the potential future users have expressed a wish for further development of the Soundscrapper. As will be discussed in the following section, the situation for many of the children we target is such that measuring the success of our intervention is impracticable or impossible in a short time perspective. The focus in this paper is on the conceptual framework and technical description, illustrated with a few examples from the first two years of small-scale testing.

1.1 Background

Young children with impaired hearing are often provided with hearing aids at an early age. The most common form of hearing impairment, sensorineural hearing loss, is caused by lack of sensory cells in the inner ear. If there is sufficient function in the inner ear an acoustic (traditional) hearing aid (HA) can be fitted so the remaining sensory cells are stimulated in an optimal way. With profound hearing loss it is not possible to enhance the acoustic input to create a hearing experience. In these cases a cochlear implant (CI) can be used to make hearing possible.

1.1.1 Cochlear implants and hearing development

A CI consists of an electrode array that is surgically inserted into the inner ear (cochlea). The electrode array is fed with an electrical stimulus pattern generated by an externally worn sound processor. The sound processor is programmed to convert the acoustic signals into an electrical pattern that is continuously forwarded by the electrode array to the hearing nerve, and then further on into the central auditory system. The auditory processes in the brain convert the synthetically produced signal pattern into a meaningful listening experience.

CI was initially designed for persons who had become deaf after acquiring spoken language, but the technology has later been successfully introduced to deaf born children as well. The surgical procedure by which the electrode array is inserted into the inner ear is often performed when the child is between six and nine months of age. The challenges are, however, quite different in the pediatric population as hearing development depends on physiological maturation as well as exposure to stimulation.

In a typically developing child, hearing is a gradually emerging skill underlying the development of spoken language. By the age 9–12 months, the majority understands several spoken words and expressions. This means that a child awaiting CI surgery is delayed several months in comparison to their hearing peers. In many cases, implantees go on developing hearing and spoken language with little or only minor difficulty. This does, however, not hold true for all individuals: A specially vulnerable group consists of children with complex needs.

1.1.2 Complex needs

Children with complex needs at an early stage of development is a small group in society. Within this group, many have limitation in vision and hearing as well as limitations in motor control. Children with complex needs are also considered for CI surgery, but the intervention required following the implantation may be different from that of typically developing children [3]. It has been suggested that the benefit of CI use for children with complex needs should be seen in a broader perspective than speech outcome [4].

Objective methods can be applied for adjusting and evaluating the HA/CI, but they are problematic in assessing the performance for young children. Subjective methods target both qualitative and quantitative aspects of hearing. Primarily these are speech recognition and pitch perception [5, 6], but also melody perception [7] and music en-

joyment [8]. However, children with hearing impairment and complex needs often show minimal response to auditory stimulation through HA/CIs.

In many cases the responses are unspecific and provide little information on the sound appreciation. In a wider perspective the lack of clear-cut feedback constitutes a serious problem. The capacity to hear is established by auditory stimulation enabling physiological maturation as well as higher level learning. Due to the lack of reactions from the child the caregivers may give up the endeavor to ensure that the HA/CI is used continuously. This may result in suboptimal stimulation and lead to further suppressing of auditory processes.

Studies by Kraus et al [9] have shown that active listening has a positive and lasting effect on perception after surgery, and therefore the child must be systematically exposed to sound in order to develop auditory capacity. Wiley et al [10] concluded that caregivers reported a variety of benefits for children using CI, such as more awareness to environmental sounds, higher degree of attention and clearer communication of needs. Studies also show that children unable to communicate with their environment in a meaningful way may have a disrupted emotional and cognitive development and experience their surroundings as being chaotic [11].

1.1.3 Special needs interfaces for sound manipulation

Within the expanding field of new musical instruments that are based on a software/hardware hybrid solution, work with special needs users has always been a prominent direction. Many of the products and prototypes that have been described (see for instance [12, 13, 14, 15, 16]) address classical music therapy needs and problems, and few look at sound perception specifically. Very little research has been done on the combination of severe hearing impairment and the other complex needs as described above, and in particular with the aim to assess hearing and stimulate active listening [17].

2. SOFTWARE AND HARDWARE

One characteristic of the target user group is that every individual has particularly demanding needs, and there is seldom much in common between the users. Additionally, a set-up that works in one session might not be possible to use at all in the next. It is therefore necessary to be able to radically change the software behavior and hardware configuration within moments. For example, a child that is still and hardly able to move an object in one session may in another session display strong or involuntary movements. The software and hardware must correspond to such diverse states, and the technician must be ready to adapt to the situation quickly to ensure a working system and safe environments for the involved persons.

In particular for the software side, sudden changes in amplitude or sound characteristics must happen in a controlled fashion. The sound should not under any circumstance exceed planned level restrictions. On the hardware side, safety must be ensured with regards to for instance sharp edges, loose parts, and wires that can strangle or unexpectedly

return thrown sensors. As the user testing has shown, the hardware will often be vigorously handled.

To accommodate quick and considerable changes, the software was written in Pure data which is an excellent environment for prototyping (and is open source software). The hardware has for the most part consisted of sensors on detachable units that are easy to place on clothing or objects, connected to an Arduino digital-analog board. This low-cost solution promotes both easy distribution to schools and individuals, and easy extensions of the system.

2.1 Design method

Unlike typical situations where the software/hardware developer engages the user in a participatory design process, or feed back responses from test sessions [18], here the decisions were mainly based on solid prior background knowledge about the users. Most importantly, the aforementioned need for quickly altering any part of the setup had to be attended.

After the first trials, which took place in real settings, relatively little has been reprogrammed or redesigned. In future revisions, we expect that the main modifications concern the software interface (which is not operated by the child) and the sensor hardware (which will need to become more durable).

Models for generating the sound output, described below, can easily be added. This was indeed done during the first sessions and is also foreseen to be a continuous process. New and replacement sensors are also expected to be added when needed. These processes possibly advocate a participatory design methodology.

2.2 Software and sound models

The sound interaction was initially inspired by the way scratch DJs treat records: how they drag the sound fast or slow over a certain spot and isolate small fragments of a sound recording [19] (hence also the name *sound scraper*). Another inspiration from scratching, though more pragmatic, was that the audio signal of this instrument typically has a lot of broadband energy that swiftly sweeps the audible frequency range [20]. This, we argued, could seem effective for listening with cochlear implants when we are not sure if a child can perceive sound in a certain frequency region or not.

In the current implementation, the software interface includes around five sound models for choosing, of around ten different ones (see Figure 1). New models can be added to the interface easily, and they are typically based on existing Pure data abstractions, for instance from Pd's patch example library or previous works by the authors. A few examples are given below.

The Looper model loops a segment of a recorded sound and was derived from the Skipproof application [21] and the Pd example `B12.sampler.transpose`. Loop segments can be varied from the whole file down to a few milliseconds, but the typical loop lengths are at least 2-300 ms. Other parameters are starting point, and playback speed ("pitch").

The Vocoder model was added to be able to 'freeze' and move around in the sound file, and it was based on the patch `I07.phase.vocoder`. Available parameters are playback speed, pitch change ("tuning"), and playback position.

The Theremin was included to allow sweeping both pure tone and harmonic sounds across a broad frequency range to assess pitch perception. To introduce variations in tones when they are kept at a stable pitch level, frequency modulation was added. The main parameters are pitch, harmonics, and frequency modulation speed and range.

Pulse trains of bandpass-filtered white noise bursts offer possibilities to manipulate parameters in a rhythmic sequence of tones. The model was added to explore the temporal resolution which can be problematic with CI. Adjustable parameters include tone attack steepness, tempo, tone duration, noise filter, and filter central frequency.

The music player, based on the `oggreadd~` object, plays compressed audio files. A compressed format was chosen as this model uses whole songs. Only two parameters can be changed: track selection and track position. The music player was not included from start, but when we devised the session program, we decided to include a mode with less interaction.

In addition to the parameters mentioned above, it is possible in all models to control the amplitude, to add echo for making sounds more complex, to add filtering for amplifying or attenuating frequency ranges.

2.3 Hardware and motion sensors

All interaction with the software from the user was projected to be achieved through capturing body movement and gestures with sensors. Such sensor data include inertia measurements, proximity, bending and pressure. In addition to these we have used game controllers. Naturally, even analysing audio or video input would work well (for recognizing speech, facial expressions and body movements see e.g. [22]), but this has not been tested in respect of personal integrity, and also since the included sensors already present a sufficiently rich environment for interaction.

In the last versions, the hardware consisted of up to four sensor "bundles" that could be placed on or near the child. Each bundle was connected to the Arduino board with a 2-3m flat-cable or twinned wire to provide the child with a surrounding space. The bundles were enclosed in protective material and could easily be fastened with tape, rubber bands or velcro. Choosing sensors is still an ongoing process, but typically one of the bundles had an inertia sensor with up to six degrees-of-freedom (accelerometer, gyroscope), one bundle had analog sensors (pressure, light intensity, bending), and one bundle had one or more momentary buttons.

Momentary buttons were used with some sophistication and not as simple triggers. For many of the children, buttons are interfaces which they are familiar with. We extract several control parameters from the pushed buttons, and they are associated with a sort of increasing and decaying energy measure. First, a parameter value has an increase corresponding to the push duration. Second, a parameter

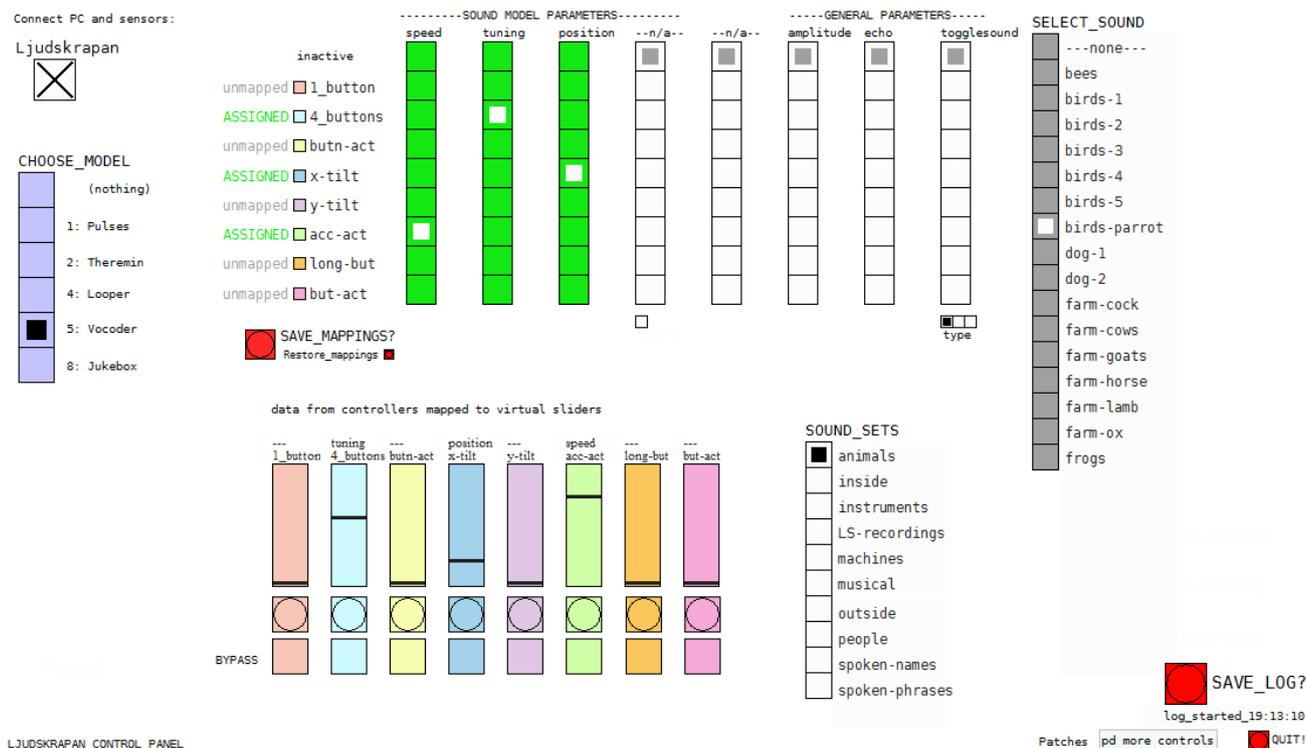


Figure 1. The software interface for the Soundscraper. The person operating the software chooses sound models to the left, mapping of parameters in the middle part, and sound files to the right. Incoming sensor data are mapped to the sliders in the lower part of the screen. The interface shows only the choices available for each sound model and sensor setup.

value has an increase corresponding to the push frequency. These can be used for each separate button or accumulated. Third, a parameter value has an increase related to the number of pushed buttons when there are more than one.

The sensor bundles, including the buttons, were integrated in toys or familiar objects, or they were placed on or around the body. To place sensors on the moving limbs, we used armbands, headbands and similar. For placement on toys or objects, the combination of expected movement, sensor type and fastening possibility had to be explored; two illustrating examples were an enticing rubber bathing duck with a light intensity sensor inside and a steering wheel with an accelerometer inside permitting “steering” in any direction.

Unlike in a typical music interface application, with special needs children we can have a conflict of interest where a movement pattern generates good sensor readings, but there is an incentive to suppress that activity. Likewise, a limited output from a sensor may have to be tolerated, as a more pressing concern is to encourage a certain behavior.

A consequence of having to settle with tracking movement patterns that are very limited, is to be able to scale the input data from the movement range. In the mapping between input sensor data and output sound parameters, we aim at always dealing with full-range signal (0..1). Due to the nature of the movements, it is necessary to rescale the sensor output during use. A problem with the rescaling is that there are not so many typical movements, such as repeating gestures, among the user group. One of the common characteristics is that the individual might be mo-

tionless for a while, then only momentarily move before returning to a calm state.

2.4 Parameter mapping strategies

One of the cornerstones in making the software and hardware flexible and adaptable is to allow for changing the parameter mapping between input and output easily. The importance of parameter mapping has been thoroughly studied [23], and we can use experience from previous projects. In a typical new instrument-situation a skilled musician performs practiced music for an informed audience [24].

Here instead, we face a situation where a child interacts with a system and there might not even be visible signs of any audio perception taking place, and similarly the motor control can often appear to be so erratic that it is not feasible to find proof of any interplay with the produced sounds. In these situations, guidelines for parameter mapping could be followed to ensure possibilities for rich and expressive interaction [14], for instance by defining “activity thresholds”, but the main challenge is to ensure that the control input is used effectively.

3. USER TESTING

For each session with a child, the caregivers and experimenters made careful predictions and plans based on the child’s physical and intellectual condition, personality and known preferences. Still in most cases, the expected or intended interaction was in at least some respects compromised by unforeseen behavior. Causes for the unforeseen

behavior could be several and hard to explain: some factors include unfamiliar environment for the child, sensed expectations from the people in the room, unusual exposure to sound, excitement or anxiety, insecurity towards new persons, or even factors unrelated to the experiment.

We have so far been conducting tests with a small population of children with severe multiple physical and cognitive impairments. The project has ethical approval, but most of the details and data from the tests are not available for inclusion in analyses. Future experiments are scheduled where more data can be included. Presumably, even results from these experiments will be challenging to use in a comprehensive analysis because of the extremely diverse nature of each session. A few of the aspects we will continue to look at are related to quantitative measures. Qualitative measures require a great deal of objective interpretations and observations that in turn necessitate personnel that are closely acquainted with the child (these qualitative measures are naturally essential in the bigger perspective).

The first quantitative measure we look at is time spent during a task which can be adapted to the situation. Typical time-measured tasks are

- preference of a frequency region over others,
- preference of a (musical) sound over others,
- preference of a motor activity over others,
- sound level preference,
- (...)

These measurements require little interaction, but it is necessary to set the conditions right so that data from a sensor do not coincide with a comfortable resting position. From the experiments, we have noted and been informed that the children in general show a considerably higher amount of attention to the task than they normally would do. It is likely needed to gather much data before making conclusions from a time-measure method.

A second quantitative measure requires a more developed motor control and intellectual capacity. By restricting the range where a sensor produces the aspired result (for instance, making an appreciated sound louder) we can evaluate the determination to achieve the wanted sound both from assessing the difficulty of using the sensor over a restricted range as well as the time spent. Extensions of this method include to evaluate if a sound parameter is preferred over others in spite that it is harder to attain it.

3.1 Evaluation

The Soundscraper concept has been tested with the aim to identify strengths and weaknesses in the design and the method. Twenty five children with complex needs were included for the evaluation.

Before the test session, details and information on the children were collected including general level of functioning, motor skills, personal interest and special fears (if any), and auditory functions. For each child a plan was outlined including adaption of sensor equipment and selection

of auditory stimulation (see Figure 2). The children were seen together with a parent, caregiver or teacher. A video recording of each session was made. The testing included reactions to and handling of sensors, answer to sound and sound manipulation, attention span, and mood following the session. All accompanying persons were asked to evaluate the child's reactions.

3.1.1 Reactions to and handling of sensors

The children reacted differently to the sensors, both those attached to the body and to the objects holding sensors. Generally, the impression was that they could be described as having a toy-like function. All children were able to either handle or move the sensor sufficiently for the purpose. One child rejected all types of sensors due to hypersensitivity in all extremities.

Sensors attached to arms or hands were generally less well tolerated than sensors manipulated purposely by the child (i.e. attached to an object or placed within reach). This was also true for children with very limited range of movement. Sensors attached to the head were well tolerated and used for especially demanding situations. When the movements were voluntary, such as for one child who could control turning and tilting of the head, the child immediately grasped the link between movement and sound. However, when the movements were mainly involuntary, it became a challenge to make relevant mappings between movement and sound. Two children with autism and autism-like symptoms devoted all attention to the cords attached to the sensors. One child moved arms and hands freely but stopped moving the extremity when a sensor was placed in the hand or attached to the arm.

3.1.2 Answer to sound and sound manipulation

All but five children showed clear reactions, and generally appreciation, to sound and sound manipulation. This group included the four children mentioned above and a child with profound hearing loss who had HA but not yet received a CI. Three children were intrigued by sound change but did not seem to make the connection between movement and sound.

3.1.3 Attention span

Five children showed limited span of interest. Characteristic of this group was that they all clearly understood the task and made the coupling between their actions and the sounding result. However, they more quickly became bored and started to act without purport and would need a more complex mapping between gestures and the sound output. For the other children, an increased attention span was registered. It should be noted that these observations were based on the children's first encounter both with us and the Soundscraper, and long-term effects on attention span were not investigated.

3.1.4 Evaluation of the children's reaction

Two children showed mild negative reactions during the session. This was the child with hypersensitivity and one child who did not make the connection between movement



Figure 2. The images show how the sensors can be adapted to different situations. In the left picture is a blind girl who disliked holding onto objects, and the sensor was thus placed on the head. In the middle is a girl with good grip and movement in her left hand. The right picture shows a boy who had a plastic rod used in gymnastics class that he enjoyed waving with, and the sensor was placed there. For all three, the sensor is the same inertia unit that measures movement. (Video stills are printed with permission, but moments where the face was covered were chosen deliberately.)

and sound. The negative mood did not persist for any extended period. A few children fell asleep directly following the session, but this was reportedly due to reasons outside the test situation. Fourteen out of twenty five caregivers were surprised by the interest in sound and the persistence showed by the child. One mother said that she did not believe that her son reacted to sound but rather to the sensor as such. However, this impression was not shared by the audiologist and the other caregivers present.

One unexpected observation was that a girl with spastic tetraplegia and involuntary reflex movements was able to relax when listening to a particular sound. During this relaxed state she was able to move one hand voluntarily. This suggests that active manipulation and listening to preferred sounds may strongly influence the overall motor pattern, which will be explored further in forthcoming tests.

4. CONCLUSIONS

The Soundscraper was successfully applied in user tests involving 25 children with hearing aids or cochlear implants, in addition to other physical or cognitive impairments. Both the software and hardware parts could be adjusted to the conditions of each individual session. The results of the preliminary testing were encouraging, and the caregivers indicated that the concept is promising and could possibly be introduced in their school activities.

The concept was judged to be stimulating and rewarding to children with listening capabilities at an early level of auditory development. On the contrary, children who already mastered conscious listening appeared to need higher degree of interaction freedom, especially when they were not able to produce sufficient or controlled movement.

Sensors that require active handling by pushing, pulling or moving an object were more likely to be accepted than sensors that were directly attached to the body of the child. Children with autism-spectrum diagnosis possibly need specially designed sensors to overcome problems not directly related to movement constraints.

The sound models were quite simple, but they provided a complex sound environment even through limited interaction. When the sensor readings were poor, creative mappings and data scaling were successfully used to compensate the scarce input. Having a small but versatile arrange-

ment of sensors that could be placed freely was a great advantage over sensors fixated to objects.

During a session with a boy having a hearing loss combined with blindness, it was noted that he was eager to explore objects with his hands. This behavior was seldom observed for this boy, who for most of the time was reluctant to use his hands. This suggests that the Soundscraper could even be useful for normal-hearing children with blindness and complex needs.

4.1 Future work

User testing will continue with a group of children included in the present study. The Soundscraper will be included in the daily activities at school. Three different functions will be explored: listening for joy and amusement; active listening and training; and using sound for communication and as a means for raising context awareness.

Future analyses of the logged data will hopefully reveal characteristics of sound perception and listening preferences. Such efforts need to be carried out in collaboration with audiologists and caregivers who know the child well. Finally, the software and hardware components need to be developed further to provide an effective and stable environment for the involved caregivers.

Acknowledgments

We would like to sincerely thank the children and parents who participated in this study and also the professionals who took part with great enthusiasm. We are grateful for kind contributions to the development of the Soundscraper from the Promobilia Foundation and the inspiration provided by The Swedish National Agency for Special Needs Education and Schools.

5. REFERENCES

- [1] M. Puckette, "Pure data: Another integrated computer music environment," in *Proc. of the International Computer Music Conference*. San Francisco: International Computer Music Association, 1996, pp. 269–272.
- [2] Arduino, "Arduino open-source electronics prototyping platform," <http://www.arduino.cc/>. [Online]. Available: <http://www.arduino.cc/>

- [3] S. Wiley, M. Jahnke, J. Meinzen-Derr, and D. Choo, "Perceived qualitative benefits of cochlear implants in children with multi-handicaps," *International Journal of Pediatric Otorhinolaryngology*, vol. 69, no. 6, pp. 791–798, 2005.
- [4] T. P. Nikolopoulos, S. M. Archbold, C. C. Wever, and H. Lloyd, "Speech production in deaf implanted children with additional disabilities and comparison with age-equivalent implanted children without such disorders." *Int J Pediatr Otorhinolaryngol*, vol. 72, no. 12, pp. 1823–1828, Dec 2008. [Online]. Available: <http://dx.doi.org/10.1016/j.ijporl.2008.09.003>
- [5] L. L. Pretorius and J. J. Hanekom, "Free field frequency discrimination abilities of cochlear implant users." *Hear Res*, vol. 244, no. 1-2, pp. 77–84, Oct 2008. [Online]. Available: <http://dx.doi.org/10.1016/j.heares.2008.07.005>
- [6] W. D. Nardo, I. Cantore, F. Cianfrone, P. Melillo, A. R. Fetoni, and G. Paludetti, "Differences between electrode-assigned frequencies and cochlear implant recipient pitch perception." *Acta Otolaryngol*, vol. 127, no. 4, pp. 370–377, Apr 2007. [Online]. Available: <http://dx.doi.org/10.1080/00016480601158765>
- [7] R. P. Carlyon, C. J. Long, J. M. Deeks, and C. M. McKay, "Concurrent sound segregation in electric and acoustic hearing." *J Assoc Res Otolaryngol*, vol. 8, no. 1, pp. 119–133, Mar 2007. [Online]. Available: <http://dx.doi.org/10.1007/s10162-006-0068-1>
- [8] K. Veekmans, L. Ressel, J. Mueller, M. Vischer, and S. J. Brockmeier, "Comparison of music perception in bilateral and unilateral cochlear implant users and normal-hearing subjects." *Audiol Neurootol*, vol. 14, no. 5, pp. 315–326, 2009. [Online]. Available: <http://dx.doi.org/10.1159/000212111>
- [9] N. Kraus, E. Skoe, A. Parbery-Clark, and R. Ashley, "Experience-induced malleability in neural encoding of pitch, timbre, and timing." *Ann N Y Acad Sci*, vol. 1169, pp. 543–557, Jul 2009. [Online]. Available: <http://dx.doi.org/10.1111/j.1749-6632.2009.04549.x>
- [10] S. Wiley, J. Meinzen-Derr, and D. Choo, "Additional disabilities and communication mode in a pediatric cochlear implant population," *International Congress Series*, vol. 1273, pp. 273 – 276, 2004, cochlear Implants. Proceedings of the VIII International Cochlear Implant Conference. [Online]. Available: <http://www.sciencedirect.com/science/article/B7581-4DNPJSX-2F/2/19888be6162034dc9a9b9a345af1fe45>
- [11] M. Wass, "Children with cochlear implants. Cognition and reading ability." Ph.D. dissertation, Linköping University, Department of Behavioural Sciences and Learning, 2009.
- [12] B. P. Challis and K. Challis, "Applications for proximity sensors in music and sound performance," in *Computers Helping People with Special Needs*, ser. Lecture Notes in Computer Science. Berlin / Heidelberg: Springer, 2008, vol. 5105, pp. 1220–1227.
- [13] S. Bhat, "Touchtone: an electronic musical instrument for children with hemiplegic cerebral palsy," in *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, ser. TEI '10. New York, NY, USA: ACM, 2010, pp. 305–306.
- [14] A. Hunt, R. Kirk, and M. Neighbour, "Multiple media interfaces for music therapy," *IEEE Multimedia*, vol. 11, no. 3, pp. 50–58, 2004.
- [15] W. L. Magee and K. Burland, "An exploratory study of the use of electronic music technologies in clinical music therapy," *Nordic Journal of Music Therapy*, vol. 17, no. 2, pp. 124–141, 2008.
- [16] A. L. Brooks and E. Petersson, "Play therapy utilizing the Sony EyeToy," in *Proc. of the 8th Annual International Workshop on Presence*, 2005, pp. 303–314.
- [17] C. Dravins, R. van Besouw, K. F. Hansen, and S. Kuške, "Exploring and enjoying non-speech sounds through a cochlear implant: the therapy of music," in *11th International Conference on Cochlear Implants and other Implantable Technologies*, Karolinska Institutet, 2010, p. 356.
- [18] L. Elblaus, K. F. Hansen, and C. Unander-Scharin, "Exploring the design space: Prototyping "The Troat v3" for the Elephant Man opera," in *Proceedings of the 8th Sound and Music Computing Conference*, Padova, Italy, July 2011.
- [19] K. F. Hansen, "The acoustics and performance of DJ scratching. Analysis and modeling." Ph.D. dissertation, Kungl Tekniska Högskolan, 2010.
- [20] K. F. Hansen, M. Fabiani, and R. Bresin, "Analysis of the acoustics and playing strategies of turntable scratching," *Acta Acustica united with Acustica*, vol. 97, pp. 303–314, 2011.
- [21] K. F. Hansen and R. Bresin, "The Skipproof virtual turntable for high-level control of scratching," *Computer Music Journal*, vol. 34, no. 2, pp. 39–50, 2010.
- [22] G. Castellano, L. Kessous, and G. Caridakis, "Emotion recognition through multiple modalities: Face, body gesture, speech," in *Affect and Emotion in Human-Computer Interaction*, ser. Lecture Notes in Computer Science, C. Peter and R. Beale, Eds. Springer Berlin/Heidelberg, 2008, vol. 4868, pp. 92–103.
- [23] E. R. Miranda and M. M. Wanderley, *New digital musical instruments: Control and interaction beyond the keyboard*. A-R Editions, 2006.
- [24] S. Jordà, "Instruments and players: Some thoughts on digital lutherie," *Journal of New Music Research*, vol. 33, no. 3, pp. 321–341, 2004.

C. ELEGANS MEETS DATA SONIFICATION: CAN WE HEAR ITS ELEGANT MOVEMENT?

Hiroko Terasawa¹, Yuta Takahashi¹, Keiko Hirota¹, Takayuki Hamano²,
Takeshi Yamada¹, Akiyoshi Fukamizu¹, Shoji Makino¹

Life Science Center of TARA, University of Tsukuba, Ibaraki, Japan.¹
JST, ERATO, Okanoya Emotional Information Project, Tokyo, Japan.²
terasawa@tara.tsukuba.ac.jp, y-takahashi@tara.tsukuba.ac.jp,
hirota@tara.tsukuba.ac.jp, hamano@japan.com, takeshi@cs.tsukuba.ac.jp,
akif@tara.tsukuba.ac.jp, maki@tara.tsukuba.ac.jp

ABSTRACT

We introduce our video-data sonification of *Caenorhabditis elegans* (*C. elegans*), a small nematode worm that has been extensively used as a model organism in molecular biology. *C. elegans* exhibits various kinds of movements, which may be altered by genetic manipulations. In pursuit of potential applications of data sonification in molecular biology, we converted video data of this worm into sounds, aiming to distinguish the movements by hearing. The video data of *C. elegans* wild type and transgenic types were sonified using a simple motion-detection algorithm and granular synthesis. The movement of the worm in the video was transformed into the sound cluster of very-short sine-tone wavelets. In the evaluation test, the group of ten participants (from both molecular biology and audio engineering) were able to distinguish sonifications of the different worm types with an almost 100% correct response rate. In the post-experiment interview, the participants reported more detailed and accurate comprehension on the timing of the worm's motion in sonification than in video.

1. INTRODUCTION

1.1 Background and Goal

One of the most promising directions in data sonification is the sonification of time-series data, because auditory perception is very sensitive to changes in time [1, 2]. EEG data sonification and seismic data sonification offer successful examples that intuitively display transitions over time, inspiring sonification of other kinds of dynamic, time-series data [3, 4]. In disciplines such as biology, researchers observe organisms by visualization or quantitative measurements, and sonification has seldom been applied as a data-observation technique. However, biological research investigates temporal change of life and organisms, and we expect that sonification may provide a means to comprehend biological phenomena from a new angle.

Copyright: © 2011 Terasawa et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Molecular biologists are currently experiencing major advances in their research methods. The development of high-resolution video recording and the usage of fluorescent protein tags have greatly expanded the possibilities for *in-situ* observation (i.e. non-destructive, real-time observation of living organisms), leading to high expectations for new discoveries by observing dynamic motions. Although such dynamic data are currently observed by “eyeballing” video data, we expect that sonifying temporal elements from video data could be equally advantageous to visual display, offering another modality of data observation in molecular biology.

Since sonification is still a new and unconventional approach for many people outside the auditory display community, we suspect suddenly switching to an abstract auditory display might seem initially implausible to biologists. In order to gain acceptance of sonification as a convincing research method, we need simple and straightforward sonification examples, where the causality between the original data and the resulting sounds can be easily grasped. Therefore, we employed video recording, which is often assumed to be the most concrete visual data, for our preliminary investigation in sonification. Our goal in this study is to examine whether we could “hear the movements we see” in a transparent manner, and thus to motivate further research directions.

1.2 Dynamic Movements of Model Organisms

Caenorhabditis elegans (*C. elegans*) is a small nematode, only about 1 mm long and formed from precisely 959 somatic cells. In the 1960s, Sydney Brenner began using the tiny worm to study the genetics of development, and it has since been used extensively as a model organism [5]. Brenner, John Sulston, and Robert Horvitz shared the 2002 Nobel prize in physiology or medicine for their discoveries in *C. elegans* concerning genetic regulation of development and programmed cell death.

C. elegans has been a popular model organism because of its favorable characteristics for biological experiments. Various genetic and biological techniques have been invented, enabling experiments on development, the nervous system, and aging/longevity.

1.3 Sonification of Image and Motion

“What we see” in video data is an image in motion, and the sonification of a moving image poses its own unique challenge, in contrast to the sonification of a static image or of motion. One common approach in the sonification of a static image is to unfold the image data into a raster sequence of pixel data and to interpret it as a waveform representation of sound [6, 7]. However, this method does not translate the visual sensation of up/down and left/right into audio in an intuitive way, and information about the object position is lost. Meanwhile, sonification of motion is often approached using acceleration sensor data or motion-capture data, at pre-selected measuring points [8, 9]. Substituting sensors and motion capture with video data has also been proposed, such as in human gait sonification, for which Boyd *et al.* extracted a phase configuration that describes the timing pattern of motions in the gait and sonified that data [10].

As that example indicates, video data represent raw image data, and so it is necessary to computationally extract meaningful and intuitive information about visual objects (such as size, speed, position, etc.) and eliminate what is unnecessary background image. Pelletier discussed the matching between visual object and sound object and proposed a framework to sonify the vector representation of corner displacement, which is a perceptually salient feature in vision [11]. This system could employ any kind of synthesis method, but the use of granular synthesis [12], which can represent the addition of large-number simple components, is suggested as one of the natural choices.

1.4 Framework of the Study

In this work, we sonified video data of *C. elegans* wild type and transgenic types. Worm movement in the video was transformed into sound cluster, using a simple motion-detection algorithm and granular synthesis. We examined the resulting sounds with an evaluation test, in which both biologists and audio engineers participated. The effect of sonification was measured with an identification task, in which the participants judge which video the presented sound was generated from. In the next sections, we describe the genetic manipulation, sonification method, and evaluation test, followed by discussion and ideas for further research.

2. GENETIC MANIPULATION OF *C. ELEGANS*

In this study, we investigated three kinds of *C. elegans*, the wild type, the red-fluorescent type, and the rolling type. The latter two types were transgenic strains, and they were generated using standard microinjection methods [13], in which DNA solution was injected into worm gonads.

Wild type: Bristol N2 wild type was provided by the Caenorhabditis Genetics Center [14].

Red-fluorescent type: We prepared transgenic worms expressing red fluorescent protein in their pharynxes (the worm’s throat) by injecting DNA (promoter *myo-2::DsRed* plasmid) into wild-type worms [15].

Rolling type: To generate transgenic worms displaying the rolling movement phenotype, the plasmid containing the *rol-6 (su1006)* gene was injected into wild-type worms [16].

C. elegans was grown on *E. coli* lawn (as food), on agar plates. Video recording was done using a Leica MZFL III microscope and Leica DFC500 digital camera with a resolution of 1168x878 pixels.

Out of these recordings, we prepared four videos (A, B, C, and D) of 20-second duration each. Table 1 shows the list of worm types and video data, and Fig. 1 shows the screenshots of the videos.

| Type of Worm | Video Data |
|------------------------------|------------|
| Wild | A, C |
| Red-fluorescent (transgenic) | B |
| Rolling (transgenic) | D |

Table 1. The List of *C. elegans* Types and Video Data

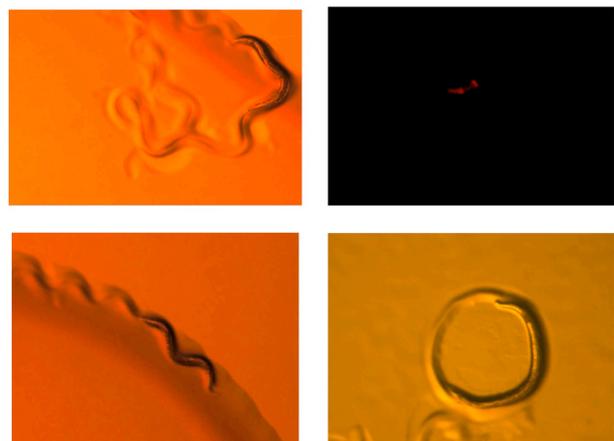


Figure 1. The snapshots of the video data (A: top-left, B: top-right, C: bottom-left, D: bottom-right).

3. SONIFICATION DESIGN AND SOUND SYNTHESIS

3.1 System Overview

The algorithm for sonification was implemented using Max/MSP/Jitter [17] as shown in Fig. 2. We apply a simple motion-detection algorithm on the video data to extract the moving worm from the background, then the image resolution is rescaled to 80 x 60 pixels. The down-sized video shows a rough figure of the worm with a cluster of pixels. Granular synthesis translates the cluster of pixels into a sound cluster.

3.2 Motion Detection

The worm is filmed on an agar plate and exhibits some background objects such as traces of the earlier movements. In order to extract the moving worm, the system reads the video frame every 40 milliseconds. The absolute difference between the current read-out frame and the prior read-out frame is calculated, enhanced by raising the value to the fourth power. Then the extracted motion is smoothed out using envelope-following. The resolution is rescaled to 80 x 60 pixels to minimize the data-flow, and thus the computational power required for granular synthesis.

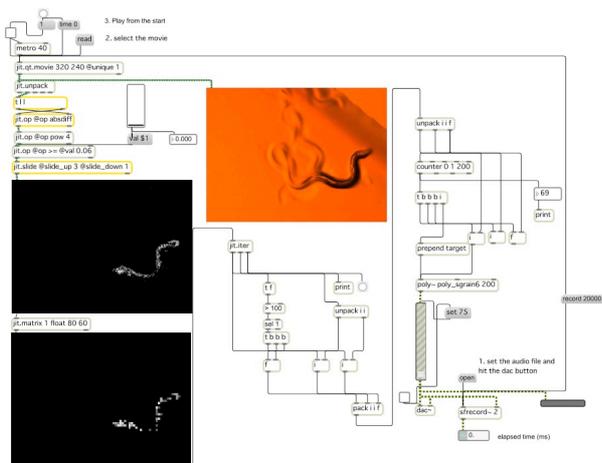


Figure 2. Max/MSP/Jitter implementation of the sonification algorithm.

3.3 Granular Synthesis

The extracted worm image is sonified using the granular synthesis technique, since the granular synthesis is suitable to capturing pixel representation of the image and reflects the complexity of the image into sound. For each pixel of the worm, a small particle of sine wave (wavelet or sound grain) was generated using the parameters of horizontal and vertical positions in the frame (x and y axes) and the intensity of the pixel value. The cluster of

the pixels for the entire worm is heard as the sum of all the corresponding wavelets.

The mapping for the granular synthesis was decided upon by trying several configurations. We designed the vertical axis to correspond with the pitch, the horizontal axis to correspond with the attack-time and duration, and the pixel value to correspond with the intensity of the wavelet. All of these acoustical characteristics are designed to vary exponentially based on the perceptual scaling [18, 19].

4. EVALUATION TEST

4.1 Procedure

We conducted an evaluation test to judge the effect of the sonification using an identification task. We designed the task to resemble the expected use, in which the users are well informed about the sonification concept and the algorithm.

Four sounds (A, B, C, D) were created from the four video-data sources (A, B, C, D, respectively). Then four video clips with matching sounds (i.e., video A and sound A) were produced. These videos are available on our website:

<http://www.tara.tsukuba.ac.jp/~terasawa/Worms/SMC2011.htm>

The participants first received an explanation of the sonification algorithm by watching the Max/MSP/Jitter patch, and then they proceeded to the practice session, in which they watched the four video clips together with sound a few times to familiarize themselves with the sound. In the test session, the participants then listened to the sounds only and were asked to identify the video from which the sound was generated. During the test session, we provided a card showing the snapshots of the video with the names (A, B, C, D), so that the participants did not need to memorize the video names. The participants were allowed to listen to the stimuli several times if they wished.

Each stimulus was presented four times, resulting in 16 stimulus presentations. The stimuli were presented in a randomized order, but ensuring that any particular stimulus would not appear twice in succession. The chance level for making a correct response at each trial is 25 %.

4.2 Participants and Environment

Five molecular biologists (three graduate students and two faculty members) and five audio engineers (two graduate students and three faculty members) participated in the experiment.

The test was conducted in a normal laboratory office space (i.e., not particularly quiet) to resemble realistic user conditions. The stimuli were presented with built-in audio of a laptop computer (Apple MacBook Air) with a closed-type stereo headphone (Sony MDR-7506). The participants adjusted the volume to a comfortable level.

4.3 Questionnaire and Interview

After the test session, we asked four yes/no/maybe questions. The questions were:

- (1) Is it easy to associate the sound and video?
- (2) Can you hear the change of the worm's position?
- (3) Can you hear the rhythm of the worm's movement?
- (4) Do you think you will improve your hearing with more practice?

After the participant answered these questions, the experimenter had a free-form interview with the participant to gather useful comments and suggestions.

5. RESULTS

5.1 Evaluation Test

The mean correct response rate from the identification task is shown in Table 2, which was calculated by averaging the percentage of correct responses across the participants.

All the molecular biologists performed the identification task perfectly. Audio engineers performed slightly less well, but still above 95% correct. Overall, the participants performed the identification task very accurately.

| Group | Mean Correct Response Rate |
|----------------------|----------------------------|
| All | 98.2% |
| Molecular Biologists | 100% |
| Audio Engineers | 96.4% |

Table 2. The Percentage of Correct Responses in the Identification Task

5.2 Questionnaire

The percentage of "yes" answers for the questionnaire is shown in Table 3. The percentage was calculated by taking the sum of responses by counting "yes" as 1, "no" as 0, and "maybe" as 0.5, divided by the number of participants.

With question 1, molecular biologists and audio engineers showed different attitudes with their confidence in hearing. This difference may be because audio engineers are more used to working with sounds, or that there were more musicians among the audio engineers.

With questions 2 and 3, both molecular biologists and audio engineers provided the same type of responses. Most of them recognized the change in position, but they were not confident that they heard the "rhythm" of the movement. We asked question 3 expecting that the participants could identify some patterns of the movement. Perhaps the use of the word "rhythm" was not appropriate because it implies very periodic patterns, while the worm shows only pseudo-periodic patterns with its movement.

With question 4, all of the participants answered that they could improve the hearing by practice. The participants showed almost 100% accuracy in the identification task. Improving the hearing would lead to the easier comprehension of the sounds, if the accuracy is already accomplished.

| Question # | Molecular Biologists | Audio Engineers |
|------------|----------------------|-----------------|
| 1 | 40% | 90% |
| 2 | 100% | 90% |
| 3 | 60% | 60% |
| 4 | 100% | 100% |

Table 3. The "Yes" Answer Percentage of the Questionnaire

6. DISCUSSION

During the interview, many of the participants stated that the main identification cues were the density of sound (i.e., the amount of wavelets) and the pattern of pitch change (i.e., vertical displacement).

The density of sound becomes very low with the red-fluorescent type. Because the number of visible cells in the worm's body is very few, only a small number of wavelets exist in the sound. With the other types, the density of sound is low when the worm is stopping or showing only tiny movements. The participants used the timing of sparse sounding and silence as a cue to identify the movement patterns.

The biologists also reported that they were able to focus on the worm movements more accurately with sound than with video. There are some moments when the worms briefly stop their motion. With sounds, such moments are easily detected. But with video, the participants tend not to notice such moments, and have the perception that the worms are moving smoothly without any interruption.

Both engineers and biologists reported they used the density and pitch cues concurrently. However, the parameters that corresponded to the horizontal position (attack time and duration) did not seem to affect the perceived quality of sound. With the use of sound cluster, the wavelets overlap each other, and such overlaps may preclude accurate perception of attack time and duration.

7. CONCLUSION AND FUTURE WORK

In this study, we investigated the potential of data sonification in molecular biology. The movements of wild-type and transgenic *C. elegans* were sonified using motion detection and granular synthesis, so that the sound cluster of wavelets represents the visual cluster of pixels in a perceptually matching manner. The evaluation test showed that both molecular scientists (*C. elegans* specialists) and audio engineers (non-specialists) could accurately comprehend the motion of worms through hearing, demonstrating that data sonification may have a strong potential for applications in molecular biology.

From this collaboration of biologists and audio engineers, various ideas for future directions are emerging. Technical ideas for improving sonification include the use of spatial audio, different kinds of mapping in the synthesis, and using acceleration as a parameter instead of position. However, beyond the technical ideas, we came to realize the potential of sonification in discovering new knowledge in molecular biology. Investigating the rhythmic aspects in the dynamic motion of worms would be of interest, such as the pumping gesture observed at the throat of a worm. Sonification may also be useful for observing the behaviors of model organisms.

In this project, we sonified the already-visible aspects and discovered that sounds can convey some information that we tend to dismiss with vision. However, the sonification of non-visual aspects in biology is a further promising direction. "Listening to the phenomena we cannot see at all" may lead to the most fascinating new discoveries, and seeking such model examples is a desirable next-stage goal.

Acknowledgments

We would like to thank Peter Wang for his generous support in preparing this manuscript. This work was supported by the Kawai Foundation for Sound Technology and Music, Japan.

8. REFERENCES

- [1] G. Kramer *et al.*, "The Sonification Report: Status of the Field and Research Agenda," Prepared for the National Science Foundation by members of the International Community for Auditory Display Editorial Committee and Co-Authors, 1999.
- [2] S. Barrass and G. Kramer, "Using sonification," in *Multimedia Systems* No. 7, 1999, pp. 23-31.
- [3] T. Hermann *et al.*, "Vocal Sonification of Pathologic EEG Features," in *Proceedings of the 12th International Conference on Auditory Display*, London, 2006, pp. 158-163.
- [4] F. Dombos, "Auditory Seismology on Free Oscillations, Focal Mechanisms, Explosions and Synthetic Seismograms" in *Proceedings of the 2002 International Conference on Auditory Display*, Kyoto, 2002, pp. 1-4.
- [5] S. Brenner, "The genetics of *Caenorhabditis elegans*," in *Genetics* No. 77, 1974, 71-94.
- [6] W. S. Yeo and J. Berger, "Raster Scanning: A New Approach to Image Sonification," in *Proceedings of the International Computer Music Conference*, 2006.
- [7] K. Jo and N. Nagano, "MonaLisa: See the Sound, Hear the Image" in *Proceedings of International Conference on New Interfaces for Musical Expression*, 2008.
- [8] S. Barrass, N. Schaffert, and T. Barrass, "Probing Preferences between Six Designs of Interactive Sonifications for Recreational Sports, Health and Fitness," in *Proceedings of ISON 2010, 3rd Interactive Sonification Workshop*, Stockholm, 2010.
- [9] J. M. Pelletier, "Sonified Motion Flow Fields as a Means of Musical Expression" in *Proceedings of International Conference on New Interfaces for Musical Expression*, 2008.
- [10] J. E. Boyd and A. Sadikali, "Rhythmic Gait Signatures from Video without Motion Capture," in *Proceedings of the 16th International Conference on Auditory Display*, Washington, D.C., 2010, pp 187-191.
- [11] J. M. Pelletier, "Perceptually Motivated Sonification of Moving Images," in *Proceedings of the International Computer Music Conference*, 2009.
- [12] C. Roads, *Microsound*, MIT press, 2002.
- [13] C. C. Mello, *et al.*, "Efficient gene transfer in *C. elegans*: extrachromosomal maintenance and integration of transforming sequences," in *EMBO J.* No. 10, 1991, pp. 3959-3970.
- [14] *Caenorhabditis Genetics Center*, University of Minnesota. URL: <http://www.cbs.umn.edu/CGC/>
- [15] P. G. Okkema, *et al.*, "Sequence requirements for myosin gene expression and regulation in *Caenorhabditis elegans*," in *Genetics* No. 135, 1993, pp. 385-404.

- [16] C. A. Peixoto, *et al.*, “Ultrastructural analyses of the *Caenorhabditis elegans* rol-6 (su1006) mutant, which produces abnormal cuticle collagen,” in *J. Parasitol* No. 84, 1998, pp. 45-49.
- [17] Cycling 74, Max 5. URL: <http://cycling74.com/>
- [18] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, Springer, 1999.
- [19] S. McAdams *et al.*, “Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes,” in *Psychological Research* 58, 1995, pp. 177–192.

USING PHYSICAL MODELS IS *NECESSARY* TO GUARANTEE STABLE ANALOG HAPTIC FEEDBACK FOR ANY USER AND HAPTIC DEVICE

Edgar Berdahl

Association pour la Création
et la Recherche sur les Outils
d'Expression (ACROE) and the
Center for Computer Research
in Music and Acoustics (CCRMA)
Edgar.Berdahl@imag.fr

Jean-Loup Florens

Association pour la Création
et la Recherche sur les Outils
d'Expression (ACROE)
and ICA Laboratory
Grenoble Institute of Techn., France
Jean-Loup.Florens@imag.fr

Claude Cadoz

Association pour la Création
et la Recherche sur les Outils
d'Expression (ACROE)
and ICA Laboratory
Grenoble Institute of Tech., France
Claude.Cadoz@imag.fr

ABSTRACT

It might be easy to imagine that physical models only represent a small portion of the universe of appropriate force feedback controllers for haptic new media; however, we argue the contrary in this work, in which we apply creative physical model design to re-examine the science of feedback stability.

For example, in an idealized analog haptic feedback control system, if the feedback corresponds to a passive physical model, then the haptic control system is guaranteed to be stable, as we show. Furthermore, we argue that it is in fact *necessary* that the feedback corresponds to a passive physical model. Otherwise, there exists a passive user-haptic device transfer function that can drive the feedback control system unstable. To simplify the mathematics, we make several assumptions, which we discuss throughout the paper and reexamine in an appendix.

The work implies that besides all of the known advantages of physical models, we can argue that we should employ only them for designing haptic force feedback. For example, even though granular synthesis has traditionally been implemented using signal modeling methods, we argue that physical modeling should still be employed when controlling granular synthesis with a haptic force-feedback device.

1. INTRODUCTION

1.1 Physical Modeling

In the field of sound and music computing, there is already a strong history of physical modeling. The most basic physical modeling approach is to study the physics of a musical instrument, and then to simulate the physical equations in a computer to synthesize sound [1, 2, 3]. However, besides merely imitating pre-existing musical instruments, new virtual instruments can be designed with a computer by simulating the acoustics of hypothetical situations [4], creating a “metaphorisation of real instruments.” Of partic-

ular importance is also that sounds generated using physical models tend to be physically plausible, enhancing the listener’s percept due to familiarity [5, 6].

Physical models can also be employed for real-time interaction. Here the perceptual advantages can be augmented by the apparent physical reality of the simulation. For example, when interacting with a virtual acoustical object, if the user changes the interaction point and the sound changes appropriately, the immersiveness of the user’s experience is enhanced, as well as the quality of the generated sound. This property is not immediately offered by techniques such as sampling, unless the musical instrument’s sound is sampled at all the possible interaction points and typically also at many different excitation velocities, which can require recording and storing large amounts of data.

By employing appropriate environments for generating large-scale physical models, composers can even create entire pieces using the physical modeling paradigm. For instance, initial conditions for mass trajectories can control the evolution of a piece, or complex inner simulated dynamics can also control timbres, notes, phrases, and even whole movements [7].

1.2 Haptic Force-Feedback Interaction According to the Ergotic Function

On a philosophical level, Claude Cadoz already defined three functions according to which a user can interact with an environment (physical or virtual). The first function is the *epistemic* function, which pertains primarily to knowing, for which a user can use the eyes, ears, or kinaesthetic and tactile touch receptors. The second function is the *semiotic* function, which users employ for transmitting symbolic information by way of the voice and body language.

In contrast, when a user exchanges significant mechanical energy with the environment by way of gesticulating, he or she uses the third, *ergotic* function for interaction [8]. For instance, employing a tool to deform an object or move it is ergotic. Bowing a string or playing a drum is also ergotic. In ergotic interaction, the user not only informs and transforms the world, but the world also informs and transforms the user. This is in some sense a consequence of Newton’s third law: for every force, there is an equal and opposite reaction force.



Figure 1. Hand holding haptic device.

The ergotic function can be substituted by neither the epistemic nor semiotic function. In total when the ergotic, semiotic, and epistemic functions are simulated, it is possible for a user to ultimately engage in *instrumental interaction* using a haptic device controlled by a physical model [9]. In the remainder of the paper, we will argue that we should employ physical models for programming haptic force-feedback controllers, effectively implying also that the ergotic function and instrumental interaction should be simulated using physical models. We begin with a more formal discussion of feedback control.

2. FEEDBACK CONTROL

Feedback systems with active elements have the potential to become unstable. For example, acoustic feedback from a loudspeaker into the microphone of a public address (PA) system can cause the PA system to become unstable. Typically howling sets in, where one or a few sinusoids steadily increase in volume until the PA system cannot become any louder or is reconfigured. Howling instability is unpleasant for listeners and should be avoided.

Force feedback haptic devices, such as the one shown in Figure 1, can similarly become unstable. A circuit calculates a feedback force as a function of the orientation of the device (and its history), and the feedback force is exerted on the device. If the haptic feedback control system becomes unstable, then the haptic device can begin to move about erratically [10]. The haptic device can be damaged, other objects in the vicinity of the device can be damaged, and if the device is particularly strong, the user could even be injured.

While instability can be interesting for art, it is clearly important to be able to design haptic feedback systems that are guaranteed to be stable. For simplicity of the mathematics and analysis, we initially assume

- zero feedback control delay,
- infinite control bandwidth,
- time-invariance,
- linearity, and
- zero initial conditions

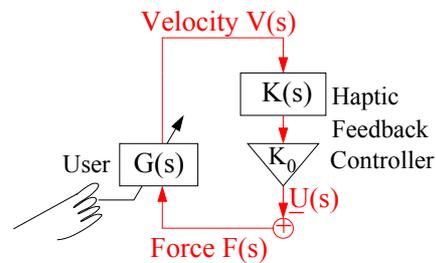


Figure 2. User $G(s)$ connected with haptic controller $K_0K(s)$ in feedback.

to arrive at a practical result, even under real conditions such as digital haptic feedback control (see Appendices A.1-A.3 and B for more discussion of the assumptions). Consequently, the analog feedback control system can be represented in the Laplace s -domain. For a time-domain function $g(t)$, the right-sided Laplace transform is

$$G(s) = \mathcal{L}\{g(t)\} = \int_{0^-}^{\infty} g(t)e^{-st} dt. \quad (1)$$

A block diagram for the feedback control system is shown in Figure 2. $V(s)$ represents the velocity of the user's hand coupled to the haptic device,¹ and $F(s)$ represents the force exerted on the haptic device. In the absence of additional external forces, $F(s) = -U(s)$, where the minus sign is used to emphasize that the system is designed to operate using negative feedback. $K(s)$ represents the haptic feedback control filter, where the scalar loop gain K_0 has been separated out (see Figure 2). Because we employ a negative feedback configuration, we take the gain

$$K_0 \geq 0. \quad (2)$$

3. USER-DEVICE TRANSFER FUNCTION $G(s)$

Since the user's hand and the haptic device are physical devices, *they can always be represented using a physical model* $G(s) = V(s)/F(s)$. Furthermore, since we are considering the linear, time-invariant case, the user's hand is stationary and holding the end of the haptic device with a constant grip. There is some friction at all frequencies, meaning that we assume the hand coupled to the haptic device can only dissipate energy, never create it.² Consequently, in the absence of feedback control, the force $F(s)$ and velocity $V(s)$ of the haptic device will never be far enough out of phase with one another to create energy. Mathematically, this implies that

$$|\angle G(s)| \Big|_{s=j2\pi f} < 90^\circ \quad (3)$$

for all frequencies f in Hz [12]. (Mathematically speaking, $G(s)$ is *strictly positive real*. For more information, consult Appendix D.)

¹ For convenience, we employ velocities rather than positions. With the zero initial conditions assumption, one can easily convert between velocity and position by integrating and in the other direction by differentiating.

² Technically speaking, it could be possible for a user to intentionally destabilize some passive physical models by continually adding energy at low frequencies; however, users seem usually to be sensible enough not to do this [11].

4. SUFFICIENCY OF PHYSICAL MODELS FOR HAPTIC FEEDBACK CONTROL

For the moment, assume that the controller $K(s)$ is determined using a passive physical model. That is, the model consists of passive elements and no energy sources. For example, the model might consist only of masses, springs, and viscous dampers (or equivalently capacitors, inductors, and resistors) all with non-negative coefficients. Then by an analogous argument to the one in the prior section, we have that

$$|\angle K(s)|_{s=j2\pi f} \leq 90^\circ \quad (4)$$

for all f since $K(s)$ is *positive real* (see Appendix D) [12]. Note that we allow frequencies at which there is zero damping—this could for example happen if there were no dampers/resistors and would result in an angle of 90° or -90° .

Next, we show that the net control system is guaranteed to be stable; however, to do so, we need to first introduce a criterion for determining the stability of the control system.

4.1 Revised Bode Stability Criterion

Since neither $K(s)$ nor $G(s)$ has any unstable poles, we can employ the Revised Bode Stability Criterion to state that the feedback system is stable if no “candidate unstable” frequency f_u exists for which:

$$|K_0 K(s)G(s)|_{s=j2\pi f_u} \geq 1 \quad (5)$$

and

$$\angle(K_0 K(s)G(s))_{s=j2\pi f_u} = -180^\circ - n(360^\circ), \quad (6)$$

for any integer n [13]. In other words, the system is stable if there is no frequency at which a sinusoid traveling all the way around the loop could interfere perfectly constructively with itself (due to (6)) with the magnitude of that sinusoid increasing with every loop (due to (5)).

4.2 Proof Of Stability

Since (2), (3) and (4) hold, there is no frequency f for which (6) can hold. Thus, we have showed that the haptic feedback control system must be stable if controlled by a physical model, for any passive user-device transfer function $G(s)$. Roughly speaking, the stability is independent of the choice of haptic device and what the user is doing.

4.3 Unconditional Stability

Note that the stability is also independent of the magnitude of K_0 . In other words, the control gain can be arbitrarily large! This remarkable property is known as *unconditional stability* [14, 15]. It indicates that under our ideal assumptions, the values of the physical model do not matter—the haptic control system is guaranteed always stable.

5. EXAMPLE

We now illustrate the proof of stability using a concrete example of a user touching a virtual resonator.

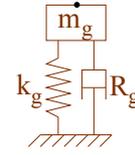


Figure 3. Simplified physical model of user-device.

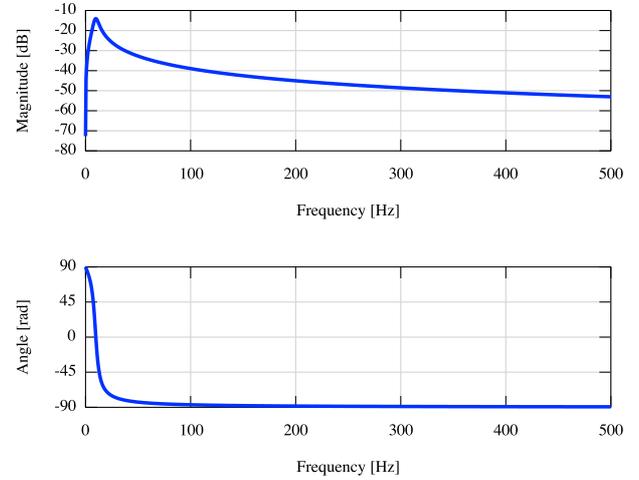


Figure 4. Magnitude and phase of user-device $G(s)|_{s=j\omega}$.

5.1 User-Device $G(s)$

Although the user can exert forces at will on the resonator, we consider these forces as inputs to the control system and not as part of the feedback loop, so we do not need to model them when examining the stability of the feedback loop. Hence, we provide a simple physical model of a user coupled to a haptic device in Figure 3. The mass m_g represents the mass of the user’s hand coupled to the haptic device. The user’s hand in conjunction with the haptic device presents a stiffness of k_g and viscous damping R_g . Much more complex models could be employed at this point, but it is not necessary for the illustrative purposes of this paper [16]. We can use the model to find $G(s)$:

$$G(s) = \frac{V(s)}{F(s)} = \frac{s}{m_g s^2 + R_g s + k_g}. \quad (7)$$

The model values could vary significantly, so for example we employ approximate parameters obtained by averaging results from a subject test, in which subjects gripped a haptic device with a grip force of about 9N [11]. In other words, we chose $m_g = 143$ g, $R_g = 5$ N/(m/s), and $k_g = 0.538$ N/mm. The phase response of the user coupled to the haptic device is shown in Figure 4. As required by (3), the phase response lies within the range $(-90^\circ \ 90^\circ)$ as illustrated in Figure 4, bottom.

5.2 Controller

The physical model for the controller is shown in Figure 5. The musical resonator has mass $m_v = 4$ g and damping coefficient $R_v = 0.01$ N/(m/s), setting the exponential decay time constant to 0.8 sec. To make the resonance frequency approximately 300Hz, we choose stiffness $k_v = 14.2$ N/mm. To limit the force that the haptic

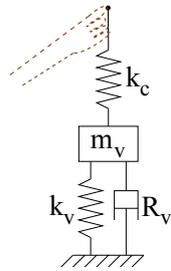


Figure 5. Physical model employed to derive controller $K_0K(s)$.

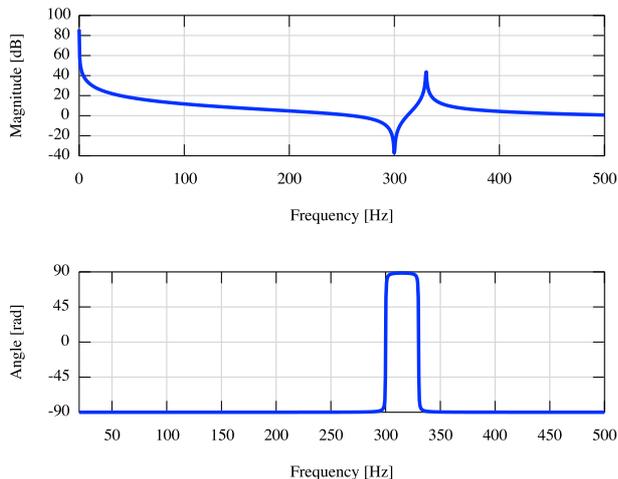


Figure 6. Magnitude and phase of controller $K_0K(s)|_{s=j\omega}$.

device must display, an additional spring is incorporated into the model $k_c = 3$ N/mm.

Solving the equations of motion and converting to the Laplace domain, we arrive at the following, from which it can be seen that k_c plays a role similar to the loop gain:

$$K_0K(s) = \frac{U(s)}{V(s)} = \frac{k_c}{s} \cdot \frac{m_v s^2 + R_v s + k_v}{m_v s^2 + R_v s + (k_v + k_c)}. \quad (8)$$

The magnitude response of the controller is shown in Figure 6 (top), which shows that the resonance frequency is increased slightly to 330Hz due to the presence of k_c . There is also an anti-resonance frequency still at approximately 300Hz. However, because the controller represents a physical model, its phase response still lies within the range $[-90^\circ, 90^\circ]$ as specified by (4), even though it comes close to its allowable boundaries in Figure 6 (bottom).

Hence, as proved in Section 4.2, no “candidate unstable frequency” f_u exists satisfying (5) and (6), so the control system is guaranteed stable. This will hold for *any* passive physical model employed to specify the controller $K_0K(s)$, which is convenient for musical practice.

6. NECESSITY OF PHYSICAL MODELS FOR HAPTIC FEEDBACK CONTROL

Among the sound and music computing community, it appears not to be known that in the following sense, it is in fact *necessary* for the haptic controller $K(s)$ to correspond to a passive physical model. The reason for this is that the

set of passive, linear physical models with collocated input and output is the same as the set of passive, linear transfer functions $K(s)$ (see [12] and Appendix D). Indeed both of these sets share the same phase relationship described by (4).

Otherwise if $K(s)$ does not correspond to a passive linear physical model, there exists a passive user-device $G(s)$ for which the feedback system can be driven unstable—in other words, a passive user and haptic device could be found for which the haptic control system would be unstable.

The proof of necessity is too long to be included in this conference paper without eliminating the examples [17]. Nevertheless in summary, it is a proof by construction that is analogous to choosing $G(s)$ such that (6) holds at some frequency f_u , and then increasing the scalar gain of $G(s)$ until (5) also holds.³

7. MAIN RESULT

Thus we have arrived at what we consider to be a rather remarkable result:

if stable feedback control of a haptic device is desired for applications in new media, then we argue that designers should not start by simply employing any arbitrary feedback, rather they should design the feedback using physical models.

This has further implications particularly in sound and music computing. When employing a haptic device to control traditionally non-physical modeling sound synthesis engines, a physical modeling approach should nonetheless be employed. This is one reason why ACROE designed the CORDIS-ANIMA physical modeling language that incorporates passive physical modeling elements such as the mass, spring, friction, conditional link, etc. for simulating the ergotic function and enabling instrumental interaction [18].

For instance, if one were to implement haptic feedback control of granular synthesis [3], a good approach would be to model the grains as small masses flowing along a river. An independent external force would cause each mass to vibrate according to its own audio grain signal. Then the user could *dip* into the river using the haptic device via a conditional link, and the output audio signal would be generated by measuring the force exerted upon the haptic device. We would recommend this approach not only because of our positive experiences with physical models, but also because of the arguments in this paper.

³ The proof involves counting the number of possible clockwise loops around the -1 point of the Nyquist plot of $K_0K(s)G(s)$. One key realization is that any counterclockwise loop would imply that either $K(s)$ or $G(s)$ were open-loop unstable, which would violate the assumptions, so there cannot be any counterclockwise loops. It is also necessary to observe that 1) a transfer function can never have a negative number of poles or zeros and 2) offsetting a transfer function by a complex constant does not change its poles.

8. CONCLUSION

Physical modeling for new media can indeed be a creative activity, and now having re-discovered this approach through a scientific stability analysis, we hope to have provided new insight for design of new media. We have attempted to present some results from the mechanical engineering literature [17] in a way that is accessible to the sound and music computing community. In re-presenting this work, we have gathered new perspective on the stability of feedback control systems and re-affirmed our enthusiasm for physical models.

Acknowledgments

We are grateful to the reviewers for their insightful comments, which we have attempted to integrate to make the paper as accessible as possible. Reviewers and readers interested in designing digital feedback controllers for plants with a specified delay may wish to consult work by Florens et al [19].

Finally, Edgar Berdahl would like to thank J. Edward Colgate for personal correspondence and Günter Niemeyer for informing him about the prior work [17].

A. NOTES ON ASSUMPTIONS

We now argue that the general results are still practically valid in light of the assumptions we made in Section 2.

A.1 User Is Time-Varying

In practice, $G(s)$ changes with time according to what the user is doing. For example, while completing certain tasks, the user changes the mobility of his or her hand [20, 21, 11]. However, this is in fact the point of the present paper—in order to ensure stable performance for *any* arbitrary user-device mobility $G(s)$, it is necessary that $K(s)$ correspond to a passive physical model. Hence, we argue that $K(s)$ may as well be designed using a passive physical model to guarantee stable performance [18].

A.2 Non-Ideal Feedback Characteristics

In practice, all feedback control systems have limited bandwidth. In addition, digital control systems exhibit additional delay in the control loop. Consequently, real controllers cannot perform significant feedback control at especially high frequencies. However, in practice, one observes that the simple theory in this paper predicts relevant aspects of practical performance, as long as one inserts a viscoelastic (optionally nonlinear) element in between the haptic device control point and the physical model, such as k_c (see Figure 5). This reduces the magnitude of the feedback control at high frequencies where practical digital control delay can be particularly problematic (see Appendix B) [10, 16].

It could also be argued that a digitized version of an ideally passive physical model may no longer be formally passive. However, we argue that the physical model should simply be discretized appropriately so that it causes an input-output delay of precisely one digital time unit. In

practice then, this time unit is aligned with the converter sampling and results in no additional, unnecessary delay that could further affect the passivity [22, 23].

We relate the discussion here also to teleoperation, in which a “master” haptic device and a “slave” haptic device are linked together using force-feedback. Signal transmission delay between two separate locations can cause even more significant delay than digital sampling. In this case, the controller cannot form a good model of a simple damper and spring to link the devices together. However, this delay can be cleverly absorbed into the controller model by incorporating a vibrating string (or equivalently an electrical transmission line) into the controller model [24]. Again, we discover a solution based on physical models!

A.3 User Is Nonlinear

In practice, a real user is nonlinear. However, the user is also dissipative. If the haptic feedback controller is passive and corresponds to a physical model, then even if the physical model is nonlinear, by the conservation of energy, the energy in the feedback control system must dissipate over time in the absence of external excitation. Hence, although the *necessity* of the nonlinear case is apparently unproven,⁴ conservation of energy proves *sufficiency*, and it is also very practical to design nonlinear haptic feedback controllers using nonlinear physical models [18].

A.4 Unstable Performance Could Be Desirable In Some Situations

In some situations, artists may desire to create control systems that are unstable. In fact, the E-Bow and Sustainiac are successful commercial products that drive vibrating guitar strings unstable in a controllable manner [25, 26]. Similarly, bowed strings, many wind instruments, and some drum roll techniques incorporate self-oscillations that have become accepted as sounding musical. Even the “unstable” Haptic Drum enables a performer to play arbitrarily complex drum rolls or drum rolls at superhumanly fast speeds [27].

Hence, there are some nonlinear situations where physical models will not be necessary for implementation, but they nevertheless seem to be sufficient given the very wide range of physical phenomena that could be modeled. Unstable behavior can be created using external energy source elements in physical modeling or negative dampers, or similar effects can be obtained by setting initial energetic conditions for objects. Indeed creativity causes us to rethink the science of physical modeling, so possible future work could someday involve studying necessity and sufficiency proofs for employing (nonpassive) physical models to implement unstable haptic musical simulations.

A.5 Transparency of Haptic Rendering

In this paper, we have considered only the stability of the control system according to classical passivity theory [17]. However, we have not considered how *transparently* the

⁴ Personal correspondence with Ed Colgate on Jan. 17, 2011

physical model is presented to the user through the haptic feedback control system. In the classical representation, improving transparency (i.e. accuracy) requires increasing control gains, which can hamper the stability of digital control systems (see Section B). For this reason, Florens et al. have introduced a new method for deriving haptic feedback control systems, which consider the stability and transparency concurrently [19]. The result is a whole new paradigm for deriving haptic feedback controllers. The method involves modeling the coupling of the user-device to the physical model (called a *temporary hybrid system* or *THS*) and adjusting model parameters to achieve optimum dynamics [19]. We are actively carrying out further research in this domain.

B. DIGITAL DELAY

In practice, it is usually more practical to employ digital feedback instead of analog feedback, especially because computers are now so widely available and inexpensive. However, digital feedback control always causes delay in the control loop, which is due to

- analog-to-digital conversion (ADC),
- computation time,
- possible additional delay due to operating system, interrupt, and bus mechanisms on the computer, and
- digital-to-analog conversion (DAC).

Although the delay is always longer than half of one sampling interval T , this is nevertheless a convenient approximation, assuming a conventional implementation of the control loop elements [28, 23].⁵ Hence, (8) becomes the following:

$$K_0K(s) = \frac{k_c}{s} \cdot e^{-sT/2} \cdot \frac{m_v s^2 + R_v s + k_v}{m_v s^2 + R_v s + (k_v + k_c)}. \quad (9)$$

The sampling interval T has no effect on the magnitude of (9) (see Figure 7, top). However, the phase response for $T = 1$ ms is shown in the thinner line of Figure 7 (bottom). There is a linear trend to further and further negative phases, which causes the phase response to dip beneath the allowable limit -90° . The phase response for $T = 0.1$ ms is shown in the thicker line of Figure 7 (bottom). It remains much closer to the limit, but for sufficiently high frequencies falls outside of the allowable range, preventing the digital controller from perfectly calculating feedback equivalent to a delayless, analog physical model.

Nevertheless, it turns out that both for $T = 1$ ms and $T = 0.1$ ms, the control system is still stable for these parameters. However, given the digital control delay, it is possible to drive the system unstable by increasing k_c , which is analogous to eliminating the “stabilizing” compliant spring k_c and attempting to bind the haptic device

⁵ Converters implemented using sigma-delta modulation are usually not fast enough, so for low-latency feedback control successive approximation ADCs and resistor-ladder DACs are often used.

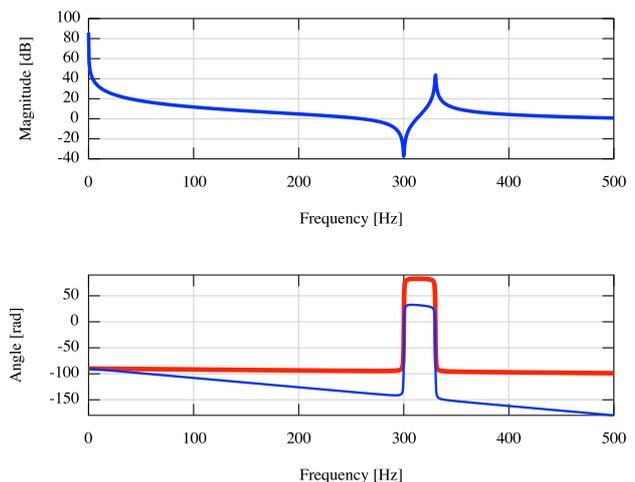


Figure 7. Magnitude and phase of digital controller $K_0K(s)|_{s=j\omega}$.

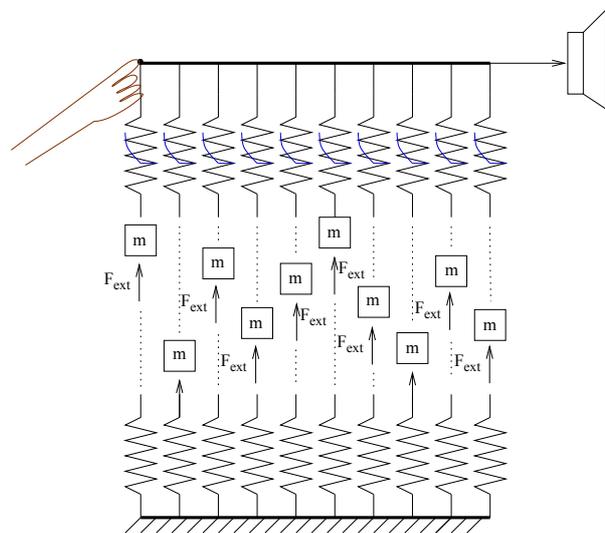


Figure 8. Physical analog model for granular synthesis using ten trapped masses.

directly to the resonator itself. However, this would be contrary to our approach of promoting stability by inserting the compliant element k_c at the interaction point. Indeed, for a more precise argument but specific to a less interesting scenario for sound and music computing, the reader should read the work by Diolati et al. [16]. Typically the shorter T can be made, the further k_c can be increased.

C. REAL-TIME USER INTERACTION WITH MORE COMPLEX EXAMPLE

To demonstrate the viability of the technology, even for digital feedback control, we briefly present an example with real-time user interaction. A physical model for a kind of granular synthesis is shown in Figure 8, in which ten “grain” masses fly back and forth vertically between a mechanical ground (below) lined with linear contact springs and squared-nonlinearity contact springs along a rigid bar (top) coupled to the user’s hand by the haptic device. Despite the specificity of this model, many other physical model scenarios could be employed to implement granular-

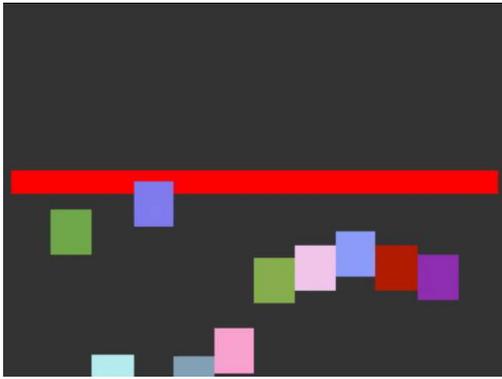


Figure 9. Snapshot from visual output of demonstration.

type synthesis. In the present example, an external force F_{ext} acts on each mass, causing it to vibrate according to input sounds. The force exerted on the haptic device is measured, highpass filtered (not shown), and passed to the audio output, as represented by the loudspeaker schematic symbol in Figure 8, top right.

For such complicated models, visual feedback is also helpful. Hence, we incorporated visual feedback in a demonstration video.⁶ A snapshot of the video is shown in Figure 9. Rather than adjusting the amplitude, frequency, density, grain length, and similar signal parameters, as is typical in granular synthesis [3], these sonic characteristics are adjustable by physical means such as the position of the haptic device, force applied to the haptic device, the stiffness and damping of the hand, etc.

D. POSITIVE REAL FUNCTIONS

For the mathematically minded, we present some information on positive real functions for describing mathematically passive systems. It underscores their equivalence to passive physical models.

Positive real functions were introduced in 1931 for synthesizing transfer functions corresponding to electrical analog circuits [12]. Since then, a rational function $\tilde{K}(s)$ has usually been defined to be *positive real* if and only if

- $\tilde{K}(s)$ is real when s is real, and
- $Re\{\tilde{K}(s)\} \geq 0$ for all s such that $Re\{s\} \geq 0$.

and similarly, a rational function $\tilde{G}(s)$ has usually been defined to be *strictly positive real* if

- $\tilde{G}(s + \epsilon)$ is positive real for all real $\epsilon > 0$ [29].

However, for our purposes it is much more convenient to use the following equivalent definitions in terms of the angle along the frequency axis. We define the rational function $\tilde{K}(s)$ to be *positive real* if and only if

- $|\angle \tilde{K}(j2\pi f)| \leq 90^\circ$ for all frequencies f ,

and similarly the rational function $\tilde{G}(s)$ is *strictly positive real* if and only if

- $|\angle \tilde{G}(j2\pi f)| < 90^\circ$ for all frequencies f [29].

⁶ <http://ccrma.stanford.edu/~eberdahl/CompMusic/GCG.m4v>

Some further properties of positive real and strictly positive real functions are [30]:

1. $1/\tilde{K}(s)$ is positive real.
2. $1/\tilde{G}(s)$ is strictly positive real.
3. If $\tilde{K}(s)$ represents either the driving point impedance or driving point mobility of a system, meaning that the sensor and actuator must be collocated, then the system is *passive* as seen from the driving point. **In other words, if $\tilde{K}(s)$ is positive real, then it corresponds to a passive, linear physical model, and vice versa.**
4. If $\tilde{G}(s)$ represents either the driving point impedance or driving point mobility of a system, meaning that the sensor and actuator must be collocated, then the system is *dissipative* as seen from the driving point. **In other words, if $\tilde{G}(s)$ is strictly positive real, then it corresponds to a dissipative, linear physical model, and vice versa.**
5. $\tilde{K}(s)$ and $\tilde{G}(s)$ are stable.
6. $\tilde{K}(s)$ and $\tilde{G}(s)$ are minimum phase.
7. The relative degrees of $\tilde{K}(s)$ and $\tilde{G}(s)$ must be less than 2.
8. No matter what causal time-domain function $f(t)$ is used to excite the driving point, the velocity response $v(t)$ will be such that $\int_0^\infty f(t)v(t)dt \geq 0$.

E. REFERENCES

- [1] C. Cadoz, A. Luciani, and J.-L. Florens, “Synthèse musicale par simulation des mécanismes instrumentaux,” *Revue d’acoustique*, vol. 59, pp. 279–292, 1981.
- [2] J. Smith III, “Synthesis of bowed strings,” in *Proceedings of the International Computer Music Conference*, Venice, Italy, 1982.
- [3] C. Roads, *The Computer Music Tutorial*. Cambridge, MA: MIT Press, February 1996.
- [4] N. Castagne and C. Cadoz, “Creating music by means of ‘physical thinking’: The musician oriented Genesis environment,” in *Proc. 5th Internat’l Conference on Digital Audio Effects*, Hamburg, Germany, Sept. 2002, pp. 169–174.
- [5] N. Castagné and C. Cadoz, “A goals-based review of physical modeling,” in *Proceedings of the International Computer Music Conference*, Barcelona, Spain, Sept. 5-9 2005.
- [6] I. Peretz, D. Gaudreau, and A.-M. Bonnel, “Exposure effects on music preference and recognition,” *Memory and Cognition*, vol. 26, no. 5, pp. 884–902, 1998.

- [7] C. Cadoz, “The physical model as metaphor for musical creation. pico..TERA, a piece entirely generated by a physical model,” in *Proceedings of the International Computer Music Conference*, Göteborg, Sweden, 2002.
- [8] A. Luciani: Ergotic/Epistemic/Semiotic Functions, *Enaction and Enactive Interfaces: A Handbook of Terms*, A. Luciani and C. Cadoz, Eds. Grenoble, France: Enactive Systems Books, ISBN 978-2-9530856-0-0, 2007.
- [9] A. Luciani: Instrumental Interaction, *Enaction and Enactive Interfaces: A Handbook of Terms*, A. Luciani and C. Cadoz, Eds. Grenoble, France: Enactive Systems Books, ISBN 978-2-9530856-0-0, 2007.
- [10] J. J. Gil and J.-L. Florens: Stability, *Enaction and Enactive Interfaces: A Handbook of Terms*, A. Luciani and C. Cadoz, Eds. Grenoble, France: Enactive Systems Books, ISBN 978-2-9530856-0-0, 2007.
- [11] K. Kuchenbecker, J. Park, and G. Niemeyer, “Characterizing the human wrist for improved haptic interaction,” in *Proceedings of the International Mechanical Engineering Congress and Exposition*, Washington, DC, USA, Nov. 2003.
- [12] O. Brune, “Synthesis of a finite two-terminal network whose driving-point impedance is a prescribed function of frequency,” *Journal of Mathematical Physics*, vol. 10, pp. 191–236, 1931.
- [13] J. Hahn, T. Edison, and T. Edgar, “A note on stability analysis using Bode plots,” *Chemical Engineering Education*, vol. 35, no. 3, pp. 208–211, 2001.
- [14] J. Q. Sun, “Some observations on physical duality and colocation of structural control sensors and actuators,” *Journal of Sound and Vibration*, vol. 194, no. 5, pp. 765–770, 1996.
- [15] M. Balas, “Direct velocity feedback control of large space structures,” *Journal of Guidance and Control*, vol. 2, pp. 252–253, 1979.
- [16] N. Diolaiti, G. Niemeyer, F. Barbagli, K. Salisbury, and C. Melchiorri, “The effect of quantization and coulomb friction on the stability of haptic rendering,” in *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Pisa, Italy, March 18–20 2005, pp. 237–246.
- [17] J. E. Colgate and N. Hogan, “Robust control of dynamically interacting systems,” *International Journal of Control*, vol. 48, no. 1, pp. 65–88, 1988.
- [18] C. Cadoz, A. Luciani, and J.-L. Florens, “CORDIS-ANIMA: A modeling and simulation system for sound and image synthesis—The general formalism,” *Computer Music Journal*, vol. 17, no. 1, pp. 19–29, Spring 1993.
- [19] J.-L. Florens, A. Voda, and D. Urma, “Dynamical issues in interactive representation of physical objects,” in *Proceedings of EuroHaptics*, Paris, France, July 3–6 2006, pp. 213–219.
- [20] A. Hajian, D. Sanchez, and R. Howe, “Drum roll: Increasing bandwidth through passive impedance modulation,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, April 20–25 1997, pp. 2294–9.
- [21] R. Shadmehr and S. Wise, *The Computational Neurobiology of Reaching and Pointing*. Cambridge, MA: MIT Press, 2005.
- [22] J.-L. Florens, C. Cadoz, and A. Luciani, “A real-time workstation for physical model of multi-sensorial and gesturally controlled instrument,” in *Proceedings of the International Computer Music Conference*, Ann Arbor, MI, USA, July 1998.
- [23] N. Lee, E. Berdahl, G. Niemeyer, and J. Smith III, “TFCS: Toolbox for the feedback control of sound,” in *Acoustics '08: 155th Meeting of the Acoustical Society of America, 5th FORUM ACUSTICUM, and 9th Congrès Francais d'Acoustique*, Paris, France, June 29–July 4 2008, available online at <http://ccrma.stanford.edu/~eberdahl/Projects/TFCS>, Last checked: March 1, 2010.
- [24] R. Anderson and M. Spong, “Bilateral control of teleoperators with time delay,” *IEEE Transactions on Automatic Control*, vol. 34, no. 5, pp. 494–501, May 1989.
- [25] G. Heet, “String instrument vibration initiator and sustainer,” U.S. Pat. 4,075,921, 1978.
- [26] A. Hoover, “Controls for musical instrument sustainers,” U.S. Pat. 6,034,316, 2000.
- [27] E. Berdahl, “Applications of feedback control to musical instrument design,” Ph.D. dissertation, Stanford University, Stanford, CA, USA, December 2009.
- [28] G. Franklin, J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 5th ed. Upper Saddle River, NJ: Prentice Hall, 2005.
- [29] J.-J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice Hall, 1991.
- [30] M. V. Valkenburg, *Introduction to Modern Network Synthesis*. Hoboken, NJ: John Wiley and Sons Inc., 1960.

PHYSICAL MODELING MEETS MACHINE LEARNING: TEACHING BOW CONTROL TO A VIRTUAL VIOLINIST

Graham Percival

Science and Music Research Group
University of Glasgow, UK
graham@percival-music.ca

Nicholas Bailey

Science and Music Research Group
University of Glasgow, UK
nick@n-ism.org

George Tzanetakis

Department of Computer Science
University of Victoria, Canada
gtzan@cs.uvic.ca

ABSTRACT

The control of musical instrument physical models is difficult; it takes many years for professional musicians to learn their craft. We perform intelligent control of a violin physical model by analyzing the audio output and adjusting the physical inputs to the system using trained Support Vector Machines (SVM).

Vivi, the virtual violinist is a computer program which can perform music notation with the same skill as a beginning violin student. After only four hours of interactive training, *Vivi* can play all of Suzuki violin volume 1 with quality that is comparable to a human student.

Although physical constants are used to generate audio with the model, the control loop takes a “black-box” approach to the system. The controller generates the finger position, bow-bridge distance, bow velocity, and bow force without knowing those physical constants. This method can therefore be used with other bowed-string physical models and even musical robots.

1. INTRODUCTION

One well-known problem with physical modelling of musical instruments is control [1] – a musical instrument is a non-linear dynamical system; some physical models even include non-deterministic elements. Producing good sound with such a system is difficult; professional classical musicians spend more than ten years learning their instruments before they can earn a living by performing. Therefore, we can expect that autonomous control of a physical model will require a great deal of research and training.

Rather than trying to reproduce the amount of control exhibited by professional musicians, we created a computer program which can be trained in a manner similar to a human violin student: by receiving feedback from a teacher and “listening” to its output. After four hours of interactive training, *Vivi, the virtual violinist* can play at approximately the level of a student with one year of experience.

This is achieved by asking a human user to classify a series of audio examples created using a violin physical model. The human judgements are used to train Support

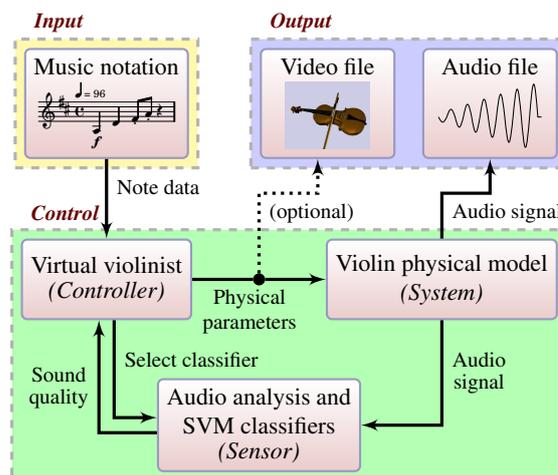


Figure 1. Music performance with *Vivi*.

Vector Machine (SVM) classifiers, which are used to adjust the bowing parameters when performing music (Figure 1). After a short period of “basic training”, *Vivi* can practise scales and pieces of music. For each performance, the human may identify any part of the audio which needs adjustment; these corrections are used to re-train the SVM, which can then perform the music again. Such corrections will influence every subsequent performance; any training on simple scales will apply to a Bach Partita.

1.1 Motivation

At the moment, producing a good violin sound can only be done by highly-trained musicians. Playing a violin with good intonation and good sound quality requires very accurate fine muscle control. This level of muscle control may be taken for granted by professional musicians, but it is a serious problem for skilled amateurs and students; many people who studied classical music as children stop playing music later in life. In addition, violin playing can become a physical impossibility due to advanced age or various medical conditions.

Our goal is to enable anybody to create violin music, regardless of physical disability or lack of training – the only barrier to producing music should be the desire to do so. People who would be satisfied with a “generic” performance should only be required to supply the sheet music, whereas those wishing to customize a performance should only be required to add high-level expressive gestures.

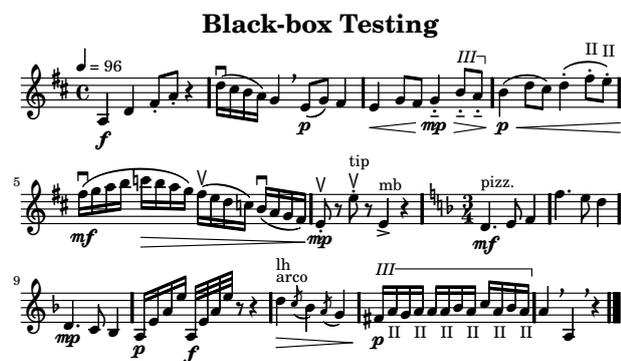


Figure 2. Music notation understood by *Vivi*.

Our work is also motivated by Miku Hatsune and the Vocaloid singing synthesis [2]. By eliminating the physical constraints of singing in a particular vocal range and style (i.e. Japanese pop music), users of this software can create vocal music which was previously impossible for them to produce. This has resulted in an extremely impressive range of music and videos from online collaborations [3].

1.2 Pedagogical inspiration and design goals

As the first author learned cello with the Suzuki method, our design draws upon some of that philosophy:

- Students do not practise alone; a parent should be involved in the daily practice.
- Students learn music from Suzuki violin book 1 [4], with a corresponding emphasis on Baroque music.
- The left-hand fingering is given in the sheet music; although Suzuki students do not read sheet music at the beginning, parents ensure that the student follows the printed fingering.
- The bow-bridge distances are set according to the dynamic. These are sometimes referred to as “bow lanes”, or the “Kreisler highway” [5].
- The amount of bow for each note is often specified by the teacher (such as “half” or “quarter bow”), and the tempo is given by the piano accompaniment.
- Students do not know physical values such as the characteristic impedance Z_c of the string.
- Students are not expected to give expressive music performances; a “robotic” performance is an acceptable place to start.

We do not encourage inexpressive music, but a virtual violinist can be useful even without expressive performances. We again consider Vocaloid to be an inspiration: musicality can be added by the user while the software itself strictly follows the given instructions.

To improve our testing framework, we wrote a piece of music (Figure 2) which contains all the supported notation.

1.3 Sound and video examples

Examples of sheet music performed by *Vivi* are available ¹. In addition to music, the website contains audio examples which illustrate technical aspects of this paper.

¹ <http://percival-music.ca/smc2011.html>

Section 2 gives an overview of related work. The rest of the paper is generally structured from the fastest units of time to the slowest. Physical modeling (which generates samples at 44100 Hz) is described in Section 3. The control cycle (which modifies physical actions at 172 Hz) is covered in Section 4. Training (which does not occur at a constant rate) is covered in Section 5, while Section 6 covers performance rules (dealing with score events occurring at approximately 1-4 Hz). We end with a few implementation details in Section 7 and the conclusion in Section 8.

2. RELATED WORK

Intelligent control of physical bowing parameters would be impossible without a synthesis engine. We chose to use the bowed-string physical modelling algorithm with modal synthesis as described in Demoucron’s Ph.D. thesis [6]. This algorithm does not include some of the more advanced knowledge of bow-string friction interaction [7], the effects of torsional waves in a bowed string [8], or the plastic thermal model of rosin friction [9], but it sounds sufficiently “violin-like” for our needs. The main alternative to modal synthesis is digital waveguide synthesis [7].

An alternate method of creating violin sound, without using a physical model, is concatenative synthesis (or Spectral Modeling Synthesis). This is an active research topic, with recent dissertations on predictive spectral envelopes for violin sounds [10], reconstructing violin bowing with Bézier curves [11], and further research combining these projects to synthesize violin sound [12].

There has been considerable interest in the bowing actions required to establish and maintain a good violin tone. An early examination of the bow force required to reach Helmholtz motion gave rise to “Schelleng diagrams” [13], while an examination of the initial attacks produced “Guettler diagrams” [14]. These have been re-examined with more accurate measurements from real musicians and bowing machines [15]. Some teachers are applying this scientific knowledge to violin pedagogy [5].

Other work has focused on the control of musical instruments. Proportional-integral-derivative (PID) controllers have been used to control a plucked string simulation and acoustic instruments [16, 17]. The famous flute-playing robot WF-4RIV [18] uses an artificial neural network to generate expressive music, and uses a control loop based on the relative strength of even- and odd-harmonics, and the overall sound intensity. The WF-4RIV also remembers its own mistakes while performing a piece, and adjusts future performances of that piece of music accordingly. Another group worked on a robotic violin bowing arm [19]; this project focused on reproducing the same sound when bowing the open D string multiple times, and analyzed the resulting sound by visually comparing the amplitude plots and spectrograms.

Finally, objective analysis of violin tone quality has been performed, allowing computers to classify sounds with high accuracy. One project focused on judging sound quality of violin instruments [20], while another classified a *legato* violin note as being performed by a professional musician or a student, and recognized mistakes in bow control [21].

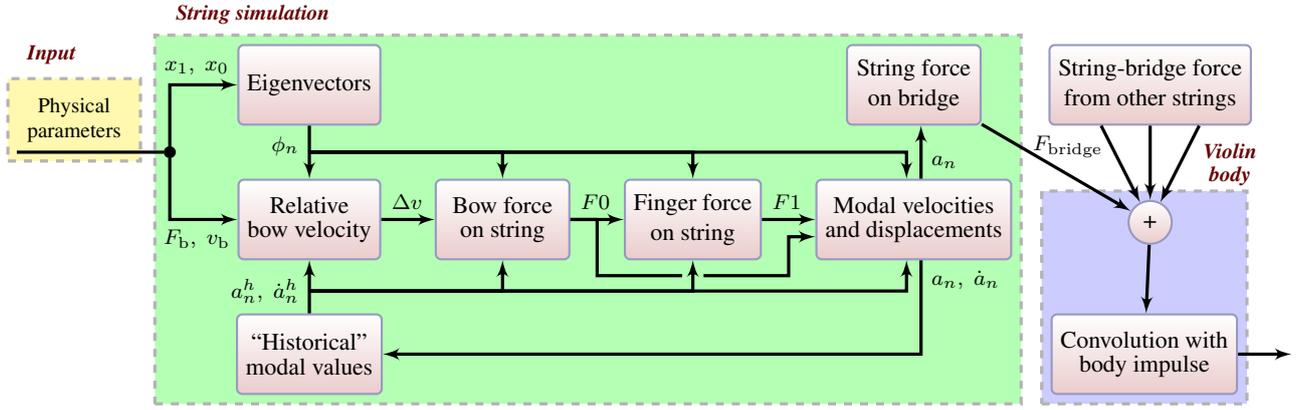


Figure 3. Overview of the violin physical modelling. Variables with a subscript_{*n*} indicate an N-element vector, lines without any text indicate samples of an audio signal, and all other variables are individual numbers.

3. VIOLIN PHYSICAL MODELING

We wrote Artifastring (“artificial fast string”), which implements the bowed-string physical modelling with modal synthesis as described in [6]. Artifastring was written in C++, and is freely available² under the GPLv3. We hope that it can be of use to other researchers without detailed knowledge of acoustics, so that more people can work on bowed string performance. To support widespread usage, SWIG bindings³ are also available.

An overview of the physical model is given in Figure 3, and a summary of the variables used is given in Table 1. We will cover only the most important equations here; for the full details, see [6] or the Artifastring source code.

The synthesis begins with a stiff string with linear density ρ_L , tension T , Young’s modulus E , diameter d , and second moment of area for a circular cross-section $I = \frac{\pi d^4}{64}$. With the sum of external forces $F(x, t)$, the wave equation is:

$$\rho_L \frac{\partial^2 y(x, t)}{\partial t^2} - T \frac{\partial^2 y(x, t)}{\partial x^2} + EI \frac{\partial^4 y(x, t)}{\partial x^4} = F(x, t) \quad (1)$$

This gives the dispersion relation with string length L :

$$\omega_{0n} = \sqrt{\frac{T}{\rho_L} \left(\frac{n\pi}{L}\right)^2 + \frac{EI}{\rho_L} \left(\frac{n\pi}{L}\right)^4} \quad (2)$$

We define the modal displacement $a_n(t)$ and force $f_n(t)$ in terms of $\phi_n(x)$. We then add a damping coefficient $r_n = B_1 + B_2(n-1)^2$ to simulate various losses along the string. Solving (1) and (2) gives us an infinity of equations, $n = 1 \dots \infty$. We limit n to 100 as a compromise between sound quality and computational efficiency.

$$\ddot{a}_n(t) + 2r_n \dot{a}_n(t) + \omega_{0n}^2 a_n(t) = \rho_L^{-1} f_n(t) \quad (3)$$

The calculation of each time step begins with $a_n^h(t)$ and $\dot{a}_n^h(t)$; these are the new modal values if no external forces are applied. To maintain a consistent terminology with [6], we use his term “historical” and the ^h superscript, but we believe that “free oscillation” would be more clear.

² <http://percival-music.ca/artifastring/>

³ <http://www.swig.org/>

The external force F_0 on the string therefore represents the extra force that must be applied to make the string at point x_1 move as described by the relative bow velocity Δv . If the bow is sticking to the string, $\Delta v = 0$. If the bow is slipping, a hyperbolic friction curve is used with the coefficients of static friction μ_s , dynamic friction μ_d , slope of the hyperbolic curve v_0 . Noise was added by picking a random value $0.95 \leq N(t) \leq 1.0$ for each calculation.

$$F_0 = \text{sign}(v_b) \left(\mu_d + \frac{\mu_s - \mu_d}{1 + \frac{|\Delta v|}{v_0 N(t)}} \right) F_b \quad (4)$$

We can also express this force using variables C_{01} , C_{02} , v_0^h , and y_1^h , but their definitions are too complicated to give here; see [6]. Equations (4) and (5) are solved together; if no real solution exists, or if $v_b \Delta v < 0$, then we reject the solution and set the bow to be sticking ($\Delta v = 0$).

$$F_0 = C_{01}(\Delta v + v_b - v_0^h) + C_{02}y_1^h \quad (5)$$

The finger force F_1 is modelled as an infinitely stiff spring at x_1 . Once $a_n(t)$ has been calculated, the bridge signal is:

$$F_{\text{bridge}}(t) = \sqrt{\frac{2}{L}} \sum_{n=1}^N a_n(t) \left(\frac{Tn\pi}{L} + EI \left(\frac{Tn\pi}{L} \right)^3 \right) \quad (6)$$

These calculations are performed once for every string, then the bridge signals are added together. The result of the sum is convolved with the impulse response of tapping a violin bridge; the output is our final audio signal.

| | |
|-------------|--|
| $y(x, t)$ | String displacement at position x , time t |
| $\phi_n(x)$ | Eigenvectors; $\phi_n(x) = \sqrt{\frac{2}{L}} \sin \frac{n\pi x}{L}$ |
| $a_n(t)$ | Modal displacement |
| x_0 | Bow position (as a ratio of string length) |
| x_1 | Finger position (ratio of string length) |
| v_b | Bow velocity (m/s) |
| F_b | Bow force (N) |
| s | String number (selects which string to use) |

Table 1. Main variables used in bowed-string algorithm.

4. VIOLINIST'S CONTROL

The virtual violinist must translate musical note data into physical actions. Due to the design goals outlined in Section 1.2, we cannot use theoretical equations (such as Schelleng's "area of stability" [13] or Guettler diagrams [14]) which rely on physical constants. We also avoid "hand-tweaking" any values – the goal is to create a virtual violinist which can perform on any instrument (be it a physical model or a violin-playing robot), so the algorithm should be as general as possible.

Another difficulty is the non-deterministic element $N(t)$ in the physical model – a series of physical actions may produce acceptable or unacceptable output depending on the random seed. To address this problem, a feedback control loop is used (Figure 4).

The violin is a non-linear system; the effects of bowing parameters are not easy to state. A rough generalization is that the bow-bridge distance and velocity together set the overall amplitude, while the bow force determines the timbre and whether Helmholtz motion is achieved [15].

4.1 Pedagogical inspiration and physical parameters

The string s and finger position x_1 can be calculated directly from each note, imitating the "fingerboard tape" often used with students. Although skilled violinists choose different fingerings for expressive purposes, this is not expected from beginning students.

The "bow lanes" allow us to specify the bow-bridge distance x_0 according to the dynamic. The length of bow and note duration fixes the average bow speed. This still allows for variation in the bow speed, but in general beginning violinists do not concern themselves with this. For simplicity we have assumed the bow will accelerate to the target velocity at the beginning of the note, and decelerate to 0 m/s at the end of the note unless it is part of a slur. Parameters x_0 and v_b are then specified from the dynamic (Table 2). More details about these parameters are given in Section 6.

The missing parameter from our description of beginning violin students is the bow force F_b . Our virtual violinist therefore concentrates on bow force, and uses human input during the training to improve the bow force calculations.

4.2 Bow force

Section 5.2 discusses how we calculate our initial force F_b and force modifier K during training. In the control loop, we alter the current F_b according to the selected SVM classifier, which examines the audio output of the physical model. Our audio analysis and machine learning is

| Dynamic | Bow position x_0 (fraction of L) | Bow velocity v_b (m/s) |
|-----------|--|-----------------------------|
| <i>f</i> | 0.08 | 0.4 |
| <i>mf</i> | 0.10 | 0.33 |
| <i>mp</i> | 0.12 | 0.26 |
| <i>p</i> | 0.14 | 0.2 |

Table 2. Physical parameters determined by dynamic.

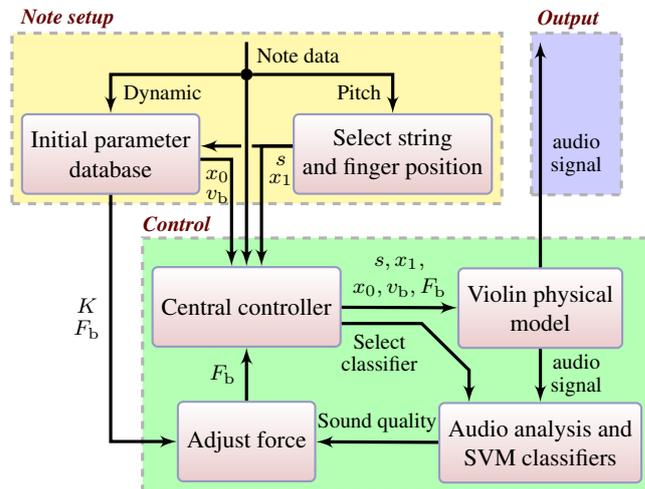


Figure 4. Control details of *Vivi*.

done with Marsyas [22]. In addition to a typical set of audio features (zero crossings, spectral centroid, rolloff, flux, crest factor, and flatness measure), we also calculate the difference between expected pitch (from s and x_1) and the detected pitch (from the YIN algorithm [23]). We used a window size of 1024, and a hop size of 256. The machine learning produces a class $c \in \{1, 2, 3, 4, 5\}$ (Table 3), which is used to adjust the bow force: $F_b \leftarrow F_b K^{(3-c)}$.

4.3 Central controller

The central controller selects which SVM classifier to use; we train a different classifier for each of the sixteen string-dynamic combinations. The controller also handles the acceleration of the bow, and adds a small amount of Gaussian randomization to F_b and v_b . This imitates the unsteady muscle control of a beginning violinist.

A different SVM classifier was used for each string-dynamic combination because each pair produces a different timbre. The 10-fold cross-validation accuracy supports our intuition: after four hours of training, the individual string-dynamic classifiers were 96–99% accurate, while a single classifier for all G string dynamics was 91% accurate, and a single classifier for all strings played *mf* was 94% accurate.

5. TRAINING

We use human input to help train *Vivi*, but only at the level of a "Suzuki parent" – the human gives feedback about audio (classifying audio as in Table 3), but the precise determination of parameters is done automatically.

| Class c | Human judgement of sound |
|-----------|--|
| 1 | not audible |
| 2 | "wispy" or "whistling" |
| 3 | acceptable |
| 4 | "harsh" or "detuned" |
| 5 | not recognizable as coming from a violin |

Table 3. Classification of audio files.

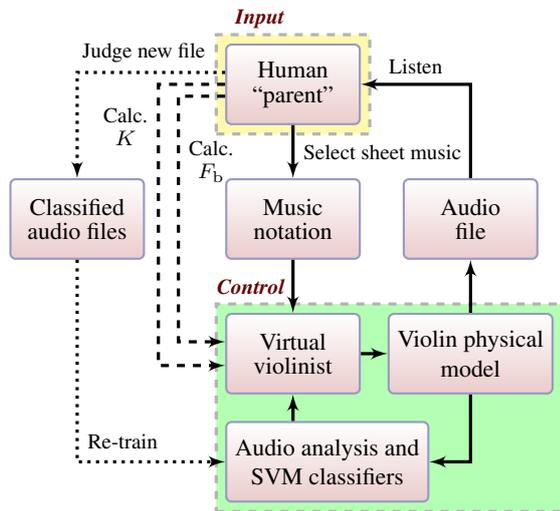


Figure 5. Post-basic training with *Vivi*. The dotted line does not involve the control cycle; the dashed lines involve the control cycle but not the audio output.

5.1 Basic training (human)

Training begins by asking the human to classify audio with a constant bow force. Each audio file is created by running the physical model for 1.0 seconds (to let the note “settle”), then recording the next 0.5 seconds. We begin with $F_b = 1.0$, then keep on doubling F_b after each audio file is categorized until the user has ranked a file as being category 5. We then return to $F_b = 0.5$ and keep on halving F_b until the user has ranked a file as being category 1. If any categories were omitted, then we keep on generating new audio files between existing category boundaries until we have at least one file for each category.

We repeated this process for three notes (seen in Figure 6) for each string / dynamic combination. This produced 15–20 audio files for each of the 16 SVM classifiers, with very high accuracy (98–99% for dynamics on the lower three strings, with the E string dynamics’ accuracy dropped to 96–97%) with 10-fold cross-validation.

We found that “Anomalous Low Frequencies” (ALF, sometimes referred to as “subharmonics” [24]) were produced quite often on the E string; in one case, we thought that an ALF note was a nicely-played note on the D string! For this reason, we displayed the playing string to the human so that they could correctly identify an ALF note as having too much bow force.



Figure 6. Musical patterns used in training the D string. Repeat signs indicate a variable number of repetitions, depending on human input or the algorithm.

5.2 Determining bow force K and F_b (automatic)

Figure 5 shows an overview of the training process after basic training. The missing parameters for our feedback control (Figure 4) are K and F_b , which we determine automatically from our classified audio files.

5.2.1 Calculating K (one per string-dynamic)

Recall that K determines how much F_b is adjusted according to the sound quality judgement. If K is too small, it will take a long time to reach a good bow force. If K is too large, F_b will oscillate between too much and not enough force. We begin by estimating three initial F_b values based on the maximum force used in the “basic training”, F_{max} . We set $F_b = \{\frac{1}{16}F_{max}, \frac{1}{4}F_{max}, F_{max}\}$. For example, a D string played *mf* gives $F_b = \{0.25N, 1.0N, 4.0N\}$.

We define a note’s stability cost as (7), in terms of the list C which is all the class values c produced by the trained SVM classifier. C is then split into sub-lists A_i , in which a sub-list (“area”) is defined by c changing from below 3 to above 3 (or vice-versa). For example, the list $C = [2, 3, 2, 4, 5, 3, 2]$ would become $A = [[2, 3, 2], [4, 5, 3], [2]]$.

$$\text{Note cost} = \prod_i^{|A|} \sum_{c \in A_i} (3 - c)^2 \quad (7)$$

The overall logic is that extreme sounds (categories 1 and 5) are worse than slightly bad sounds (categories 2 and 4). The ideal system would begin with a force below (or above) the correct amount, then quickly move to a good force (category 3). Sounds which alternate between categories 1,2 and 4,5 will maximize the overall multiplication.

We then perform the pattern in Figure 6, once for each initial F_b , and multiply all the note costs together. The resulting number has a large amount of variation, so we repeat this whole process 12 times and take the inter-quartile geometric mean; this is the final cost of a candidate K (Figure 7). We select the candidate K with the lowest cost.

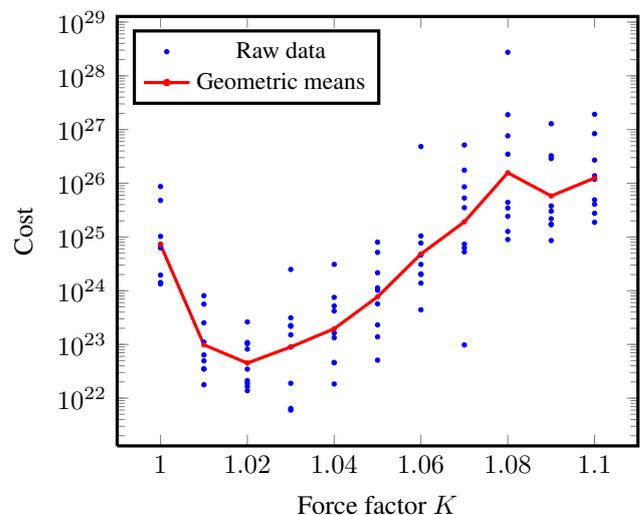


Figure 7. Sample force factors of the D string played *mf*. Extra points were added to show the spread of randomness.

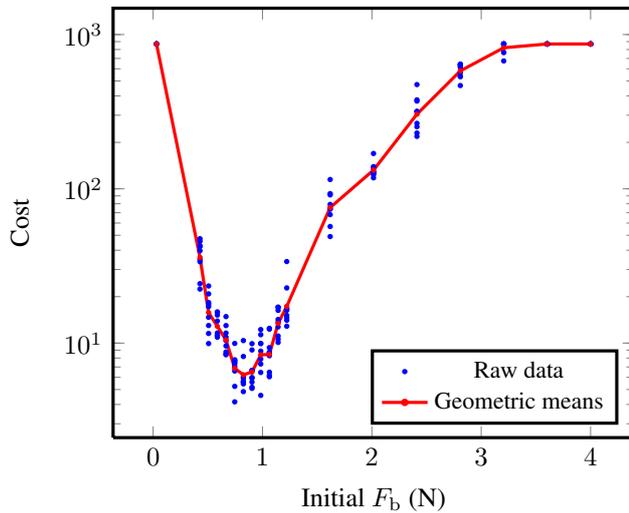


Figure 8. Sample initial forces of the D string played *mf* with an open string, including the “zoomed-in” area. Extra points were added to show the spread of randomness.

5.2.2 Calculating initial F_b (three per string-dynamic)

After calculating K , the real (i.e. not estimated) initial F_b values are calculated from the range F_{\min} to F_{\max} . As with the calculation for K , we calculate a candidate’s cost by playing a musical pattern (1 bar) from Figure 6. The F_b cost (8) is straightforward, but in this case C only represents the classifier values c that were part of the “attack” portion of the note. The attack is over when (9) is true, where L_N is a list of the past $N = 9$ values, and $M = 0.5$. This corresponds to “when the note is stable”.

$$\text{Note cost} = \sum_{c \in C} (3 - c)^2 \quad (8)$$

$$M > \frac{1}{N} \sum_{c \in L_N} (3 - c)^2 \quad (9)$$

The individual note costs are multiplied together, then the process is repeated 4 times, and the inter-quartile geometric mean is the overall cost of the candidate F_b . After finding the lowest cost, we “zoom in” to that area by setting F_{\min} to the F_b immediately lower than our lowest value, and F_{\max} to the F_b higher. The process is repeated, gathering more candidates (Figure 8). We select the candidate F_b with the lowest cost.

This whole process is repeated three times, varying the left-hand finger positions as seen in Figure 6. For the sample D string *mf*, the initial F_b drops from 0.8 N (for an open string) to 0.4 N (for a finger 4 semitones higher) to 0.3 N (finger 7 semitones above open string).

Violinists will notice that this generates an “on the string” bow-stroke. Skilled violinists will vary the amount of initial bow-force considerably, including starting with $F_b = 0$ for an “off the string” bow-stroke. However, the first bow-stroke taught by the Suzuki method is “on the string” – in fact, this bow-stroke is a large contributor to the “Suzuki sound” for which their students are so famous!

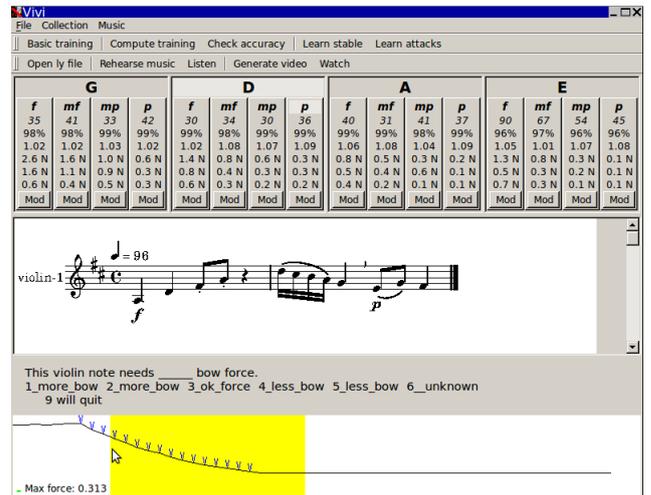


Figure 9. Screenshot of *Vivi*. From top to bottom the numbers represent: number of audio files classified, 10-fold cross-validation accuracy, force factor K , and initial F_b for a finger at 0 (open string), 4, and 7 semitones.

5.3 Practicing sheet music (human)

After the human has completed the one-time “basic training” and the computer has completed the automatic determination of K and F_b , the bow control is extremely poor⁴.

To improve the bow control, *Vivi* should “rehearse” sheet music selected by the human, who can then listen to the entire performance, or listen to individual notes by clicking on them. For each note, the user can correct any judgments by selecting a portion of a note and indicating what the category should be; this is shown in Figure 9. After correcting a few notes, the user should tell *Vivi* to re-train the SVM classifier, and then re-calculate K and/or F_b . Any sheet music can be used for this interactive practice, but scales are quite useful as they provide few distractions.

6. MUSIC PERFORMANCE

We now turn to the calculation of the note data which is the input to Figure 4. Rather than providing a list of musical notation, we refer to our “black-box” testing framework in Figure 2, which includes all notation understood by *Vivi*.

6.1 Mapping basic notation to note data

Many aspects of musical notation can be translated directly into physical actions with no debate. In the absence of special string markers (“IL.”), the written pitch will determine the string and left-hand finger position. A notated slur means that we should not change bow directions; without a slur, we decelerate the bow speed to 0 m/s at the end of the note, and set the next note’s direction to be the opposite of the current note.

The markers “pizz.” and “arco” are similarly clear – notes that should be pizzicato are simulated as having a very large F_b , pulling the string until the “bow” slips, and then setting F_b to be 0. Arco notes return to the normal bowing.

⁴ <http://percival-music.ca/smc2011.html>

| Notation | Musical action | Value |
|-----------------------|-------------------|----------|
| Staccato | Shorten duration | 0.7 |
| Portato | Shorten duration | 0.9 |
| Accent | Increase force | 2.5 |
| Last note on a string | Lighten bow force | 0.1 s |
| " | " | $0.5F_b$ |
| Breath mark | Shorten duration | 0.6 |
| " | " | $0.3F_b$ |
| (all) | Max. bow accel. | 5.0 m/s |

Table 4. *Vivi*'s interpretation of style-specific notation. Unless units are given, all values are expressed as a factor of the unmodified value which would be used.

Text such as “frog”, “lh”, “mb”, “uh”, “tip” refers to the position of contact along the bow; these are specified as 0.1, 0.25, 0.5, 0.75, and 0.9. An upbow or downbow mark will set the bowing direction as specified.

6.2 Mapping style-specific notation to note data

Other aspects of musical notation are subject to interpretation or musical styles. For now, we simply hard-coded values for stylistic interpretation, as our focus was on control and training rather than expressive music performance. We listened to *Vivi*'s performances of book 1 Suzuki music [4] and adjusted parameters to match the way we expected the pieces to sound. Values are given in Table 4.

(*De*)*crescendi* are interpreted as linear interpolation between the beginning and ending dynamics. The physical parameters of fractional dynamics are similarly interpreted as a linear interpolation of the two closest dynamics.

6.3 Variable skill

The timing and intonation can be deliberately degraded to simulate a beginning violinist by applying Gaussian randomization with standard deviation σ to the note's pitch and onset time. If a note occurs on an open string, then we apply no randomization to the pitch, but still alter the timing. We defined 4 skill levels, expressed in pairs of $(\sigma_{pitch}, \sigma_{timing})$ from worst to best: (0.5, 0.01), (0.1, 0.005), (0.05, 0.001), (0, 0). Values were estimated by listening.

7. IMPLEMENTATION

Vivi was written primarily in python using the pyqt4⁵ bindings, and is freely available⁶ under the GPLv3. Audio analysis and machine learning was performed with Marsyas, while the “training” functionality was written in C++ and made available to the main application with SWIG.

Sheet music was written in LilyPond⁷. In addition to creating PDFs, we extracted musical events from the score by writing event listeners in scheme and attaching them to the relevant portions of the sheet music engraving process.

⁵<http://www.riverbankcomputing.com/static/Docs/PyQt4/html/>

⁶<http://percival-music.ca/vivi.html>

⁷<http://lilypond.org>



Figure 10. Frame of automatically-produced video. Many thanks to Marcos Press for creating the violin and bow models in Blender, and releasing it under the GPLv3.

The audio analysis and machine learning provides a significant performance penalty compared to running the physical model by itself. Generating the 29 seconds of audio for the music in Figure 2 with the physical model alone took⁸ 4 seconds, while the complete feedback control took 17 seconds. Calculating K and the initial F_b forces for each string-dynamic was a more substantial bottleneck, taking 326 seconds. Fortunately, this task is embarrassingly parallel, so on a processor capable of running four threads at once it takes approximately 25 minutes to complete all automatic training. In addition to creating an audio file, we also record all physical actions to a file. This can be used to generate a video with Blender⁹ (Figure 10).

8. CONCLUSIONS AND FUTURE WORK

We have presented *Vivi*, *the virtual violinist*; a computer program which can perform sheet music with approximately the skill of a violinist with one year of experience. Intelligent control of the violin physical model is performed with SVM classifiers, and some physical parameters were pre-computed by simulating the performance of the control loop before performing sheet music.

Some major facets of violin music are lacking – we do not perform any glissandi, chords, or vibrato. This is no accident; these skills are not taught to beginning Suzuki violin students. We plan to follow the path of human students, adding features in the same order in which human students learn them. On a similar note, so far we have used the same “style” for all sheet music. This worked for the mostly-Baroque repertoire of book 1, but it will not suffice for later volumes of Suzuki repertoire. We will therefore add the ability for the user to select different “styles”, or even automatically select a style based on the composer.

One possibility to improve the control loop is to add haptic feedback to the machine learning. Human violinists using haptic feedback with a violin physical model can exert

⁸ Measured on an Intel Core2 Quad CPU running Ubuntu 10.04.

⁹<http://www.blender.org/>

much better control [25]. However, relying on haptic feedback may complicate any future research in using *Vivi* with a robot violinist. Another possibility to improve the sound quality is to perform more training, and/or investigate the use of other machine learning algorithms.

Finally, we would like to teach *Vivi* how to play viola and cello. The physical model should be able to simulate other bowed string instruments, and our supervised training should be easily applicable to other instruments. Our ultimate goal is to allow composers to generate audio for an entire string quartet, given only the sheet music.

9. REFERENCES

- [1] J. O. Smith, "Physical Modeling Synthesis Update," *Computer Music Journal*, vol. 20, no. 2, pp. 44–57, 1996.
- [2] H. Kenmochi and H. Ohshita, "VOCALOID – commercial singing synthesizer based on sample concatenation," in *Interspeech*, 2007.
- [3] M. Hamasaki, H. Takeda, and T. Nishimura, "Network analysis of massively collaborative creation of multimedia contents: case study of hatsune miku videos on nico nico douga," in *UXTV '08*. New York, NY, USA: ACM, 2008, pp. 165–168.
- [4] S. Suzuki, *Violin Part Volume 1*. Summy-Birchard Music, 1978.
- [5] C. D. Collins, "Connecting Science and the Musical Arts in Teaching Tone Quality: Integrating Helmholtz Motion and Master Violin Teachers' Pedagogies," Ph.D. dissertation, George Mason University, 2009.
- [6] M. Demoucron, "On the control of virtual violins: Physical modelling and control of bowed string instruments," Ph.D. dissertation, IRCAM, Paris, 2008.
- [7] S. Serafin, "The sound of friction: real-time models, playability and musical applications," Ph.D. dissertation, CCRMA, Stanford University, 2004.
- [8] E. Bavu, J. Smith, and J. Wolfe, "Torsional waves in a bowed string," *Acta Acustica – Acustica*, vol. 91, N2, p. 241, 2005.
- [9] J. Woodhouse and P. Galluzzo, "The Bowed String As We Know It Today," *Acta Acustica – Acustica*, vol. 90, pp. 579–589, July/August 2004.
- [10] A. Pérez, "Enhancing Spectral Synthesis Techniques with Performance Gestures using the Violin as a Case Study," Ph.D. dissertation, Universitat Pompeu Fabra, 2009.
- [11] E. Maestre, "Modeling instrumental gestures: an analysis/synthesis framework for violin bowing," Ph.D. dissertation, Universitat Pompeu Fabra, 2009.
- [12] E. Maestre, A. Pérez, and R. Ramírez, "Gesture sampling for instrumental sound synthesis: violin bowing as a case study," in *International Computer Music Conference*, 2010.
- [13] J. C. Schelleng, "The bowed string and the player," *Journal of the Acoustical Society of America*, vol. 53, no. 1, pp. 26–41, 1973.
- [14] A. A. K. Guettler, "Acceptance limits for the duration of pre-Helmholtz transients in bowed string attacks," *Journal of the Acoustical Society of America*, vol. 101, pp. 2903–2913, 1997.
- [15] E. Schoonderwaldt, "Mechanics and acoustics of violin bowing: Freedom, constraints and control in performance," Ph.D. dissertation, KTH, Sweden, 2009.
- [16] E. Berdahl, G. Niemeyer, and J. O. Smith, "Feedback Control of Acoustic Musical Instruments," *Technical Report STAN-M-120*, no. 120, June 2008.
- [17] E. Berdahl and J. O. Smith, "Inducing Unusual Dynamics in Acoustic Musical Instruments," in *International Conference on Control Applications*, 2007, pp. 1336–1341.
- [18] J. Solis, K. Taniguchi, T. Ninomiya, K. Petersen, T. Yamamoto, and A. Takanishi, "Improved musical performance control of WF-4RIV: Implementation of an expressive music generator and an automated sound quality detection," in *Robot and Human Interactive Communication*, Aug. 2008, pp. 334–339.
- [19] K. Shibuya, S. Matsuda, and A. Takahara, "Toward Developing a Violin Playing Robot – Bowing by Anthropomorphic Robot Arm and Sound Analysis," in *Robot and Human interactive Communication*, 2007, pp. 763–768.
- [20] P. Wrzecziono and K. Marasek, "Violin Sound Quality: Expert Judgements and Objective Measurements," in *Advances in Music Information Retrieval*, Z. Ras and A. Wiczkowska, Eds. Springer Berlin / Heidelberg, 2010, vol. 274, pp. 237–260.
- [21] J. Charles, "Playing Technique and Violin Timbre: Detecting Bad Playing," Ph.D. dissertation, Dublin Institute of Technology, 2010.
- [22] G. Tzanetakis, "Marsyas: a case study in implementing Music Information Retrieval Systems," in *Intelligent Music Information Systems: Tools and Methodologies*, S. Shen and L. Cui, Eds. Information Science Reference, 2007.
- [23] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.
- [24] M. Kimura, "How to produce subharmonics on the violin," *Journal of New Music Research*, vol. 28, pp. 178–184, 1999.
- [25] A. Luciani, J.-L. Florens, D. Couroussé, and J. Castet, "Ergotic Sounds: A New Way to Improve Playability, Believability and Presence of Virtual Musical Instruments," *Journal of New Music Research*, vol. 38, no. 3, pp. 309–323, 2009.

PARAMETRIC TROMBONE SYNTHESIS BY COUPLING DYNAMIC LIP VALVE AND INSTRUMENT MODELS

Tamara Smyth, Frederick Scott

School of Computing Science, Simon Fraser University
 tamaras@cs.sfu.ca, fss3@cs.sfu.ca

ABSTRACT

In this work, a physics-based model of a trombone coupled to a lip reed is presented, with the parameter space explored for the purpose of real-time sound synthesis. A highly configurable dynamic lip valve model is reviewed and its parameters discussed within the context of a trombone model. The trombone model is represented as two separate parametric transfer functions, corresponding to tapping a waveguide model at both mouthpiece and bell positions, enabling coupling to the reed model as well as providing the instrument’s produced sound. The trombone model comprises a number of waveguide filter elements—propagation loss, reflection at the mouthpiece, and reflection and transmission at the bell—which may be obtained through theory and measurement. As oscillation of a lip reed is strongly coupled to the bore, and playability strongly dependent on the bore and bell resonances, it is expected that a change in the parameters of one will require adapting the other. Synthesis results, emphasizing both interactivity and high-quality sound production are shown for the trombone in both extended and retracted positions, with several example configurations of the lip reed.

1. INTRODUCTION

In this work, a physics-based model of a trombone is presented, suitable for real-time sound synthesis, emphasizing both interactive control parameters and high-quality sound production.

The sound produced by the trombone may be seen as the coupling of the input pressure from the lips (the product of the volume velocity and the bore opening’s characteristic impedance) with the instrument bore and bell—a convolution of the lip-valve signal and the trombone impulse response.

In previous work [1], a parametric model of the trombone’s transfer function is obtained in two positions: one tapped at the position of the mouthpiece and the other outside the bell. The former may be coupled to a lip-valve model, providing feedback of bore resonances and the pressure difference across the lip valve (required for dynamic models in which the bore pressure influences the behaviour of the vibrating lips [2]), while the latter may be convolved with the lip-valve signal to provide the instrument’s produced sound.

The instrument body model, discussed in Section 2, employs a measurement and a processing technique from pre-

vious work [3, 1], whereby waveguide elements are estimated from several measurements of the system’s impulse response, with the system having incrementally varying boundary conditions to allow for the isolation and estimation of filter transfer functions. The work in [1] focused on obtaining waveguide elements for the trombone instrument model, while the work here focuses on coupling this instrument to a generalized reed model.

The parameter space of a dynamic lip valve model is explored when coupled to a trombone synthesis model. To provide context and present parameters, the valve model is discussed in Section 3. Coupling to the trombone model with and without a mouthpiece is discussed in Section 4.

2. TROMBONE INSTRUMENT MODEL

It is well known that wave propagation in wind instrument bores may be modeled in one dimension using the waveguide structure shown in Figure 1, with a bi-directional delay line of length M samples accounting for the acoustic propagation delay in the cylindrical and/or conical tube section of a given length, and filter elements $\lambda(z)$, $R_0(z)$, $R_L(z)$ and $T_L(z)$, accounting for the propagation loss, reflection at the mouthpiece, and open-end reflection and transmission occurring at the position of the bell, respectively, all of which may contain delays, poles or “long-memory” information on the acoustics of non-cylindrical and non-conical bore sections [4, 5]. That is, the approach shown in Figure 1 separates an instrument horn into its cylindrical/conical and flared sections, with lumped filters accounting for the reflection and transmission of the flared bell—properties which contribute significantly to the instrument’s characteristic resonances.

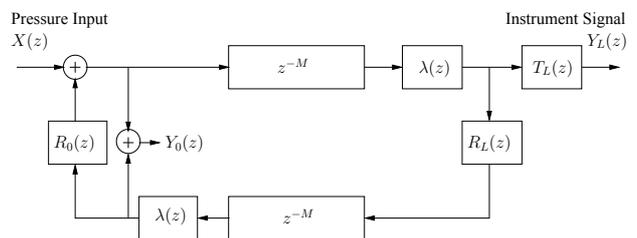


Figure 1. Waveguide model of a cylindrical tube with commuted propagation loss filters $\lambda(z)$, open-end terminating reflection and transmission filters $R_L(z)$ and $T_L(z)$ respectively, and a reflection filter $R_0(z)$ at the (effectively) closed end termination corresponding to the position of the mouthpiece. Two instrument transfer functions (1) and (2) are developed for observation points yielding $Y_0(z)$ and $Y_L(z)$, corresponding to the bore base and the instrument output, respectively, in response to input pressure $X(z)$.

Copyright: ©2011 Tamara Smyth et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

For synthesis applications, it is useful to tap the signal flow diagram in Figure 1 at two different positions, producing pressures $Y_0(z)$ and $Y_L(z)$ in response to input pressure $X(z)$. This yields two separate instrument transfer functions which may be used to couple the instrument to a dynamic lip reed model at the mouthpiece, as well as produce the instrument's sound output at the bell.

As shown in [1], ignoring the time-varying component in the mouthpiece, the global transfer function $H = Y_0/X$ at the bore base is given in the z -domain by

$$H(z) = \frac{Y_0(z)}{X(z)} = \frac{1 + \lambda^2(z)R_L(z)z^{-2M}}{1 - \lambda^2(z)R_L(z)R_0(z)z^{-2M}}, \quad (1)$$

where $\lambda(z)$ is the propagation loss and $R_0(z)$ and $R_L(z)$ are the reflection functions describing the boundaries at the position of mouthpiece and bell, respectively, and where $Y_0(z)$ is based on the power series expansion

$$Y_0(z) = X(z)(1 + \lambda^2(z)R_L(z)z^{-2M}) \times [1 + R_0(z)R_L(z)\lambda^2(z)z^{-2M} + R_0(z)R_L(z)\lambda^2(z)z^{-2M} + \dots].$$

Similarly, the global transfer function $G = Y_L/X$ at the instrument output (at the bell) is given in the z -domain by

$$G(z) = \frac{Y_L(z)}{X(z)} = \frac{T_L(z)\lambda(z)z^{-M}}{1 - \lambda^2(z)R_L(z)R_0(z)z^{-2M}}, \quad (2)$$

where $T_L(z)$ is the transmission function of the bell, and where $Y_L(z)$ is based on the power series expansion

$$Y_L(z) = X(z)T_L(z)\lambda(z)z^{-M} \times [1 + R_0(z)R_L(z)\lambda^2(z)z^{-2M} + R_0(z)R_L(z)\lambda^2(z)z^{-2M} + \dots].$$

Expressing the instrument model in this way conveniently allows for the alternate equivalent representation shown in Figure 2, whereby the input pressure $x(t) = Z_0U(t)$, the product of the characteristic impedance Z_0 and volume flow $U(t)$ derived from a reed model in response to a blowing pressure $p_m(t)$, is convolved with instrument impulse responses $h(t)$ and $g(t)$, the inverse Fourier transforms of the frequency responses corresponding to (1) and (2), respectively. Implementing the model in this way, allows for extensions of the trombone instrument model that are not bounded by the physical constraints of the waveguide model, the basis for convolutional synthesis in [6],

2.1 Trombone Model Parameters

Whether using waveguide or convolutional synthesis implementation, the model described by (1) and (2) comprise several filter elements describing the acoustic characteristics of the system:

- **The delay of M samples** accounts for the acoustic propagation delay in the bore, the value typically being set according to the bore's effected length or the desired sounding pitch.
- **Propagation/wall losses** $\lambda(\omega)$ are well described theoretically [7, pp. 193-196], with a parametric filter described in [8], allowing for real-time changes according to tube size and length.

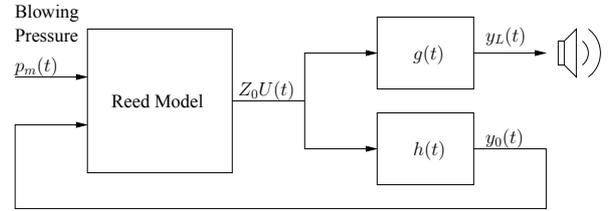


Figure 2. A convolutional synthesis approach to the signal flow diagram shown in Figure 1, with signals $h(t)$ and $g(t)$ being the impulse responses of the instrument tapped at the positions $y_0(t)$ and $y_L(t)$, the inverse transforms of (1) and (2), respectively. The input pressure is the product of the characteristic (wave) impedance Z_0 at the mouthpiece and the volume flow $U(t)$, a signal generated by a reed model in response to a blowing pressure $p_m(t)$.

| Part | Length (cm) | Radius (cm) |
|--------------------------|-------------|-------------|
| t. inner slide (1) | 70.8 | 0.69 |
| t. outer slide, ext. (2) | 53 | 0.72 |
| slide crook (3) | 17.7 | 0.74 |
| b. outer slide, ext. (4) | 53 | 0.72 |
| b. inner slide (5) | 71.1 | 0.69 |
| gooseneck (6) | 24.1 | 0.71 |
| tuning slide (7) | 25.4 | 0.75,1.07 |
| bell flare (8) | 56.7 | 1,10.8 |

Table 1. Trombone tubular sections (numbers correspond to parts in Figure 3) and dimensions, including top (t.) and bottom (b.) inner and outer slides, retracted and extended (ext.).

- **The reflection and transmission at the bell, $R_L(z)$ and $T_L(z)$,** respectively, may be derived either from a computational model or from measurement, with the former emphasizing parametrization and ability to change the bell contour during performance, and with the latter offering assumed greater accuracy. Because the trombone bell is not expected to change during performance, and because it disassembles easily from the trombone bore, its reflection and transmission functions may be estimated using the measurement technique described in [1].
- **The reflection at the mouthpiece position $R_0(\omega)$.** As this is expected to change during performance with the vibrating lips changing both the mouthpiece volume and the opening to the bore, it is not suitably obtained using the methods described in [3, 1], and is better developed within the context of coupling with the dynamic lip reed model discussed in Section 4.

A complete trombone (mouthpiece omitted), is shown in Figure 3, with corresponding trombone components and dimensions provided in Table 1. Figure 3 shows an interior view of the complete trombone in both retracted and extended positions, producing bores with effective lengths of 209.1 cm and 315.1 cm respectively, with asterisks showing possible cylindrical junctions that may or may not be considered depending on the desired level of accuracy (they are omitted here). Trombone components 1-7 in Figure 3 are modeled as a single cylindrical waveguide section, following dimensions in Table 1 for appropriate delay length

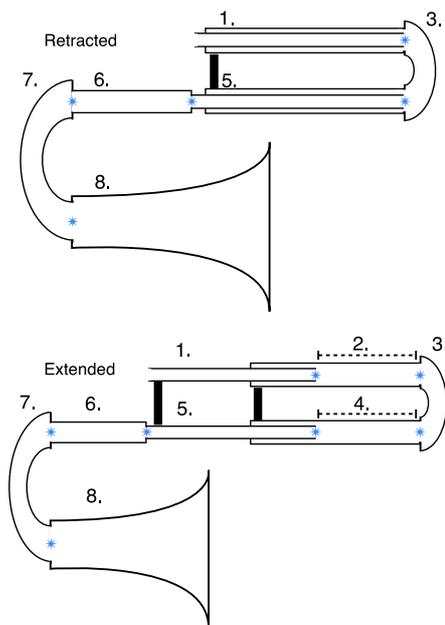


Figure 3. Interior view of trombone, in both fully retracted and fully extended positions, showing assembly of components from Table 1.

and average radius, the parameters required for the propagation loss model described in [8].

3. LIP VALVE MODEL

In reed instruments, input air pressure from the lungs/mouth controls the oscillation of a valve by creating a pressure difference across its surface. The oscillation of a “blown open” valve, a typical characterization of the trombone’s lip reed [9], is strongly coupled to the bore, making playability (and thus a *regular* non-chaotic oscillation of reed), highly dependent on the bore and bell resonances. It is expected, therefore, that the configuration and parameter values of a dynamic reed model would be dependent on the instrument to which it is coupled, as well as any changes—such as an extending slide—occurring during performance.

Here, the generalized pressure-controlled valve model, first introduced in [2], is reviewed to provide context for the configuration and playing control values explored within the context of the trombone model (described in Section 2).

3.1 Generalized Pressure Control Valve

The generalized parametric model of a pressure-controlled valve [2], affords the user the ability to design a continuum of reed configurations, including “blown-open”, “blown-closed”, and the “swinging door”, typically seen in wind and vocal systems, simply by setting model parameters.

Figure 4 illustrates one model of oscillation of the blown open configuration, with the displacement of the valve being given by its angle θ from the vertical axis. The valve classification is determined in part by its initial position θ_0 (its equilibrium position in the absence of flow), and in part by the use of an optional *stop*—a numerical limit placed to

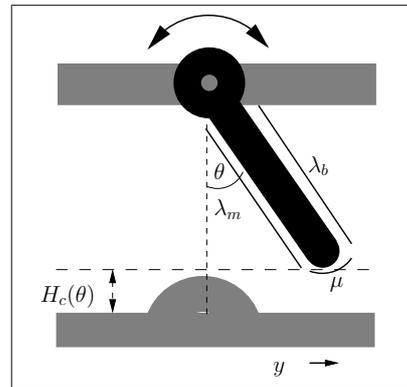


Figure 4. The blown open configuration of the generalized valve model, showing geometric parameters λ_m , the length of the valve that sees the mouth pressure, λ_d , the length of the valve that sees the valve’s downstream pressure, and μ , the length of the valve that sees the flow. Changing these parameters will change the corresponding component forces of the overall driving force $F = F_m + F_b + F_U$.

constrain the range of θ . In Figure 4, the stop is placed at the center vertical axis, $\theta = 0$. If a stop is placed in the channel, the configuration is further determined by the initial equilibrium position of the valve θ_0 : an initial position to the left of the stop, at $\theta_0 < 0$, will cause the reed to *blow closed*, while an initial position to the right of the stop, $\theta_0 > 0$, will cause the reed to *blow open*. The trombone classification, typically considered to be *blown open*, is implemented with $\theta_0 > 0$ plus a stop.

The valve channel area is critical to the volume flow and resulting sound of the instrument. As the reed angle θ changes, the valve opening area A changes according to channel height H_c

$$A(\theta) = wH_c(\theta), \quad (3)$$

where w is the width of the channel. The height of the valve channel in the presence of an oscillating reed may be specified by the function $H_c(\theta)$, a number of possible functions depending on choice of valve. To approximate channel area of a lip reed, the height function may be set to

$$H_c(\theta) = 1 - \cos \theta. \quad (4)$$

The geometry of the valve may be further specified by setting the effective length of the reed that sees the mouth pressure λ_m , the reed length that sees the bore pressure λ_b , and the reed length that sees the flow, given by μ (see Figure Figure 4). These variables have an audible effect on the overall driving force acting on the reed, given by F in (5), and can be seen as offering finer control of embouchure.

Once the valve is set into motion, the value for θ is determined by the second order differential equation

$$m \frac{d^2\theta(t)}{dt^2} + m2\gamma \frac{d\theta(t)}{dt} + k(\theta(t) - \theta_0) = F, \quad (5)$$

where m is the effective mass of the reed, γ is the damping coefficient, k is the stiffness of the reed, and F is the overall driving force acting on the reed, a function of the mouth and bore pressure, and flow in contact with the reed. The frequency of vibration for this mode is given by $\omega_v = \sqrt{k/m}$.

Discretization, equivalent to applying a bilinear transform, yields the transfer function in the z -domain

$$\frac{\theta(z)}{F(z) + k\theta_0} = \frac{1 + 2z^{-1} + z^{-2}}{a_0 + a_1z^{-1} + a_2z^{-2}}, \quad (6)$$

and the corresponding difference equation

$$\theta(n) = [F_k(n) + 2F_k(n-1) + F_k(n-2) - a_1\theta(n-1) - a_2\theta(n-2)]/a_0, \quad (7)$$

where $F_k(n) = F(n) + k\theta_0$, and

$$\begin{aligned} a_0 &= m\alpha^2 + mg\alpha + k, \\ a_1 &= -2(m\alpha^2 - k), \\ a_2 &= m\alpha^2 - mg\alpha + k, \end{aligned}$$

and $\alpha = 2/T$, where T is the sampling period, and $g = 2\lambda$. Since pole frequencies are well below the Nyquist limit (half the sampling rate), there is no need for pre-warping.

The force driving the reed F is equal to the sum of the forces acting on the reed, $F = F_m + F_b + F_U$, where $F_m = w\lambda_m p_m$ is the force acting (in the positive θ direction) on the surface area $\lambda_m w$, $F_b = -w\lambda_b p_b$, is the force acting (in the negative θ direction) on the surface area $\lambda_b w$, and F_U is the force applied by the flow (which forces the reed open) given by

$$F_U = \text{sign}(\theta)w\mu \left(p_m - \frac{\rho}{2} \left(\frac{U(t)}{A(t)} \right)^2 \right). \quad (8)$$

As can be seen by (8), the total force driving the reed is dependent on the valve classification, since the sign of θ is determined by its limits.

The differential equation governing air flow through the valve, fully derived in [10], is given by

$$\frac{dU(t)}{dt} = (p_m - p_b) \frac{A(t)}{\mu\rho} - \frac{U(t)^2}{2\mu A(t) + U(t)T}. \quad (9)$$

where p_m is mouth pressure, p_b is the bore pressure (see discussion in the following section), $A(t)$ is the cross sectional area of the valve channel, and μ is the length of reed that sees the flow. Equation (9) is used to update the flow U every sample period (given by the inverse of the sampling rate).

There are, therefore, three variables that evolve over time in response to an applied mouth pressure p_m : the displacement of the reed θ (determined using 7), the flow U , determined using the update given by (9), and the pressure at the base of the bore p_b , obtained using either waveguide synthesis or a low-latency convolution [11], as mentioned in Section 2.

Since the contour of the bore and bell to which the reed is connected strongly influences the reed's oscillation, the valve model must have a new configuration and set of parameter values for each new instrument application.

4. COUPLING LIP AND INSTRUMENT MODELS

Connecting the lip model to the instrument can be done in two ways: The first would be more simplistic and would omit a mouthpiece. The second would be to introduce a new element, the mouthpiece (described below), to account for the resonance created by the mouthpiece's cup volume and its backbore constriction.

4.1 Mouthpiece Model

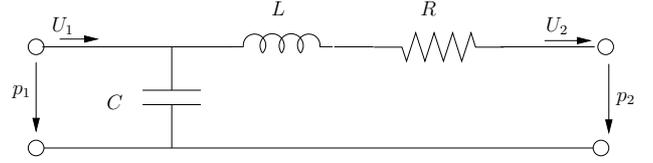


Figure 5. The system diagram for the mouthpiece model. The cup volume is represented by the capacitor, and the attached narrow constriction is modeled as a series inductance L and dissipative element R , which accounting for wall losses.

Improvements to the trombone model may be made by including the effects of a mouthpiece, an important structural element to the complete trombone instrument, which introduces a characteristic resonance determined by the combination of the cup volume and constriction diameter of the backbore [7]. As shown in [7, 12, 13] (and others), the mouthpiece may be modeled by the equivalent electrical circuit shown in Figure 5. The mouthpiece consists of a cup having a volume V , and presents an acoustic compliance C given by

$$C = \frac{V}{\rho c^2}, \quad (10)$$

where ρ is the air density and c is the velocity of sound in air. The cup is followed by a constricted passage before entering into the wider trombone bore, with the constriction behaving as a series inductance (inductance in electrical terms) given by

$$L = \frac{\rho l_c}{S_c}, \quad (11)$$

where l_c is the length and S_c is the cross-sectional area of the constriction. The dissipative element R in series with this inductance represents viscous and thermal losses. Its value for the mouthpiece has been obtained through experiment in [14].

Inserting a mouthpiece between the reed and bore models requires a new expression for the volume flow entering the bore (it is no longer that coming directly from the lips), as well as a new expression for the downstream pressure used when in the dynamic lip reed model (it is no longer the bore base pressure). These quantities are termed $U_2(t)$ and $p_1(t)$, respectively, in Figure 5.

The mouthpiece model provides a volume flow $U_2(t)$ into the bore and a pressure $p_1(t)$ in the mouthpiece, in response to a volume flow $U_1(t)$ entering the mouthpiece (generated by the lip reed model) affixed to the instrument having a pressure of $p_2(t)$ at the bore base.

Taking the Laplace transform of the differential equations describing the mouthpiece model in Figure 5 leads to the system's frequency domain input-output matrix

$$\begin{bmatrix} U_1(s) \\ p_1(s) \end{bmatrix} = \begin{bmatrix} s^2 LC + sRC + 1 & sC \\ sL + R & 1 \end{bmatrix} \begin{bmatrix} U_2(s) \\ p_2(s) \end{bmatrix}, \quad (12)$$

which may be rearranged and discretized to yield expressions for U_2 and p_1 in response to U_1 and p_2 , given in the z -domain as

$$U_2(z) = \frac{U_1(z)(1 + 2z^{-1} + z^{-2}) - C\alpha p_2(z)(1 - z^{-2})}{a_{m0} + a_{m1}z^{-1} + a_{m2}z^{-2}}, \quad (13)$$

| Quantity | Variable | Value |
|-------------------------|-------------------------|-------------------------------|
| radius of exhaust | a | 8mm |
| valve width | w | 2.3 mm |
| valve length | $\lambda_m = \lambda_b$ | 23.2 mm |
| valve mass | m | .3g |
| valve thickness | μ | 6mm |
| initial displacement | θ_0 | 0.01mm |
| mouthpiece volume | V | $5 \times 10^{-6} \text{m}^3$ |
| mouthpiece choke length | l_c | 48 cm |
| mouthpiece choke radius | a_c | 4.5 mm |

Table 2. Example valve parameters values.

where

$$\begin{aligned} a_{m0} &= LC\alpha^2 + RC\alpha + 1 \\ a_{m1} &= -2(LC\alpha^2 - 1) \\ a_{m2} &= LC\alpha^2 - RC\alpha + 1 \end{aligned}$$

and

$$p_1(z) = \frac{U_2(z)(b_0 + b_1z^{-1}) + p_2(z)(1 + z^{-1})}{1 + z^{-1}}, \quad (14)$$

where

$$b_0 = L\alpha + R \quad \text{and} \quad b_1 = -L\alpha + R,$$

and $\alpha = 2/T$, where T is the sampling period. The corresponding difference equations are given by

$$\begin{aligned} U_2(n) &= [U_1(n) + 2U_1(n-1) + U_1(n-2) - \\ &C\alpha(p_2(n) - p_2(n-2)) - \\ &a_1U_2(n-1) - a_2U_2(n-2)]/a_0, \end{aligned}$$

and

$$\begin{aligned} p_1(n) &= b_0U_2(n) + b_1U_2(n-1) + \\ &p_2(n) + p_2(n-1) - p_1(n-1). \end{aligned}$$

Again, as for the case of discretizing the valve displacement, no pre-warping is required.

5. CONCLUSIONS

In this work, a previously presented trombone model is augmented with a mouthpiece and coupled to a dynamic lip reed model, to explore the resulting parameter space as well as experiment with its synthesis capabilities. Example outputs of the model are shown in Figure 6 and Figure 7, showing the effects of the mouthpiece, with the slide both retracted and extended, in both time and frequency domains, using parameter values in Table 2. To illustrate the effects of the mouthpiece, all other parameters remain the same. It should be noted however, that in actual practice, a change in the instrument parameter will likely require an adapted change in the lip reed as well.

Acknowledgments

The authors would like to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Council for the Arts (CCA) through the Discovery and New Media Initiative programs.

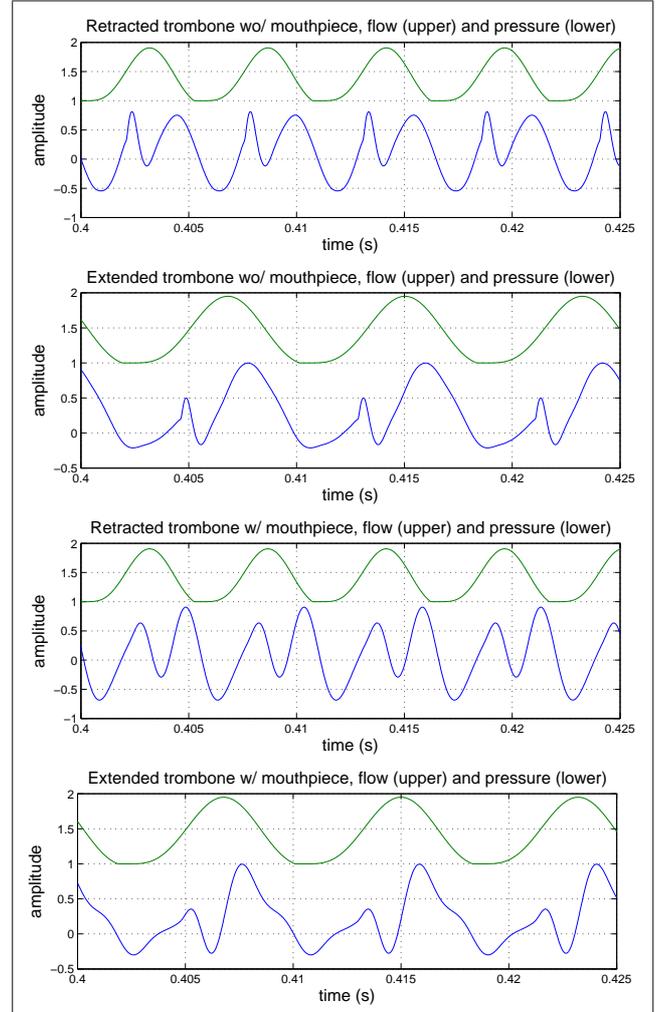


Figure 6. Synthesis examples using parameter values from Table 1 and Table 2, showing flow versus bell output pressure (produced sound) in a note's steady state, with slide in both retracted and extended positions, and with and without a mouthpiece.

6. REFERENCES

- [1] T. Smyth and F. S. Scott, "Trombone synthesis by model and measurement," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. Article ID 151436, p. 13 pages, 2011, doi:10.1155/2011/151436.
- [2] T. Smyth, J. Abel, and J. O. Smith, "A generalized parametric reed model for virtual musical instruments," in *Proceedings of ICMC 2005*. Barcelona, Spain: International Computer Music Conference, September 2005, pp. 347–350.
- [3] T. Smyth and J. Abel, "Estimating waveguide model elements from acoustic tube measurements," *Acta Acustica united with Acustica*, vol. 95, no. 6, pp. 1093–1103, 2009.
- [4] J. O. Smith, "Physical audio signal processing for virtual musical instruments and audio effects," <http://ccrma.stanford.edu/~jos/pasp/>, December 2008, last viewed 8/18/2010.
- [5] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, "Discrete-time modelling of musical instruments," *Report on Progress in Physics*, vol. 69, pp. 1–78, 2006.
- [6] T. Smyth and J. Abel, "Convolutional synthesis of wind instruments," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'07)*, New Paltz, New York, October 2007.
- [7] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. Springer-Verlag, 1995.
- [8] J. Abel, T. Smyth, and J. O. Smith, "A simple, accurate wall loss filter for acoustic tubes," in *DAFX 2003 Proceedings*. London, UK: International Conference on Digital Audio Effects, September 2003, pp. 53–57.
- [9] N. H. Fletcher, "Autonomous vibration of simple pressure-controlled valves in gas flows," *Journal of the Acoustical Society of America*, vol. 93, no. 4, pp. 2172–2180, April 1993.
- [10] T. Smyth, J. Abel, and J. O. Smith, "The feathered clarinet reed," in *Proceedings of the International Conference on Digital Audio Effects (DAFx'04)*, Naples, Italy, October 2004, pp. 95–100.
- [11] W. G. Gardner, "Efficient convolution without input-output delay," *Journal of the Audio Engineering Society*, vol. 43, no. 3, pp. 127–136, March 1995.
- [12] M. van Walstijn, "Discrete-time modelling of brass and reed woodwind instruments with application to musical sound synthesis," Ph.D. dissertation, University of Edinburgh, 2002.
- [13] B. Krach, S. Petrasch, and R. Rabenstein, "Digital sound synthesis of brass instruments by physical modeling," in *Proceedings of the International Conference on Digital Audio Effects (DAFx'04)*, Naples, Italy, October 2004.
- [14] P. Dietz, "Simulation of trumpet tones via physical modeling," Master's thesis, Dept. of Electrical Engineering, Bucknell University, Lewisburg, USA, 1988.

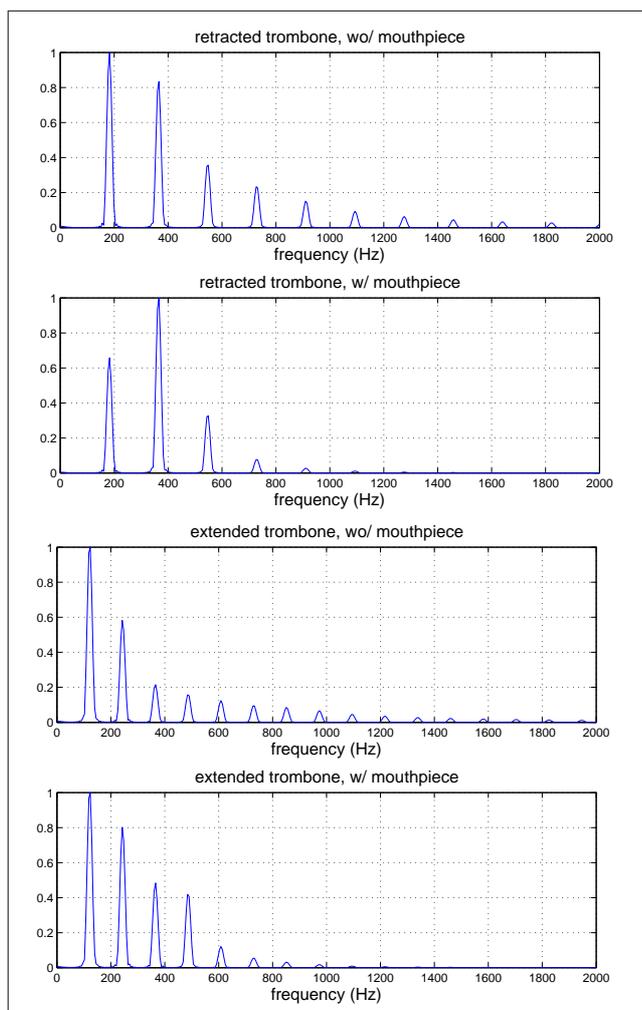


Figure 7. Synthesis examples from Figure 6, showing effects of the mouthpiece in the frequency domain

DISTANCE MAPPING FOR CORPUS-BASED CONCATENATIVE SYNTHESIS

Diemo Schwarz
 UMR STMS
 Ircam–CNRS–UPMC
 schwarz@ircam.fr

ABSTRACT

In the most common approach to corpus-based concatenative synthesis, the unit selection takes places as a content-based similarity match based on a weighted Euclidean distance between the audio descriptors of the database units, and the synthesis target. While the simplicity of this method explains the relative success of CBCS for interactive descriptor-based granular synthesis—especially when combined with a graphical interface—and audio mosaicing, and still allows to express categorical matches, certain desirable constraints can not be formulated, such as disallowing repetition of units, matching a disjunction of descriptor ranges, or asymmetric distances. We therefore propose a new method of mapping the individual signed descriptor distances by a warping function that can express these criteria, while still being amenable to efficient multi-dimensional search indices like the k D-tree, for which we define the preconditions and cases of applicability.

1. INTRODUCTION

Corpus-based concatenative synthesis matches snippets of sounds in a database of sound segments labeled by audio descriptors, to a target also given by descriptors [1]. We call the segments of sound with their description *units*, and the database of units the *corpus*.

In its many incarnations¹ [2], corpus-based concatenative synthesis is most often based on finding the units closest to the target in a multi-dimensional descriptor space, with the most frequently used approach being a weighted Euclidean distance, where the weight w_i is expressing the relative importance of each descriptor $1 \leq i \leq D$ in the match. This means finding the units closest to the current position x in the descriptor space in a geometric sense, on appropriately scaled dimensions by calculating the weighted square Euclidean distance C^t between x and all $1 \leq j \leq N$ units with

$$C^t(j) = \sum_{i=1}^D w_i d_i(j) \quad (1)$$

¹ A constantly updated overview of the many different approaches, applications, and related work concerning CBCS can be found on http://imtr.ircam.fr/imtr/Corpus-Based_Sound_Synthesis_Survey.

based on the per-descriptor distance d_i

$$d_i(j) = \frac{(x(i) - \mu(j, i))^2}{\sigma(i)^2} \quad (2)$$

where μ is the (N, D) matrix of unit descriptor data and σ the standard deviation of each descriptor over the corpus. Either the unit with minimal C^t is selected, or one randomly chosen from the set of units with $C^t < r^2$, when a selection radius r is specified, or, third, one from the set of the k closest units to the target.

This paradigm of similarity for unit selection is simple, easy to understand, especially when coupled with a 2D or 3D representation of and interaction with the data, as in the CATART system² for real-time interactive corpus-based concatenative synthesis [3], and can be efficiently implemented using tree-based search indices [4].

The distance-based paradigm can also be applied to categorical descriptors, e.g. a class-based selection like “*start a new sequence of sounds with a unit in the attack class, then continue with units from the sustain class*” can be expressed by giving integer class-index values as a descriptor, and a sufficiently high weight that pushes any non-matching class far enough away.

The same goes for boolean descriptors that can be used to include or exclude specific units from the selection by giving a binary descriptor and target value and assuring that a non-match will effectively remove the unit from the result set by a high weight.

However, it is difficult to express combined matches of classes (e.g. “*play units from the trumpet or trombone class*”), and certain additional constraints, for instance that all units played in the last p seconds should be different.

Unit Selection by Constraint Resolution Alternative methods based on constraint resolution formulate the unit selection as a CSP (constraint satisfaction problem) [5]. While promising to be more expressive and flexible, the drawback of these methods is that for each new constraint type, code has to be written that integrates it into the constraint solver, and that the CSP itself is NP-complete, such that local search strategies have to be applied to make it computationally tractable, without guarantee to find the best match in a given amount of time. This also means that scalability to larger databases is not assured. Additionally, the CSP approach has only rarely been applied to real-time interactive corpus-based concatenative synthesis [6, 5], and needs a full constraint solver, which is not

² <http://imtr.ircam.fr/imtr/CataRT>

usually integrated in common real-time interactive sound processing systems.

The aim of this article is thus to reformulate some of the abovementioned additional constraints as a distance-based match, in order to integrate them in commonly used real-time synthesis systems [3, 7]. In section 2 we'll introduce a method of mapping the Euclidean per-descriptor signed distances in order to express unit selection constraints such as avoiding repetition, selecting pitch intervals and chords, and introducing asymmetric distances, detailed in section 3. Section 4 will examine the two cases of how these mapped distances can still be used in conjunction with the efficient k D-tree multi-dimensional search index [4], distinguishing the cases of static and dynamic distance mapping functions.

2. DISTANCE MAPPING FUNCTIONS

Our solution to obtaining more flexibility for expressing various selection criteria while still staying in the distance-based paradigm of unit selection, is to map each individual signed descriptor distance calculation through a distance mapping function $f_i(d, \mathcal{P}) : \mathbb{R} \rightarrow \mathbb{R}$ for descriptor i with parameter set \mathcal{P} , before calculating the sum in equation (1), replacing d_i from equation (2) by d'_i :

$$d'_i = \frac{\left(f_i(x(i) - \mu(j, i))\right)^2}{\sigma(i)^2} \quad (3)$$

This allows to pull units in a certain relation to the target closer or push them further away, creating "shortcuts" and, in a sense, folding the descriptor space. Mapped distances can also introduce asymmetry in the distance space, which allows to express lower or upper bounds for selection, as detailed in the following section.

3. APPLICATION EXAMPLES

We will now show several examples of applying distance mapping functions to express unit selection criteria that were not possible using a linear distance alone.

3.1 Range Queries and Note Filtering

For music composition or performance (see for example the applications detailed in [3]), often, the most important criterion is to select precise pitches from a corpus of instrument sounds. Secondary selection criteria could then be, for instance, a certain brilliance, loudness, etc.

With unmapped distances, a given brilliance and loudness target might have been best satisfied with a pitch not matching the given target. Increasing the weight on pitch might also not help here in the general case.

In order to express the priority of the pitch selection, a binary distance mapping function is introduced as

$$f_{\text{range}}(d, r) = \begin{cases} 0 & \text{if } -r \leq d \leq r \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

This mapping, associated with a high weight, will effectively exclude all pitches outside range r around the target

from selection. If the most precise choice within the given range should be favoured, a composite linear and binary distance mapping function can be used:

$$f_{\text{range+}}(d, r) = \begin{cases} d & \text{if } -r \leq d \leq r \\ \infty & \text{otherwise} \end{cases} \quad (5)$$

Here, we use infinity (in practice, a very high real value) for the non-matching case, in order to still be able to adapt the weights of the within-range match of pitch, or actually any descriptor. See figure 1 for a plot of the range distance mapping functions.

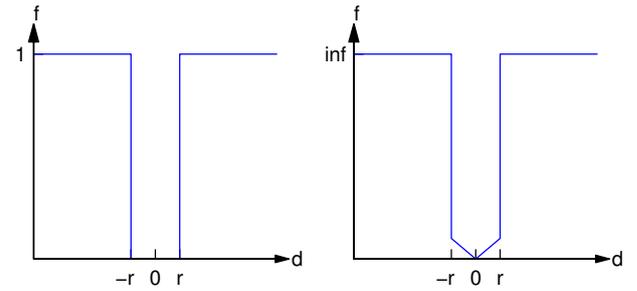


Figure 1. The f_{range} (left) and $f_{\text{range+}}$ (right) distance mapping functions.

This principle can be extended to multiple pitch or category matches by defining a multi-range distance mapping function with notches of size r at distances δ_i as

$$f_{\text{notch}}(d, r, \delta) = \begin{cases} 0 & \text{if } \exists \{i \mid \delta_i - r \leq d \leq \delta_i + r\} \\ 1 & \text{otherwise} \end{cases} \quad (6)$$

Again, in order to favour precise matches in the notch centres, we can use

$$f_{\text{notch+}}(d, r, \delta) = \begin{cases} d - \delta & \text{if } \exists \{i \mid \delta_i - r \leq d \leq \delta_i + r\} \\ \infty & \text{otherwise} \end{cases} \quad (7)$$

Figure 2 shows the notched distance functions applied to an octave match, supposing pitch in half-tones. Any chord should be given here.

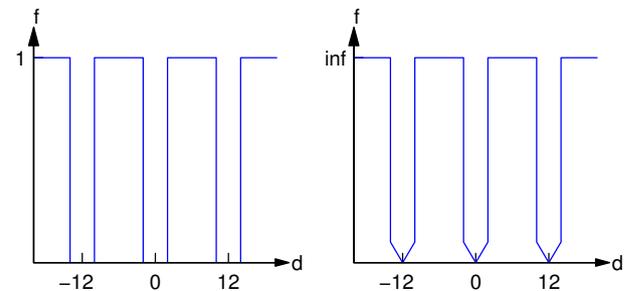


Figure 2. The f_{notch} (left) and $f_{\text{notch+}}$ (right) distance mapping functions.

3.2 Asymmetric Distances

Let's turn now to a case of automated collage or audio mosaicing on a non-uniformly segmented corpus. Here, a given target unit is to be replaced by a unit selected from the corpus. If a continuous output sound is desired, the selected unit should not be shorter than the target unit. It can be longer, however, because a longer database unit can always be cut on playback.

Therefore, we define an asymmetric duration distance mapping function as

$$f_{\text{asym}}(d) = \begin{cases} \infty & \text{if } d < 0 \\ d & \text{otherwise} \end{cases} \quad (8)$$

See figure 1 for a plot of the asymmetric distance mapping function which still prefers a unit matching the exact target duration, to avoid a too large discrepancy between the description of the original unit and the shortened unit.

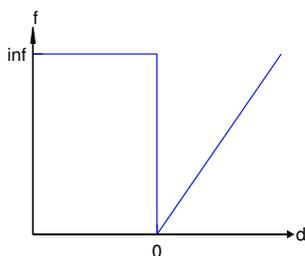


Figure 3. The f_{asym} asymmetric duration distance mapping function.

3.3 Dynamic Taboo Lists

While the above applications were mostly static, and thus could have been implemented by a pre-transformation of descriptor values, we now turn to a dynamic application, where the CSP formulation, or specific programming,³ used to be the only available solutions: the avoidance of repetitions of played units within a certain temporal window p .

We can formulate this within our distance-based selection paradigm by the introduction of a new descriptor *last played time*, initialised to $-\infty$. When a unit is played, its value is set to the current time. We then keep the target value for this descriptor set to the current time, and introduce an asymmetric distance mapping function for it as

$$f_{\text{taboo}}(d, p) = \begin{cases} 0 & \text{if } d > p \\ \infty & \text{otherwise} \end{cases} \quad (9)$$

³ As one reviewer correctly remarked, avoiding repetitions could be realised by a time-ordered queue of last played units. However, this wouldn't provide the three following advantages:

First, expressing non-repetition as linear distance allows to express a soft criterion that can be balanced with the distances expressing other selection criteria by weighting, as mentioned for equation (10).

Second, the computational complexity of maintaining and searching the queue for the k units at each selection would add to the k D-tree search, which can already remove the taboo units without additional complexity.

Third, for modular programming environments such as MAX/MSP, an important design criterion is the generality of the modules. Regarding CBCS, this means having a general distance-calculation module that expresses all needs is preferable to a specifically programmed unit selection module.

This allows to calculate in one integrated step the selection criteria on the static database, and the dynamic and history-dependent constraint of avoiding repetitions.

We could now introduce a less strict taboo by allowing to repeat units earlier, when no better choices are available in the corpus, by giving a minimum window p_{min} and a maximum penalty d_{max} for a distance mapping function like

$$f_{\text{taboo+}}(d, p, p_{\text{min}}, d_{\text{max}}) = \begin{cases} 0 & \text{if } d > p \\ (d - p) \frac{d_{\text{max}}}{p_{\text{min}} - p} & \text{if } d > p_{\text{min}} \\ d_{\text{max}} & \text{otherwise} \end{cases} \quad (10)$$

Figure 4 shows the two taboo distance functions.

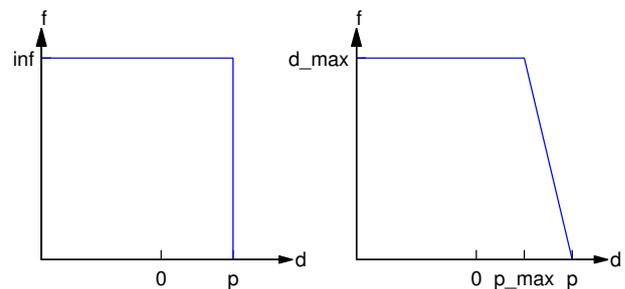


Figure 4. The f_{taboo} (left) and $f_{\text{taboo+}}$ (right) distance mapping functions.

4. INTEGRATION INTO SEARCH INDICES

To perform our distance-based unit selection, and in fact in any application of content-based retrieval, finding the database unit that minimises the target distance C^t is solved most efficiently by a *branch and bound* search algorithm based on the tree-structured index provided by the k D-tree, representing a hierarchical decomposition of the descriptor space by hyperplanes s_n . Details of the indexing and search algorithms are given in previous work [4].

During search, whole subtrees are *pruned*, i.e. discarded from the search, by applying an elimination rule based on the farthest neighbour found so far. This removes a large amount of the distance calculations between feature vectors needed otherwise, resulting in a sublinear time complexity. Several variants of the algorithm are compared by D'haes et al. [8].

We will argue here that our introduction of distance mapping functions does not invalidate the applicability of the k D-tree search index, as soon as certain conditions are met.

First of all, any static distance mapping function that is already applied when building the search index, and whose parameters don't change when searching the tree, is evidently integrated in the index structure.

For the case of dynamically changing distance mapping functions such as in equations (6) or (9), we can define

as necessary condition that the function only *increases* the distance, i.e. iff

$$\forall_{\mathcal{P}} \forall_d f(d, \mathcal{P}) \geq d \quad (11)$$

This works, because the elimination rule states that any subtree that lies on the opposite side of its splitting hyper-plane s_n when seen from the target point t , and where the distance from t to s_n is greater than the distance d_{kmax} of the farthest neighbour found so far, does not need to be searched. By increasing the mapped distance, we push points further away from the query point, so that the elimination rule still holds. (If we'd decrease the distance and pull points closer, the elimination rule might have already pruned these points from the search, although they might now be part of the nearest neighbours.)

This condition is met for equations (5), (7), and (8), but equations (4) and (6) do not qualify. There is, however, a workaround to make any distance mapping function amenable to the k D-tree search index, which is to start from an all zero distance $d_i = 0$ while building the index, so that any mapped distance will be greater than the original distance. This actually means that descriptor i was excluded from building the search tree, and that the query will use the index on all but descriptor i , and then using the mapped d'_i to sift the result list.

4.1 Influence on the computational complexity

As to the question how the distance mapping influences on the performance of search in the k D-tree, our tests showed only a slight deterioration, with an increase in the number of vector-to-vector comparisons relative to the number of distance mapped descriptors and the parameters of the distance mapping functions.

A detailed evaluation that measures the number of comparisons for a certain number of test cases and situates them between the baseline of brute force search and the optimal k D-tree without distance mapping is planned for future work.

5. IMPLEMENTATION

The distance mapping algorithm described here is implemented as C-libraries and within the `mm.mahalanobis` and `mm.knn` externals within the free MAX/MSP extension library FTM&Co [9] at <http://ftm.ircam.fr>, providing real-time optimised data structures, and thus available in the CATART real-time interactive CBCS system [3], and within the MUBU externals [7] at <http://imtr.ircam.fr>.

In FTM&Co, the distance mapping functions are conveniently and flexibly represented as break-point function (BPF) objects, which also allows to edit them manually in the graphic externals `ftm.editor` and `IMTR Editor` [7].

6. CONCLUSIONS

We have seen in this article a simple and efficient way to formulate many additional criteria, that are desirable for

musical use of interactive corpus-based concatenative synthesis. These criteria go beyond multi-dimensional proximity and were not possible to be expressed in the habitual framework of unit selection based on Euclidean distances.

Our formulation of distance mapping by a warping function integrates these criteria in the distance calculation while still keeping efficient selection methods based on k D-trees and branch and bound search applicable, with only little loss of efficiency.

The functional formulation of the constraints means that different distance mapping functions could be interpolated to smoothly crossfade from one solution space to another.

Acknowledgments

The work presented here is partially funded by the *Agence Nationale de la Recherche* within the project *Topophonie*, ANR-09-CORD-022, <http://topophonie.fr>.

7. REFERENCES

- [1] D. Schwarz, "Corpus-based concatenative synthesis," *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 92–104, Mar. 2007, special Section: Signal Processing for Sound Synthesis.
- [2] —, "Concatenative sound synthesis: The early years," *Journal of New Music Research*, vol. 35, no. 1, Mar. 2006, special Issue on Audio Mosaicing.
- [3] D. Schwarz, R. Cahen, and S. Britton, "Principles and applications of interactive corpus-based concatenative synthesis," in *Journées d'Informatique Musicale (JIM)*, GMEA, Albi, France, Mar. 2008.
- [4] D. Schwarz, N. Schnell, and S. Gulluni, "Scalability in content-based navigation of sound databases," in *Proc. ICMC*, Montreal, QC, Canada, 2009.
- [5] J.-J. Aucouturier and F. Pachet, "Jamming With Plunderphonics: Interactive Concatenative Synthesis Of Music," *Journal of New Music Research*, vol. 35, no. 1, Mar. 2006, special Issue on Audio Mosaicing.
- [6] J.-J. Aucouturier, F. Pachet, and P. Hanappe, "From sound sampling to song sampling," in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, Barcelona, Spain, Oct. 2004, pp. 1–8.
- [7] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, and R. Borghesi, "MuBu & friends – assembling tools for content based real-time interactive audio processing in Max/MSP," in *Proc. ICMC*, Montreal, 2009.
- [8] W. D'haes, D. van Dyck, and X. Rodet, "PCA-based branch and bound search algorithms for computing K nearest neighbors," *Pattern Recognition Letters*, vol. 24, no. 9–10, pp. 1437–1451, 2003.
- [9] N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Müller, "FTM—Complex Data Structures for Max," in *Proc. ICMC*, Barcelona, 2005.

EMOTIONAL RESPONSE TO MAJOR MODE MUSICAL PIECES: SCORE-DEPENDENT PERCEPTUAL AND ACOUSTIC ANALYSIS

Sergio Canazza, Giovanni De Poli, Antonio Rodà

Dep. Information Engineering

University of Padova

{sergio.canazza, giovanni.depoli, antonio.roda}@unipd.it

ABSTRACT

In the Expressive Information Processing field, some studies investigated the relation between music and emotions, proving that it is possible to correlate the listeners' main appraisal categories and the acoustic parameters which better characterize expressive intentions, defining score-independent models of expressiveness. Other researches take to account that part of the emotional response to music results from the cognitive processing of musical structures (key, modalities, rhythm), which are known to be expressive in the context of the Western musical system. Almost all these studies investigate emotional responses to music by using verbal labels, that is potentially problematic since it can encourage participants to simplify what they actually experiencing. Recently, some authors proposed an experimental method that makes no use of verbal labels. By means of the multidimensional scaling method (MDS), a two-dimensional space was found to provide a good fit of the data, with arousal and emotional valence as the primary dimensions. In order to emphasize other latent dimensions, a perceptual experiment and a comprehensive acoustic analysis was carried out by using a set of musical pieces all in major mode. Results show that participants tend to organize the stimuli according to three clusters, related to musical tempo and to timbral aspects such as the spectral energy distribution.

1. INTRODUCTION

Information about music performance, structured as metadata, could further the development of new application such as automatic expressive performance or active listening, and offer a contribution to improve systems in the context of content-based retrieval, entertainment, and music education. Moreover, the study of music is not limited to the artistic field. Indeed, the power of music to arouse in the listener a rich set of sensations, such as images, feelings, or emotions, can have many applications. In the information technology field, a musical signal can contribute to the multimodal/multisensory interaction, communicating events and processes, providing the user with information through sonification, or giving auditory warnings. In this

sense, sound design requires great attention and a deep understanding of the influence of musical parameters on the user's experience.

The communication of expressive content by music can be studied at three different levels, considering: (1) the expressive intentions of the performer, (2) the listener's perceptual experience, and (3) the composers message.

(1) Most studies on the performance expressiveness aim at understanding the systematic presence of deviations from the musical notation as a communication means between musician and listener (see, e.g. [1, 2]). Deviations introduced by technical constraints (such as fingering) or by imperfect performer skill, are not normally considered part of expression communication and thus are often filtered out as noise. The analysis of these systematic deviations has led to the formulation of several models (e.g., [3, 4, 5, 6]) which aim to describe where, how and why a performer modifies, sometimes unconsciously, the score notation. It should be noticed that, although deviations are only the external surface of something deeper and often not directly accessible, they are quite easily measurable, and thus widely used to develop computational models in scientific research and generative models for musical applications.

(2) These studies investigated the relation between music and emotions, showing a sort of isomorphism between musical expression and listeners' affective responses. Perceptual studies proved how, generally speaking, it is possible to correlate the listeners' main appraisal categories and the acoustic parameters which better characterize expressive intentions ([7, 8] for reviews).

(3) This research takes in account that part of the emotional response to music results from the cognitive processing of musical structures (key, modalities, rhythm), which are known to be expressive in the context of the Western musical system. For example, musical features such as modulation, grace notes, and harmonic progressions, are often associated with emotional responses in the verbal reports of participants [9]. Peretz, Gagnon, and Bouchard [10] demonstrated that rhythm and modality (major vs. minor) contribute to happiness or sadness. These studies are developed in [11, 12]. Generally, they analyze the elements of the musical structure and the musical phrasing that are critical for a correct interpretation of composers message.

Some studies investigated the relation between music and emotions, proving that is possible to correlate the listeners' main appraisal categories and the acoustic parameters

which better characterize expressive intentions, defining score-independent models of expressiveness [8].

[13] address the question of whether expressive information can be communicated (and recognized) by means of features which are not strictly related to the score. Thus, relevant musical attributes for differentiating expressions (such as articulation) can be replaced by more physical features (e.g. attack time). Professional performers of violin and flute were asked to play musical performances in order to convey different expressive intentions, described by the adjectives that lie on the affective space (happy, sad, angry and calm), and on the sensorial space (light, heavy, soft and hard). With the aid of machine learning techniques we found the audio features that are most relevant for the recognition of different expressive intentions. Using these features as coordinates, we could place the expressions on a feature space and obtain an objective measure of physical similarity. In particular, we extracted and selected a set of audio features from a set of expressive performances played by professional musicians on violin and flute. These features were tested and confirmed by the leave-one-out cross validation, and they can be grouped according to local audio features (using non overlapping frames of 46ms length), and event features (using sliding windows with 4s duration and 3.5s overlap).

A common characteristic of almost all these studies was to investigate emotional responses to music by using verbal labels. The use of verbal labels is potentially problematic since it can encourage participants to simplify what they actually experience [14] and the subjects responses may be conditioned by the different semantic nuances of the same word. Recently, Bigand [12] investigates the emotion conveyed by musical pieces, carrying out some perceptual experiments without making use of verbal labels. Musically trained and untrained listeners were asked to listen 27 different musical excerpts and to freely group those that conveyed similar subjective emotions. By means of the multidimensional scaling method (MDS), a two dimensional space was found to provide a good fit of the data, with arousal and emotional valence as the primary dimensions (Fig. 1). In particular, the excerpts resulted grouped in four clusters, characterized by i) high arousal and high valence (HAHV), ii) low arousal and high valence (LAHV), iii) high arousal and low valence (HALV) and iv) low arousal and low valence (LALV).

Though in his paper Bigand refers to a hypothetical third axis, this aspect is not discussed in detail. Since much of the variance in the results of the Bigand's experiment is due to the mode (major or minor) of the musical pieces, we planned an experiment to investigate other secondary aspects of the relation between music and emotions. Musically trained and untrained listeners were asked to listen the 23 different musical excerpts, all in major mode, and to group those that conveyed similar subjective emotions (see Sec. 2.1). The statistical analysis of the responses (see Sec. 2.2), showed that the listeners organized the musical stimuli in three clusters. In order to investigate the nature of these associations (both the four Bigand's clusters and the three clusters of the new experiment), we carried

out a detailed acoustic analysis of the musical stimuli (see Sec. 3). This analysis allowed us to relate the subjects' answers with the music features and to identify relations among the musical and the affective domains, in order to emphasize the existence of secondary factors that characterize the perception of emotion in music. To this end, we have selected musical pieces only in a major mode, a parameter largely related to the axis of the affective valence.

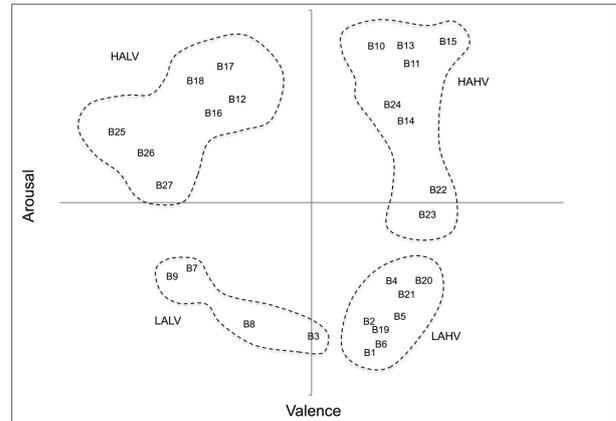


Figure 1. The 27 excerpts of the experiment in Bigand [12], mapped on a two-dimensional space. Dashed lines represent the four affective clusters: high arousal and high valence (HAHV); high arousal and low valence (HALV); low arousal and high valence (LAHV); low arousal and low valence (LALV). Figure adapted from [12].

2. PERCEPTUAL EXPERIMENT IN MAJOR MODE

An experiment has been carried out in order to emphasize the existence of secondary factors that characterize the perception of emotion in music. In the previous section, it has been noted that much of the variance in the results of the Bigand's experiment is due to the modality (major or minor) of the musical pieces. To reduce the effect of this component, we decided to follow the same experimental method, but applying it to musical pieces only in major mode.

2.1 Materials and method

For the experiment, 23 musical excerpts have been chosen as follows: 11 pieces are taken from Bigand in [12], selecting those in a major mode, that are numbered 1, 4, 5, 6, 11, 13, 14, 15, 20, 21, and 23; 12 other pieces were chosen by the Western music repertoire, from XVII to XX century. In particular, the added excerpts are all in a major mode and have been chosen to be representative of various compositional styles. They correspond either to the beginning of a musical movement, or to the beginning of a musical theme or idea. The duration of the excerpts is on average of 30s.

The procedure follows the one already used in [12]. The experiment was conducted using an especially developed software interface. Participants were presented with a visual pattern of 23 loudspeakers, representing the 23 ex-

cerpts in a random order. They were required first to listen to all of these excerpts and to focus their attention on the emotional experience of the listening. They were then asked to look for excerpts that induced a similar emotional experience and to drag the corresponding icons in order to group these excerpts. They were allowed to listen to the excerpts as many times as they wished, and to regroup as many excerpts as they wished. The experiment were performed by a total of 40 participants. Of these, 20 did not have any musical experience and are referred to as non-musicians; 20 were music students for at least five years and are referred to as musicians. The duration of the test was about 30 minutes.

2.2 Results

Participants have formed an arbitrary number N of groups. Each group G_k contains the stimuli that the a subject thinks similar (i.e., that induces a similar emotive experience). The dissimilarity matrix A is defined by counting how many times two excerpts i and j are not included in the same group:

$$A[i, j] = \begin{cases} A[i, j] + 1 & \text{if } i \in G_k \wedge j \notin G_k \\ A[i, j] & \text{otherwise} \end{cases} \quad (1)$$

$\forall i, j = 1, \dots, 23$ and $\forall k = 1, \dots, N$.

The dissimilarity matrix was analyzed by means of a Multi-Dimensional Scaling (MDS) method. The location of the 23 excerpts along the two principal dimensions is represented in Figure 2. The excerpts that are close in this space are those evaluated to be more similar by the subjects. The musical pieces coming from the Bigand's experiment have maintained their original numeration (with a 'B' before the number); the other pieces, instead, have been labeled without any letter. Moreover, the MDS solution was compared with a cluster analysis performed on the dissimilarity matrix. The three main clusters are marked in Figure 2 by means of dotted lines.

In Bigand's experiment, the selected excerpts were grouped in two clusters (see Fig. 1): LAHV and HAHV. It can be noted that the excerpts B1, B4, B5, B6, and B21 are still grouped together in one cluster, named Low-Arousal (LA) cluster. Differently, the other Bigand's excerpts are divided into two clusters, named High-Arousal clusters (HA1 and HA2). In short, the first dimension of the MDS is related with the arousal dimension of Figure 1, whereas the second dimension does not seem connected to any of the axes identified by Bigand in his study.

3. ACOUSTIC ANALYSIS OF PREVIOUS EXPERIMENT

3.1 Feature extraction

In order to relate subjects' answers with musical features, we carried out a detailed acoustic analysis of the musical stimuli both of Bigand's and present experiment. A set of acoustic features were calculated for each excerpt. The set was chosen among those features that in previous listening experiments [15] were found to be important for discriminating different emotions and were also used to classify

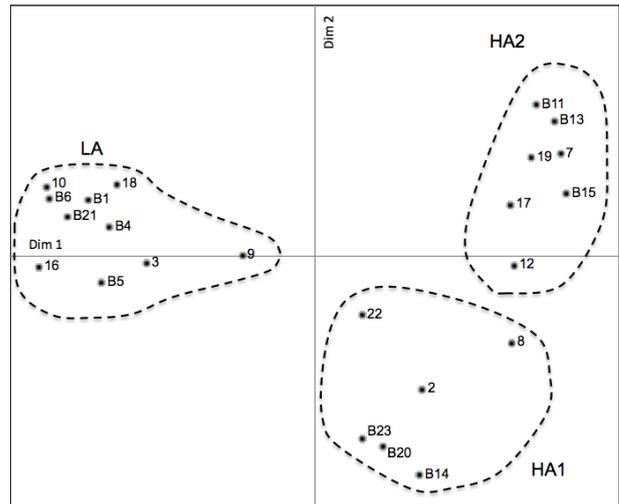


Figure 2. MDS analysis on experiment data. Dashed lines represent the outcome of the cluster analysis.

the style [16] and the expressive content in musical performances [17] and [13]. We computed the features using non-overlapping frames (of 46-ms length), and then we considered their mean value within sliding windows (with 4-s duration and 3.5-s overlap). The window size allows to include a reasonable number of events, and it roughly corresponds to the size of the echoic memory. In total, we collected a set of 13 audio features. See Tab. 1 for a formal description of the features. The features are: a) *ZeroCross* consists in counting the number of times the audio signal changes sign. It can be considered as a simple indicator of noisiness; b) *RMS* takes into account the global energy of the signal, computed as the root average of the square of the amplitude (root-mean-square); c) *Centroid* is the first moment of the spectral magnitude. It is related with the impression of "brightness" of a sound [18], because a high centroid value means that the sound energy is concentrated at the higher frequencies; d) *Brightness* measures the amount of energy above the frequency of 1000 Hz. The result is expressed as a number between 0 and 1; e) *Spectral ratios (SRs)* over different frequency bands of the spectrum are other useful indications of the spectrum shape. The spectrum is divided in three regions: below 534 Hz (*SRl*), from 534 to 1805 Hz (*SRm*), and above 1805 Hz (*SRh*); f) *Rolloff* is the frequency such that the 85% of the total energy is contained below that frequency. It is related to the "brightness" of the sound; g) *Roughness* is calculated starting from the results of Plomp and Levelt [19], that proposed an estimation of the dissonance degree between two sinusoids, depending on the ratio of their frequency. The total roughness for a complex sound can be calculated by computing the peaks of the spectrum, and taking the average of all the dissonance between all possible pairs of peaks [20]; h) *SpectralFlux* is the distance between the spectrum of each successive frame; i) *LowEnergy* is the percentage of frames showing less-than-average energy. It is an assessment of the temporal distribution of energy, in order to see if it remains constant throughout the signal, or if some frames are more contrastive than others;

l) *Tempo* is the musical velocity of the performance. Since many of the 27 excerpts have a complex polyphonic structure, it is not easy to have a good estimation of this feature using an automatic routine. Then, the *Tempo* of each excerpt was estimated by means of the manual annotations of an expert as the average of the piece; m) *Mode* is a basic aspect of the musical structure. In Western tonal music there are two modes, named major and minor mode. Also in this case, we used the annotations of an expert who analysed the musical sheets.

Starting from the calculated features, we selected the subset of features related both to the four clusters of the Bigand's experiment and the three clusters of the experiment in major mode. The feature selection procedure consists in finding the audio features that give the highest classification ratings. A wrapper approach based on sequential feature selection (SFS) [21] is applied with reference to a linear classifier. The feature selection procedure was applied twice. The first time we selected the set of features that classify the 23 excerpts, with a minimum error rate, following the classes specified by the four clusters HAHV, LAHV, HALV, and LALV. The SFS process selected the following four features, in order of selection: *Tempo*, *Mode*, *Centroid*, and *RMS*. The minimum error rate is 18%. Then, we selected the set of features that classify, with a minimum error rate, the 23 excerpts following the classes specified by clusters LA, HA1, and HA2. The SFS process selected the following three features, in order of selection: *Tempo*, *Rolloff*, *Zerocross*. The minimum error rate is 23%.

3.2 Results

Tab. 2 shows the mean values of the four features selected for the Bigand's clusters, calculated for each excerpt of his experiment. The excerpts belonging to the clusters with high arousal (i.e. HAHV and HALV) are characterized, with a few exceptions, by a high value of *Tempo*. In particular, the mean value among the excerpts of HALV is 127bpm, HAHV is 100bpm, LAHV is 63bpm, and LALV is 47bpm ($F = 11.2$ on 3 and 23 *df*, $p < 0.001$), where bpm stands for beats-per-minute. The excerpts belonging to the clusters with low valence (i.e. HALV and LALV) are characterized by a minor mode; all excerpts except number 25 and 26 that are atonal pieces. On the contrary, all the excerpts but one of the HAHV cluster have a major mode and the excerpt 24, taken from a Stravinsky's composition, has an uncertain tonality based on two superposed major chords. The excerpts of the LAHV cluster are mostly characterized by a major mode. A Chi-squared analysis showed that modality is significantly related with the valence factor ($\chi^2 = 14.9$, *df* = 2, $p < 0.001$). In regard to the other two selected features, a high *Centroid* value characterizes the clusters with high valence (the average value is 1588Hz for HAHV, 1573Hz for LAHV, 1426Hz for HALV, and 1348Hz for LALV), whereas a high *RMS* value distinguishes the clusters with high arousal from the others (the average value is 0.094 for HAHV, 0.080 for LAHV, 0.098 for HALV, and 0.057 for LALV). However, for both these features, the differences are not statistically

| cluster | excerpt | Tempo [bpm] | Mode | Centroid [Hz] | RMS |
|---------|---------|-------------|--------------|---------------|-------|
| HAHV | 10 | 109 | major | 1643 | 0.075 |
| | 11 | 53 | major | 2684 | 0.091 |
| | 13 | 103 | major | 1737 | 0.080 |
| | 14 | 102 | major | 1141 | 0.151 |
| | 15 | 145 | major | 1473 | 0.067 |
| | 22 | 103 | major | 1376 | 0.060 |
| | 23 | 59 | major | 1047 | 0.053 |
| | 24 | 123 | undetermined | 1603 | 0.174 |
| LAHV | 1 | 61 | major | 1694 | 0.086 |
| | 2 | 77 | minor | 2322 | 0.067 |
| | 4 | 53 | major | 1288 | 0.089 |
| | 5 | 53 | major | 1075 | 0.108 |
| | 6 | 50 | major | 1078 | 0.086 |
| | 19 | 65 | minor | 2091 | 0.105 |
| | 20 | 65 | minor | 1345 | 0.051 |
| | 21 | 76 | major | 1691 | 0.045 |
| HALV | 12 | 157 | minor | 1097 | 0.061 |
| | 16 | 142 | minor | 1074 | 0.220 |
| | 17 | 149 | minor | 1844 | 0.045 |
| | 18 | 144 | minor | 1760 | 0.174 |
| | 25 | 151 | undetermined | 1725 | 0.056 |
| | 26 | 88 | undetermined | 1487 | 0.054 |
| | 27 | 58 | minor | 997 | 0.079 |
| LALV | 3 | 40 | minor | 1106 | 0.018 |
| | 7 | 48 | minor | 1034 | 0.088 |
| | 8 | 50 | minor | 1615 | 0.073 |
| | 9 | 51 | minor | 1634 | 0.048 |

Table 2. Acoustic features related to the clusters resulting from the Bigand's experiment.

significant ($F < 0.9$ on 3 and 23 *df*, $p > 0.05$).

Tab. 3 shows the mean values of the three features selected for the experiment in major tonality, calculated for each excerpt. The excerpts belonging to the cluster with low arousal (LA) are characterized, with a few exceptions, by a low value of *Tempo*. On the contrary, the clusters with high arousal (HA1 and HA2) are characterized by a high value of *Tempo*. In particular, the mean value among the excerpts is 59bpm for LA, 93bpm for HA1, and 97bpm, for HA2. The ANOVA test shows that these differences are statistically significant ($F = 8.3$ on 2 and 20 *df*, $p < 0.01$). On the contrary, no significant difference exists between the *Tempo* of HA1 and HA2 clusters. It means that *Tempo* feature is related to the dimension 1 (LA versus HA1 and HA2), but it is not related to the dimension 2 (HA1 versus HA2).

As concern the *Rolloff* feature, significant difference exists among the mean values of the three clusters ($F = 9.8$ on 2 and 20 *df*, $p < 0.01$): 1923Hz for LA, 2225Hz for HA1, and 3828Hz for HA2. Considering the clusters two by two, the difference between LA and HA1 is not significant, while it is significant between HA1 and HA2. This result means that the dimension 2 can be related to *Rolloff* feature.

Finally, the mean values of *Zerocross* feature are 585 for cluster LA, 732 for HA1, and 1150 for HA2 ($F = 11.5$ on 2 and 20 *df*, $p < 0.01$). Similar to *Rolloff*, the difference is not significant between LA and HA1, while it is significant between HA1 and HA2. Tables 4 and 5 summarize qualitatively the results of the two acoustic analyses.

| | |
|---------------------|--|
| <i>RMS</i> | $\sqrt{\frac{1}{n} \sum_{n=1}^N x(f, n)^2}, f = 1, \dots, M$ |
| <i>Zerocross</i> | $\sum_{n=1}^{N-1} \mathbf{I}\{\text{sign}(x(f, n)) \neq \text{sign}(x(f, n+1))\}, f = 1, \dots, M$ |
| <i>Centroid</i> | $\frac{\sum_{k=1}^N F(f, k)X(f, k)}{\sum_{k=1}^N X(f, k)}, f = 1, \dots, M$ |
| <i>Brightness</i> | $\frac{\sum_{k=k_{1000}+1}^N X(f, k)}{\sum_{k=1}^N X(f, k)}, f = 1, \dots, M$ |
| <i>SRI</i> | $\frac{\sum_{k=1}^{k_{534}} X(f, k)}{\sum_{k=1}^N X(f, k)}, f = 1, \dots, M$ |
| <i>SRm</i> | $\frac{\sum_{k=k_{534}+1}^{k_{1805}} X(f, k)}{\sum_{k=1}^N X(f, k)}, f = 1, \dots, M$ |
| <i>SRh</i> | $\frac{\sum_{k=k_{1805}+1}^N X(f, k)}{\sum_{k=1}^N X(f, k)}, f = 1, \dots, M$ |
| <i>Rolloff</i> | $f(k_{85}), \text{ where } k_{85} = \min(k_0) : \frac{\sum_{k=1}^{k_0} X(f, k)}{\sum_{k=1}^N X(f, k)} > 0.85, f = 1, \dots, M$ |
| <i>Spectralflux</i> | $\sqrt{\sum_{k=1}^N [X(f+1, k) - X(f, k)]^2}, f = 1, \dots, M-1$ |
| <i>Lowenergy</i> | $\frac{\sum_{f=1}^M \mathbf{I}\{\text{rms}(x(f)) < \text{rms}(x)\}}{M}$ |

Table 1. List of the acoustic features. The signal x is blocked in M frames of N samples. Let be $x(f, n)$ the signal amplitude of the sample n at the frame f ; $X(f, k)$ the spectrum magnitude of the bin k at the frame f and $F(f, k)$ the center frequency of that bin; k_{f_t} the bin corresponding to the frequency f_t ; $\mathbf{I}\{A\}$ the indicator function equal to 1 if A is true and 0 otherwise; $\text{sign}(x)$ a function equal to 1 if $x \geq 1$ and 0 otherwise; $\text{rms}(x(f))$ the *RMS* value over the frame f and $\text{rms}(x)$ the *RMS* value over the entire signal x .

| cluster | excerpt | Tempo [bpm] | Rolloff [Hz] | Zerocross |
|---------|---------|-------------|--------------|-----------|
| LA | B1 | 61 | 2372 | 521 |
| | 3 | 52 | 1868 | 516 |
| | B4 | 53 | 1747 | 938 |
| | B5 | 53 | 1106 | 370 |
| | B6 | 50 | 1028 | 443 |
| | 9 | 78 | 2289 | 576 |
| | 10 | 54 | 2707 | 468 |
| | 16 | 56 | 1069 | 449 |
| | 18 | 60 | 2560 | 713 |
| | B21 | 76 | 2487 | 852 |
| HA1 | 2 | 120 | 3210 | 784 |
| | 8 | 98 | 3234 | 1044 |
| | B14 | 102 | 1799 | 655 |
| | B20 | 103 | 1582 | 735 |
| | 22 | 76 | 1741 | 498 |
| | B23 | 59 | 1786 | 675 |
| HA2 | 7 | 84 | 2714 | 817 |
| | B11 | 53 | 4972 | 1650 |
| | 12 | 104 | 4367 | 1083 |
| | B13 | 103 | 3177 | 972 |
| | B15 | 145 | 2495 | 827 |
| | 17 | 72 | 3229 | 1121 |
| | 19 | 116 | 5841 | 1579 |

Table 3. Acoustic features related to the clusters resulting from the experiment in key major.

4. CONCLUSIONS

An experiment has been carried out in order to emphasize the existence of secondary factors that characterize the perception of emotion in music. To this end, we have selected musical pieces only in a major mode, a parameter largely related to the axis of the affective valence. The results show that participants tend to organize the stimuli according to three clusters. The meaning of these clusters has been investigated by means of an in-depth acous-

| Cluster | Mode | Tempo |
|---------|------|-------|
| LALV | - | - |
| LAHV | + | - |
| HALV | - | + |
| HAHV | + | + |

Table 4. Relation among clusters and selected features in the Bigand's experiment.

| Cluster | Tempo | Rolloff | Zerocross |
|---------|-------|---------|-----------|
| LA | - | - | - |
| HA1 | + | - | - |
| HA2 | + | + | + |

Table 5. Relation among clusters and selected features in the major mode experiment.

tic analysis, that revealed a significative correlation between some musical/acoustic features and the subject's responses: *Tempo*, *Rolloff* (a feature related to the brightness of the sound), and *Zerocross* (related to the noisiness of the sound) are the parameters selected to be the most representative of the found clusters. The analysis of the acoustic features on one hand confirms the results of previous research [22][12], i.e. the main parameters that characterize the affective responses to music are *Tempo* and *Mode*. On the other hand, it gave rise to other aspects that affect the emotional perception of music, such as timbral elements related to the spectral energy distribution.

Interesting similarities are further recognizable with the results of score-independent studies (see [8, 13]) which explored the relation between timbral parameters and musi-

cal expression, suggesting the existence of a common level of representation for music expressiveness both in score-dependent and score-independent contexts.

5. REFERENCES

- [1] N. P. McAngus Todd, "The kinematics of musical expression," *Journal of the Acoustical Society of America*, vol. 97, pp. 1940–1949, 1995.
- [2] C. Palmer, "Music performance," *Annual Review of Psychology*, vol. 48, pp. 115–138, 1997.
- [3] B. H. Repp, "Diversity and commonality in music performance: an analysis of timing microstructure in Schumann's "Träumerei"," *Journal of The Acoustical Society of America*, vol. 92, pp. 2546–2568, 1988.
- [4] —, "Expressive timing in Schumann's "Träumerei:" an analysis of performances by graduate student pianists," *Journal of The Acoustical Society of America*, vol. 98, pp. 2413–2427, 1995.
- [5] —, "A microcosm of musical expression: I. quantitative analysis of pianists' timing in the initial measures of chopin's etude in E major," *Journal of The Acoustical Society of America*, vol. 104, pp. 1085–1100, 1998.
- [6] —, "A microcosm of musical expression: II. quantitative analysis of pianists' dynamics in the initial measures of chopin's etude in E major," *Journal of The Acoustical Society of America*, vol. 105, pp. 1972–1988, 1999.
- [7] P. N. Juslin and J. A. Sloboda, *Music and emotion. Theory and research*. Oxford University Press, 2001.
- [8] S. Canazza, G. De Poli, A. Rodà, and A. Vidolin, "An abstract control space for communication of sensory expressive intentions in music performance," *Journal of New Music Research*, vol. 32, no. 3, pp. 281–294, 2003.
- [9] J. A. Sloboda, "Music structure and emotional response: Some empirical findings," *Psychology of music*, vol. 19, pp. 110–120, 1991.
- [10] I. Peretz, L. Gagnon, and B. Bouchard, "Music and emotion: Perceptual determinants, immediacy and isolation after brain damage," *Cognition*, vol. 68, pp. 111–141, 1998.
- [11] A. Gabrielsson and E. Lindstrom, "The influence of the musical structure on emotional expression," in *Music and emotion. Theory and research*, P. N. Juslin and J. A. Sloboda, Eds. Oxford University Press, 2001, pp. 223–249.
- [12] E. Bigand, S. Vieillard, F. Madurell, J. Marozeau, and A. Dacquet, "Multidimensional scaling of emotional responses to music: The effect of musical expertise and of the duration of the excerpts," *Cognition and Emotion*, vol. 19, no. 8, pp. 1113–1139, 2005.
- [13] L. Mion and G. De Poli, "Score-independent audio features for description of music expression," *IEEE Trans. Speech, Audio, and Language Process*, vol. 16, no. 2, pp. 458–466, 2008.
- [14] K. Scherer, "Affect bursts," in *Emotions: Essays on emotion theory*, S. van Goozen, N. E. van de Poll, and J. A. Sergeant, Eds. Erlbaum, 1994, pp. 161–196.
- [15] P. N. Juslin, "Communicating emotion in music performance: A review and a theoretical framework," in *Music and Emotion: Theory and Research*, P. N. Juslin and J. A. Sloboda, Eds. New York: Oxford Univ. Press, 2001, pp. 305–333.
- [16] R. Dannenberg, B. Thorn, and D. Watson, "A machine learning approach to musical style recognition," in *Proc. Int. Comput. Music Conf. (ICMC97)*, San Francisco, CA, 1997, pp. 344–347.
- [17] A. Friberg, E. Schoonderwaldt, P. Juslin, and R. Bresin, "Automatic real-time extraction of musical expression," in *Proc. Int. Comput. Music Conf. (ICMC02)*, Goteborg, Sweden, 2002, pp. 365–367.
- [18] E. Schubert, J. Wolfe, and A. Tarnopolsky, "Spectral centroid and timbre in complex, multiple instrumental textures," in *Proceedings of the International Conference on Music Perception and Cognition*, N. W. University, Ed., Illinois, 2004.
- [19] R. Plomp and W. J. M. Levelt, "Tonal consonance and critical bandwidth," *Journal of the Acoustical Society of America*, vol. 38, no. 4, pp. 548–560, 1965.
- [20] W. A. Sethares, *Tuning, Timbre, Spectrum, Scale*. Springer-Verlag, 1998.
- [21] A. Whitney, "A direct method of nonparametric measurement selection," *IEEE Trans. Comput.*, vol. 20, no. 9, pp. 1100–1103, 1971.
- [22] L. Gagnon and I. Peretz, "Mode and tempo relative contributions to happy–sad judgements in equitone melodies," *Cognition and Emotion*, vol. 17, no. 1, pp. 25–40, 2003.

EXPLAINING MUSICAL EXPRESSION AS A MIXTURE OF BASIS FUNCTIONS

Maarten Grachten

Department of Computational Perception
Johannes Kepler University, Linz, Austria
<http://www.cp.jku.at/people/grachten>

Gerhard Widmer

Department of Computational Perception
Johannes Kepler University, Linz, Austria
<http://www.cp.jku.at/people/widmer>

Austrian Research Institute for
Artificial Intelligence, Vienna, Austria

ABSTRACT

The quest for understanding how pianists interpret notated music to turn it into a lively musical experience, has led to numerous models of musical expression. One of the major dimensions of musical expression is loudness. Several models exist that explain loudness variations over the course of a performance, in terms of for example phrase structure, or musical accent. Often however, especially in piano music from the romantic period, performance directives are written explicitly in the score to guide performers. It is to be expected that such directives can explain a large part of the loudness variations. In this paper, we present a method to model the influence of notated loudness directives on loudness in piano performances, based on least squares fitting of a set of basis functions. We demonstrate that the linear basis model approach is general enough to allow for incorporating arbitrary musical features. In particular, we show that by including notated pitch in addition to loudness directives, the model also accounts for loudness effects in relation to voice-leading.

1. INTRODUCTION AND RELATED WORK

When a musician performs a piece of notated music, the performed music typically shows large variations in tempo, loudness, articulation, and, depending on the nature of the instrument, other dimensions such as timbre and note attack. It is generally acknowledged that one of the primary goals of such variations is to convey an expressive interpretation of the music to the listener. This interpretation may contain emotional elements (e.g. to play a piece ‘solemnly’), and also elements that convey musical structure (e.g. to highlight a particular melodic voice, or to mark a phrase boundary) [1, 2].

These insights, which have grown over decades of music performance research, have led to numerous models of musical expression. The aim of these models is to explain the variations of loudness and tempo as a function of the

structural interpretation of the music. For example, Todd [3] proposes a model of loudness that is a function of the phrase structure of the piece. Another example is Parnutt’s model of musical accent [4].

Our current approach is limited to expressive dynamics. For this reason we will not discuss models of expressive timing here. More specifically, we will focus on the piano music of Chopin. This music is exemplary of classical music from the romantic period, which mainly evolved in Europe during the 19th century. Although this focus is admittedly very specific, it is often used to study expressive music performance (as in the seminal works of Repp [5]), since the music from the romantic period is characterized by dramatic fluctuations of tempo and dynamics.

Common dynamics annotations include *forte* (*f*), indicating a loud passage, *piano* (*p*) indicating a soft passage, *crescendo*/*decrescendo* indicating a gradual increase (resp. decrease) in loudness, respectively. Other, less well-known markings prescribe a dynamic evolution in the form of a metaphor, such as *calando* (“growing silent”), and *smorzando* (“dying away”).

Although it is clear that these annotations are a vital part of the composition, they are not always unequivocal. Their precise interpretation may vary from one composer to the other, which makes it a topic of historical and musicological study. (See Rosenblum [6] for an in depth discussion of the interpretation of dynamics markings in the works of different composers.)

Another relevant question concerns the role of dynamics markings. In some cases, dynamics markings may simply reinforce an interpretation that musicians regard as natural, by their acquaintance with a common performance practice. In other words, some annotated markings may be implied by the structure of the music. In other cases, the composer may annotate highly specific and non-obvious markings, and even fingerings, to ensure the performance achieves the intended effect. An example of this is the music of Beethoven.

The research presented here is intended to help clarify the interpretation of dynamics markings, and how these markings shape the loudness of the performance, in interaction with other aspects of the music. Very generally speaking, the aim is to develop a new methodology for musicological research, that takes advantage of the possibilities of digitized musical corpora, and of advances in statistics and

machine learning – an aim shared with Beran and Mazzola [7].

In a more specific sense, our research follows an intuition that underlies many studies of musical expression, namely that musical expression consists of a number of individual factors that jointly determine what the performance of a musical piece sounds like [8]. The goal is then to identify which factors can account for expression, in casu loudness variations, and to disentangle their contributions to the loudness of the performance.

To this end we present a rather simple model of expressive loudness variations, based on the idea of signal decomposition in terms of basis functions. The primary goal of this approach is to quantify the influence of dynamics markings on the loudness of a performance, but the model is general enough to allow for the inclusion of a wide range of features other than dynamics markings, such as pitch and motivic structure.

The outline of the paper is as follows: In section 2, we describe the model, and the basis functions used in the model. In section 3, we show how the model is used to represent loudness variations in real performances, and perform experiments to evaluate the predictive value of the model, as trained on the data. The results are discussed in section 4, and conclusions and future work can be found in section 5.

2. A LINEAR BASIS MODEL OF EXPRESSIVE DYNAMICS

As stated in the previous section, the model reflects the idea that different aspects of the music jointly shape variation in loudness. When we ignore the possibly complex ways in which aspects might interact, and assume that the eventual loudness is a weighted mixture of these aspects, a linear basis model is an obvious choice to model loudness variation. In this model, one basis function is created for each dynamics annotation, and the performed loudness is regarded as a linear combination of these basis-function. Even if it is simple, we believe that this approach captures the notion that part of the interpretation of a dynamics marking is constant, and that the degree to which the dynamics marking is followed by the performer, is variable. For example, the annotation *p* indicates that the passage spanned by the range of that *p* is to be played softly. This can be represented by a basis-function that removes a constant amount from the loudness curve over that range. The weight of that basis-function determines *how much* the performance gets softer. We will illustrate this further in subsection 2.1.

2.1 Mapping from score to basis

We distinguish between three categories of dynamics annotations, based on their scope, as shown in table 1. The first category, *constant*, represents markings that indicate a particular loudness character for the length of a passage. The passage is ended either by a new *constant* annotation, or the end of the piece. *Impulsive* annotations indicate a change of loudness for only a brief amount of time, usually only the notes over which the sign is annotated. The last

| Category | Examples |
|-----------|---|
| Constant | <i>f, ff, fff, mf, mp, p, pp, ppp, vivo, agitato, appassionato, con anima, con forza, con fuoco, dolce, dolcissimo, espressivo, leggero, leggerissimo</i> |
| Impulsive | <i>fz, sf, sfz, fp</i> |
| Gradual | <i>calando, crescendo, decrescendo, diminuendo, smorzando, perdendosi</i> |

Table 1. Three categories of dynamics markings

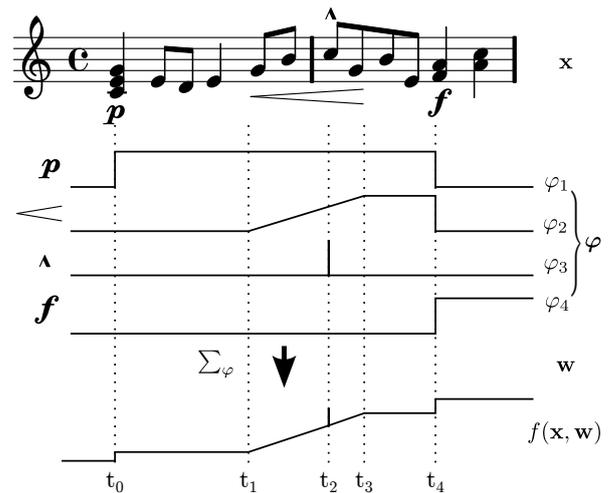


Figure 1. Example of basis functions representing dynamics annotations

category contains those annotations that indicate a gradual change from one loudness level to the other. We call these annotations *gradual*.

Based on their interpretation, as described above, we assign a particular basis function to each category. The constant category is modeled as a step function that has value 1 over the affected passage, and 0 elsewhere. Impulsive annotations are modeled by a unit impulse function, which has value 1 at the time of the annotation and 0 elsewhere. Lastly, gradual annotations are modeled as a combination of a ramp and a step function. It is 0 until the start of the annotation, linearly changes from 0 to 1 between the start and the end of the indicated range of the annotation (e.g. by the width of the ‘hairpin’ sign indicating a crescendo), and maintains a value of 1 until the time of the next constant annotation, or the end of the piece.

As an illustration, figure 1 shows a fragment of notated music with dynamics markings, and the corresponding basis-functions. The bottom-most curve is a weighted sum of the basis functions φ_1 to φ_4 (using unspecified weights).

2.1.1 Other types of basis-functions

The basis functions shown in figure 1 represent dynamics annotations, and are functions of score time. That is, when two or more notes have the same onset time, the value of

the basis function of these notes will be equal. However, we can also conceive of basis-functions more generally, as functions of the note itself, that may yield different values for notes even if their onset times coincide. This generalization allows us to represent a much larger range of score information as basis-functions.

We will briefly discuss three features that we will include in the model in the form of basis-functions. Firstly, we can include information about the decorative role of notes into the model, by defining a basis function that acts as an indicator function. This function evaluates to 1 for notes that have been marked in the score as grace notes, and to 0 otherwise.

Furthermore, we include a polynomial pitch model into our linear basis model, simply by adding basis functions that map each note to powers of its pitch value. For instance, using the midi note number representation of pitches, the four basis-functions of a third order polynomial pitch model would map a note with pitch 72 to the vector $(72^0, 72^1, 72^2, 72^3) = (1, 72, 5184, 373248)$. There is no need to treat the coefficients of this model separately – by aggregating the polynomial pitch basis functions into the overall model, the coefficients of the polynomial model simply are a subset of the weights of the model. Obviously, the ranges of the polynomial basis-functions will be very diverse. Therefore, in order to keep the model weights in roughly the same range, it is convenient to normalize all basis-functions to the interval $[0, 1]$.

Lastly, we include a more complex feature, based on Narmour's Implication-Realization model of melodic expectation [9]. This model allows for an analysis of melodies that includes an evaluation of the degree of 'closure' occurring at each note¹. Closure can occur for example due to metrical position, completion of a rhythmic or motivic pattern, or resolution of dissonance into consonance. We use an automatic melody parser that detects metric and rhythmic causes of closure [10]. The output of this parser allows us to define a basis-function that expresses the degree of closure at each note.

Note that we cannot say in advance whether the inclusion of such features will improve our model. By including the features into the model as basis functions we merely create a possibility for the model to explain loudness variations as a function of those features.

2.2 The model

To specify a linear basis model of expressive dynamics, we represent a musical performance as a list of pairs $((x_1, y_1), \dots, (x_n, y_n))$, where n is the number of notes in the performance, x_i is a representation of the score attributes of the i -th note, and y_i is the loudness value of the i -th note in the performance. We will refer to the vector (x_1, \dots, x_n) as \mathbf{x} , and to the vector (y_1, \dots, y_n) as \mathbf{y} .

We then define a basis function as a function $\varphi_k(\cdot)$ that takes the n elements of \mathbf{x} as arguments to produce a real

¹ Narmour's concept of closure is subtly different from the common notion of musical closure in the sense that the latter refers to 'ending' whereas the former refers to the inhibition of the listener's expectation of how the melody will continue. In spite of the difference in meaning, both notions are arguably related.

valued vector of size n . Once a set $\varphi = (\varphi_1(\cdot), \dots, \varphi_m(\cdot))$ of m basis functions is fixed, it can be applied to a musical score \mathbf{x} to yield a matrix $\varphi(\mathbf{x}) = (\varphi_1(\mathbf{x}), \dots, \varphi_m(\mathbf{x}))$ of size $n \times m$, where n is the number of notes in \mathbf{x} .

The definition of the elements of \mathbf{x} is not mathematically relevant, since \mathbf{x} will only appear as an argument to φ . Suffice it to say that the elements of \mathbf{x} contain basic note information such as notated pitch, onset time and offset time, and any dynamics markings that are annotated in the score.²

The model is defined as a function y of the score $\mathbf{x} = (x_1, \dots, x_n)$ and a vector of weights $\mathbf{w} = (w_1, \dots, w_m)$, such that the loudness is a linear combination of the basis functions:

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \varphi(\mathbf{x}) \quad (1)$$

Thus, for note x_i , the predicted loudness is computed as:

$$\hat{y}_i(\mathbf{w}) = f(x_i, \mathbf{w}) = \mathbf{w}^T \varphi(x_i) = \sum_j^m w_j \varphi_j(x_i) \quad (2)$$

2.3 Learning and prediction with the linear basis model

Given performances in form (\mathbf{x}, \mathbf{y}) we can use the model in equation (1) to estimate the weights \mathbf{w} , which is a simple linear regression problem. The most common approach to this kind of problem is to compute \mathbf{w} as the least squares solution [11], that is, the \mathbf{w} that minimizes the sum of the squared differences between the loudness predictions $y(\mathbf{x}, \mathbf{w})$ of the model and the observed loudness \mathbf{y} :

$$\mathbf{w}_{\mathbf{x}, \mathbf{y}} = \operatorname{argmin}_{\mathbf{w}} \sum_i^n [y_i - \hat{y}_i(\mathbf{w})]^2 \quad (3)$$

To find the optimal \mathbf{w} for a set of musical performances $((\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_K, \mathbf{y}_K))$, we can use two different approaches. One way is to compute a vector $\mathbf{w}_{\mathbf{x}_k, \mathbf{y}_k}$ for each performance k according to equation (3), and combine the $\mathbf{w}_{\mathbf{x}_k, \mathbf{y}_k}$'s in some way to form a final estimate of $\mathbf{w}_{\mathbf{x}, \mathbf{y}}$, e.g. by taking the median for each weight w_j .

Another approach is to concatenate the respective \mathbf{x}_k 's and \mathbf{y}_k 's of the performances into a single pair (\mathbf{x}, \mathbf{y}) , and to find $\mathbf{w}_{\mathbf{x}, \mathbf{y}}$ directly according to equation (3). For this to work however, φ must have the same number of columns, i.e. the same number of basis functions, for each \mathbf{x}_k . This may or may not be the case, depending on how we define our basis functions. In the following paragraph, we briefly explain two alternative approaches to defining basis functions.

2.3.1 Local and global bases

The mapping of dynamics annotations can be done in different ways. The first possibility is to define a new basis function for each dynamics marking that we encounter in the score. This means for example, that repeated *crescendo* annotations are represented in different basis functions, and

² the representation of dynamics markings can be realized for example by indicator functions over the elements of \mathbf{x}

that each *crescendo* can be fitted to the observed loudness independent of the other *crescendi*. This approach leads to basis functions that are zero throughout the piece, except for the passage where the corresponding annotation applies. We call such basis functions *local*. They have the benefit that they make the model more flexible, and therefore allow for better approximations of the data. A drawback of this method is that we obtain as many weight parameters as we have *crescendi*, rather than a single weight estimation for the *crescendo* sign in general. Also, as the number of basis functions we obtain for a musical piece varies, depending on the number of dynamics annotations in the score, the fitting method by concatenating musical performances, as proposed above, becomes impossible.

Alternatively, we can choose to assign a single basis function to each type of marking. This implies for example, that we combine the different step functions of each p in the piece into a single function, e.g. by summing them up. We call this a *global* basis function.

Note that the basis functions for the other features mentioned (features related to pitch, grace notes, and I-R closure) are all global: for each feature the values of all notes in the performance are aggregated into a single basis function. There is therefore a fixed set of basis-functions representing the global features, independent of the piece.

2.3.2 Prediction with local and global bases

In the global case, once a weight vector \mathbf{w} has been learned from a data set D , predictions for a new piece \mathbf{x} can be made easily, by computing the matrix $\varphi(\mathbf{x})$ and subsequently the dot product $f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \varphi(\mathbf{x})$.

In the case where dynamics annotations are modeled by local basis functions, the \mathbf{w} 's that have been learned for each piece in the training set may have different lengths. In this case, we split up the \mathbf{w} 's of each training piece into a set of weights \mathbf{w}_A that correspond to global basis functions (i.e. for the non-dynamics features), and the remaining weights \mathbf{w}_B , for the dynamics annotations, which vary in number. Over the weights \mathbf{w}_A (which have fixed size), we take the median. The weights \mathbf{w}_B of all pieces in the training set are pooled. This pool includes multiple weights for each dynamics marking. To predict weights for the dynamics markings of a test piece, we use a support vector machine [12], that has been trained on the pool of weights \mathbf{w}_B from the training data. The medians over the \mathbf{w}_A 's and the SVM predictions from the pool of \mathbf{w}_B 's are then appended to yield the final vector \mathbf{w} used for predicting the loudness of the new piece.

3. MODEL EVALUATION

To evaluate the different features, and basis modeling approaches we have discussed above, we use it to model and predict the loudness in a set of real performances. Details of the data set are given in subsection 3.1. We wish to highlight that this data set is larger than any other data set we know of, that has a similar level of recording precision (exact onset times and loudness for each performed note).

We evaluate two aspects of the model variations. The first is the goodness-of-fit, that is, how well can the model rep-

resent the data (subsection 3.2). The second is the predictive accuracy (subsection 3.3). In both cases, we compare different features sets, for both basis modeling approaches we discussed, *local* and *global*.

We use the following abbreviations to refer to the different kinds of features: DYN: dynamics annotations. These annotations are represented by one basis function for each marking in table 1, plus one basis function for accented notes; PIT: a third order polynomial pitch model (3 basis functions)³; GR: the grace note indicator basis; IR: two basis-functions, one indicating the degree of closure, and another representing the squared distance from the nearest position where closure occurs. The latter feature forms arch-like parabolic structures reminiscent of Todd's model of dynamics [3].

The total number of parameters in the model is thus 30 (DYN) + 3 (PIT) + 1 (GR) + 2 (IR) + 1 (constant basis) = 37, or less, depending on the subset of features that we choose. In the evaluation, we omit the feature combinations that consist of only GR and IR, since we expect their influence on loudness to be marginal with respect to the features DYN and PIT.

3.1 Data Set

For the evaluation we use the Magaloff corpus [13] – a data set that comprises live performances of virtually the complete Chopin piano works, as played by the Russian-Georgian pianist Nikita Magaloff (1912-1992). The music was performed in a series of concerts in Vienna, Austria, in 1989, on a Bösendorfer SE computer-controlled grand piano [14] that recorded the performances onto a computer hard disk. The data set comprises more than 150 pieces, adding up to almost 10 hours of music, and containing over 330,000 performed notes. These data, which are stored in a native format by Bösendorfer, were converted into standard MIDI format, representing loudness values as a parameter named *velocity*, taking values between 0 (silent), and 127 (loudest). For the purpose of this experiment, velocity values have been transformed to have zero-mean per piece.

Information about dynamics markings in the score was obtained from optical music recognition from the scanned musical scores (see [13] for details). We have used the Henle Urtext Edition wherever possible.

3.2 Goodness-of-fit of the loudness representation

To quantify how well the model is able to capture loudness variations of the performances. We compute the optimal weight vector \mathbf{w} of the model for each piece in the data set. In the global case, a single weight vector is computed on the whole data set, and is applied to the basis $\varphi(\mathbf{x})$ of each piece \mathbf{x} . In the local case, a \mathbf{w} was computed for each piece, and used to fit the model to the data. This is done for the different features and combinations discussed at the beginning of section 3.

³ The chosen polynomial order of 3 was chosen as most appropriate, after a visual inspection of scatterplots showing the relationship between loudness and pitch. The constant basis function that is part of the polynomial pitch model is omitted because it is subsumed by a default constant basis function included in every basis combination.

| Basis (global) | r | | R^2 | |
|----------------|--------------|---------|--------------|---------|
| | avg. | std. | avg. | std. |
| DYN | 0.332 | (0.150) | 0.133 | (0.117) |
| PIT | 0.456 | (0.108) | 0.219 | (0.097) |
| DYN+PIT | 0.565 | (0.106) | 0.330 | (0.122) |
| DYN+PIT+GR | 0.567 | (0.107) | 0.332 | (0.123) |
| DYN+PIT+IR | 0.575 | (0.102) | 0.341 | (0.120) |
| DYN+PIT+GR+IR | 0.577 | (0.102) | 0.343 | (0.120) |
| Basis (local) | | | | |
| DYN | 0.497 | (0.170) | 0.276 | (0.160) |
| PIT | 0.456 | (0.108) | 0.219 | (0.097) |
| DYN+PIT | 0.670 | (0.113) | 0.462 | (0.146) |
| DYN+PIT+GR | 0.671 | (0.113) | 0.463 | (0.146) |
| DYN+PIT+IR | 0.678 | (0.109) | 0.471 | (0.142) |
| DYN+PIT+IR+GR | 0.678 | (0.109) | 0.472 | (0.142) |

Table 2. Goodness of fit of the model; See section 3 for abbreviations

The results of this evaluation are shown in table 2. The goodness-of-fit is expressed in two quantities: r is the Pearson product-moment correlation coefficient, denoting how strong the observed loudness, and the loudness values of the fitted model correlate. The quantity R^2 is the coefficient of determination, which is defined as:

$$R^2 = 1 - \frac{SS_{err}}{SS_{obs}}, \quad (4)$$

where:

$$SS_{obs} = \sum_i^n (y_i - \bar{y})^2, \quad SS_{err} = \sum_i^n (y_i - \hat{y}_i(\mathbf{w}))^2. \quad (5)$$

The coefficient of determination is a measure for how much of the loudness variance is accounted for by the model. In the case of a perfect fit $R^2 = 1$, since $SS_{err} = 0$. In the undesirable case where the variance of the loudness increases by subtracting the model fit from the observations, we have $SS_{err} > SS_{obs}$, and R^2 will be negative. Table 2 lists the average and standard deviations of the r and R^2 values over 154 musical pieces.

The results show that both the strongest correlation, and the highest coefficient of determination is achieved when using local basis for dynamics markings, and including all features. This is unsurprising, since in the global setting a single weight vector is used to fit all pieces, whereas in the local setting each piece has its own weight vector. Furthermore, since adding features increases the number of parameters in the model, it will also increase the goodness-of-fit.

3.3 Predictive accuracy of the model

The additional flexibility of the model, by using local bases and adding features, may increase its goodness-of-fit. However, it is doubtful that it will help to obtain good model predictions for unseen musical pieces. To evaluate the accuracy of the predictions of a trained model for an unseen

| Basis (global) | r | | R^2 | |
|----------------|--------------|---------|--------------|---------|
| | avg. | std. | avg. | std. |
| DYN | 0.192 | (0.173) | 0.020 | (0.100) |
| PIT | 0.422 | (0.129) | 0.147 | (0.111) |
| DYN+PIT | 0.462 | (0.125) | 0.161 | (0.156) |
| DYN+PIT+GR | 0.462 | (0.125) | 0.161 | (0.156) |
| DYN+PIT+IR | 0.462 | (0.124) | 0.162 | (0.155) |
| DYN+PIT+GR+IR | 0.462 | (0.124) | 0.162 | (0.154) |
| Basis (local) | | | | |
| DYN | 0.192 | (0.179) | 0.024 | (0.109) |
| PIT | 0.415 | (0.137) | 0.149 | (0.149) |
| DYN+PIT | 0.459 | (0.126) | 0.151 | (0.220) |
| DYN+PIT+GR | 0.459 | (0.123) | 0.153 | (0.195) |
| DYN+PIT+IR | 0.455 | (0.130) | 0.141 | (0.231) |
| DYN+PIT+IR+GR | 0.457 | (0.123) | 0.188 | (0.126) |

Table 3. Predictive accuracy the model in a leave-one-out scenario; See section 3 for abbreviations

piece, we perform a leave-one-out cross-validation over the 154 pieces. The predictions are evaluated again in terms of averaged r and R^2 values over the pieces, which are shown in table 3.

The average correlation coefficients between prediction and observation for the local and global basis settings are roughly similar, ranging from weak ($r = .19$) to medium correlation ($r = .46$). In the global setting, increasing the complexity of the model does not affect its predictive accuracy, whereas in the local setting, maximal predictive accuracy is achieved for models of moderate complexity (including dynamics, pitch, and grace note information). The decrease of accuracy for more complex models is likely to be caused by overfitting.

Interestingly, the highest proportion of explained variance ($R^2 = .19$) is achieved by the predictions of the local model with all available features (DYN+PIT+IR+GR). However, it should be noted that the standard deviation of R^2 is rather large in most cases, indicating that for some pieces a much larger proportion of the loudness variance can be explained than for others.

4. DISCUSSION OF RESULTS

The results presented in the previous section show a substantial difference in the contribution of dynamical annotations (DYN) and pitch (PIT) to the performance of the model. The fact that pitch explains a larger proportion of the loudness variance than the dynamics annotations may come as a surprise, given that dynamics annotations are by nature intended to guide variations in loudness.

Although the data set spans a large set of performances, it is important to realize that the results are derived from performances of a single performer, performing the music of a single composer. The importance of pitch as a predictor for loudness may be different for other performers, composers, and musical genres. Specifically, we hypothesize that the fact that pitch has a strong predictive value for loudness in our data set may be a consequence of *melody*

lead. This phenomenon, which has been the subject of extensive study (see [15, 16]), consists in the consistent tendency of pianists to play melody notes both louder and slightly earlier than the accompaniment. This makes the melody more clearly recognizable by the listener, and may improve the sensation of a coherent musical structure. In many musical genres (though not all), the main melody of the music is expressed in the highest voice, which explains the relationship between pitch loudness.

This effect is clearly visible in figure 2, which displays observed, fitted, and predicted loudness for the final measures of Chopin's Prelude in B major (Opus 28, Nr. 11). In this plot, the loudness of simultaneous notes is plotted at different (adjacent) positions on the horizontal axis, for the ease of interpretation. Melody notes are indicated with dotted vertical lines. It is easily verified by eye that the loudness of melody notes is substantially higher than the loudness of non-melody notes. This effect is very prominent in the predictions of the model as well.⁴

Although observed and predicted loudness are visibly correlated, figure 2 shows that the variance of the prediction is substantially lower than that of the observation, meaning that expressive effects in the predicted performance are less pronounced. The lower variance is most likely caused by the fact that the (relatively small set of) model parameters has been optimized to performances of a wide range of different pieces, preventing the model from accurately capture loudness variance for individual performances. This problem may require a more sophisticated model, or alternatively, a separate treatment of musical pieces with distinct musical characters.

In spite of this, the results generally show that using a simple linear basis model, it is possible to capture a substantial proportion of loudness variations, both in function of dynamics annotations in the score and as a consequence of more implicit phenomena such as melody lead. We believe that this kind of model can provide a general methodology to study the factors that influence musical expression.

The model may also be of use to model variation in the articulation of notes. However, the applicability of the model to other aspects of musical expression, may not be straight-forward in all cases. For example, expressive timing (e.g. in terms of inter-onset interval (IOI) ratios) is a phenomenon that affects the time dimension of the performance. Therefore, it is not desirable to predict IOI values independently for simultaneous notes.

5. CONCLUSIONS AND FUTURE WORK

The work presented in this paper corroborates a growing insight in music performance research: that even if musical expression is a highly complex and subjective phenomenon, it is by no means fully unsystematic. We have shown that using a simple linear basis model, we can generate loudness predictions from musical scores that show

substantial positive correlation with loudness as observed in human performances by a professional pianist.

The model has several advantages. Firstly, it embodies the common intuition that expression in music performance is a result of multiple factors that jointly determine how the performance sounds. Secondly, it is concise: with 37 parameters, it is possible to explain almost 29% of the loudness variance in a data set of over 330,000 performed notes (table 2).

Improvements to the model can be conceived at different fronts. For example, a more sophisticated approach may be taken to infer the weight vectors from a data set. In particular, a Bayesian approach seems attractive, in which a prior probability distribution over weights is specified. Another improvement would be to learn basis-functions from the data, or adapt manually specified basis-functions. For this, techniques developed in the field of dictionary learning, such as *matching pursuit*, might be used. Finally, it is desirable to assess the quality of predicted loudness curves by subjective evaluation through listening tests, in addition to numerical comparison of predictions with target performances.

Acknowledgments

This research is supported by the Austrian Research Fund (FWF, Z159 "Wittgenstein Award"). We are indebted to Mme. Irene Magaloff for her generous permission to use her late husband's performance data for our research. We are grateful to Sebastian Flossmann for his effort in the preparation of the Magaloff corpus. For this research, we have made extensive use of free software, in particular R, python, and GNU/Linux.

6. REFERENCES

- [1] E. F. Clarke, "Generative principles in music," in *Generative Processes in Music: The Psychology of Performance, Improvisation, and Composition*, J. Sloboda, Ed. Oxford University Press, 1988.
- [2] C. Palmer, "Music performance," *Annual Review of Psychology*, vol. 48, pp. 115–138, 1997.
- [3] N. Todd, "The dynamics of dynamics: A model of musical expression," *Journal of the Acoustical Society of America*, vol. 91, pp. 3540–3550, 1992.
- [4] R. Parncutt, *Perspektiven und Methoden einer Systemischen Musikwissenschaft*. Germany: Peter Lang, 2003, ch. Accents and expression in piano performance, pp. 163–185.
- [5] B. H. Repp, "Diversity and commonality in music performance - An analysis of timing microstructure in Schumann's "Träumerei";" *Journal of the Acoustical Society of America*, vol. 92, no. 5, pp. 2546–2568, 1992.
- [6] S. P. Rosenblum, *Performance practices in classic piano music: their principles and applications*. Indiana University Press, 1988.

⁴Sound examples of musical fragments with loudness predicted by the model can be found at www.cp.jku.at/research/TRP109-N23/BasisMixer/midis.html

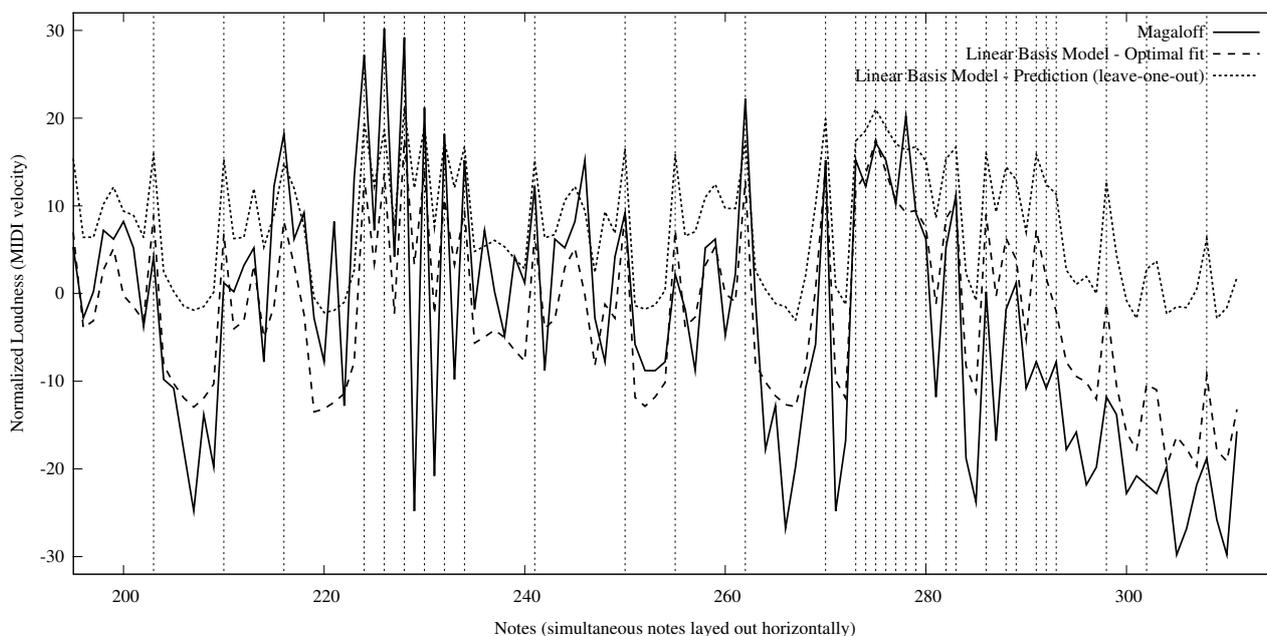


Figure 2. Observed, fitted, and predicted note-by-note loudness of Chopin’s Prelude in B major (Opus 28, Nr. 11), from measure 16 onwards; Fitting and prediction was done using the global basis DYN+PIT+GR+IR (see section 3); Vertical dotted lines indicate melody notes

- [7] J. Beran and G. Mazzola, “Analyzing musical structure and performance— a statistical approach,” *Statistical Science*, vol. 14, no. 1, pp. 47–79, 1999.
- [8] C. Palmer, “Anatomy of a performance: Sources of musical expression,” *Music Perception*, vol. 13, no. 3, pp. 433–453, 1996.
- [9] E. Narmour, *The analysis and cognition of basic melodic structures : the Implication-Realization model*. University of Chicago Press, 1990.
- [10] M. Grachten, “Expressivity-aware tempo transformations of music performances using case based reasoning,” Ph.D. dissertation, Pompeu Fabra University, Barcelona, Spain, 2006, ISBN: 635-07-094-0.
- [11] A. Björck, *Numerical methods for least squares problems*. SIAM, 1996.
- [12] B. Boser, I. Guyon, and V. Vapnik, “A training algorithm for optimal margin classifiers,” in *Fifth Annual Workshop on Computational Learning Theory*. Pittsburgh: ACM, 1992, pp. 144–152.
- [13] S. Flossmann, W. Goebel, M. Grachten, B. Niedermayer, and G. Widmer, “The Magaloff Project: An Interim Report,” *Journal of New Music Research*, vol. 39, no. 4, pp. 369–377, 2010.
- [14] R. A. Moog and T. L. Rhea, “Evolution of the Keyboard Interface: The Bösendorfer 290 SE Recording Piano and the Moog Multiply-Touch-Sensitive Keyboards,” *Computer Music Journal*, vol. 14, no. 2, pp. 52–60, 1990.
- [15] B. Repp, “Patterns of note onset asynchronies in expressive piano performance,” *Journal of the Acoustical Society of America*, vol. 100, no. 6, pp. 3917–3932, 1996.
- [16] W. Goebel, “Melody lead in piano performance: expressive device or artifact?” *Journal of the Acoustical Society of America*, vol. 110, no. 1, pp. 563–572, 2001.

A COMPARISON OF PERCEPTUAL RATINGS AND COMPUTED AUDIO FEATURES

Anders Friberg

Speech, music and hearing, CSC
KTH (Royal Institute of Technology)
afriberg@kth.se

Anton Hedblad

Speech, music and hearing, CSC
KTH (Royal Institute of Technology)
ahedblad@kth.se

ABSTRACT

The backbone of most music information retrieval systems is the features extracted from audio. There is an abundance of features suggested in previous studies ranging from low-level spectral properties to high-level semantic descriptions. These features often attempt to model different perceptual aspects. However, few studies have verified if the extracted features correspond to the assumed perceptual concepts. To investigate this we selected a set of features (or musical factors) from previous psychology studies. Subjects rated nine features and two emotion scales using a set of ringtone examples. Related audio features were extracted using existing toolboxes and compared with the perceptual ratings. The results indicate that there was a high agreement among the judges for most of the perceptual scales. The emotion ratings energy and valence could be well estimated by the perceptual features using multiple regression with $\text{adj. } R^2 = 0.93$ and 0.87 , respectively. The corresponding audio features could only to a certain degree predict the corresponding perceptual features indicating a need for further development.

1. INTRODUCTION

The extraction of features is a fundamental part of most computational models starting with the audio signal. Therefore there exists a large number of features suggested in the literature, see e.g. [1]. They can be broadly divided in two categories: (1) Low-level features often based on short-time measures. These are often different spectral features such as MFCC coefficients, spectral centroid, or the number of zero crossings per time unit but also psychoacoustic measures such as roughness and loudness. (2) Mid-level features with a slightly longer analysis window. The mid-level features are often typical concepts from music theory and music perception such as beat strength, rhythmic regularity, meter, mode, harmony, and key strength. They are often verified by using ground-truth data with examples annotated by experts. In addition, a third level consists of semantic descriptions such as emotional expression or genre, see Figure 1. The distinction between mid and low-level features is in real-

Copyright: © 2011 Anders Friberg and Anton Hedblad. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ity rather vague and was made in order to point to the differences in complexity and aims.

For modeling the higher-level concepts such as emotion description or genre it is not certain that the mid-level features derived from classic music theory (or low-level features) is the best choice. In emotion research a number of more rather imprecise overall estimations has been successfully used for a long time. Examples are pitch (high/low), dynamics (high/low) or harmonic complexity (high/low), see e.g. [2,3]. This may indicate that human music perception is retrieving something other than traditional music theoretic concepts such as the harmonic progression. This is not surprising since it demands substantial training to recognize an harmonic progression but it also points to the need for finding what we really hear when we listen to music.

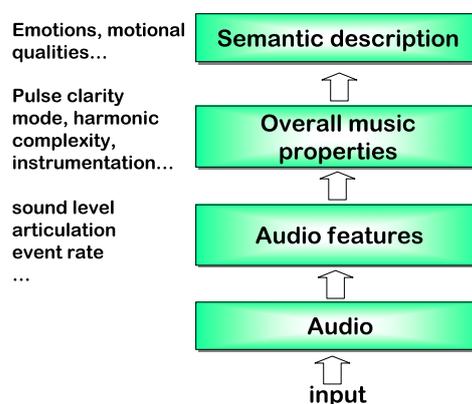


Figure 1. The different layers of music features and descriptions.

The present study is part of a series of studies in which we investigate features derived from different fields such as emotion research and ecological perception, define their perceptual values and develop computational models. We will call these perceptual features to emphasize that they are based on perception and to distinguish them from their computational counterpart.

In this paper we will report on the estimation of nine perceptual features and two emotion descriptions in a listening experiment and compare the ratings with combinations of existing audio features derived from available toolboxes.

2. RINGTONE DATABASE

The original set of music examples were 242 popular ringtones in MIDI format used in a previous experiment [4]. The ringtones were randomly selected from a large commercial database consisting of popular music of various styles. They were in a majority of cases instrumental polyphonic versions of the original popular songs. The average duration of the ringtones was about 30 s. The MIDI files were converted to audio using a Roland JV-1010 MIDI synthesizer. The resulting wav files were normalized according to the loudness standard specification ITU-R BS. 1770.

In a previous pilot experiment 5 listeners, with moderate to expert music knowledge, estimated the 9 features below for all music examples, see also [5]. The purpose was both to reduce the set so that it could be rated in one listening experiment and to enhance the spread of each feature within the set. For example, it was found that many examples had a similar tempo. The number of examples was reduced to 100 by selecting the extreme cases of each perceptual rating. This slightly increased the range and spread of each variable. This constituted the final set used in this study.

3. PERCEPTUAL RATINGS

3.1 Perceptual features

This particular selection of perceptual features was motivated by their relevance in emotion research but also from the ecological perspective, see also [5]. Several of these features were used by Wedin [6] in similar experiment. Due to experimental constraints the number was limited to nine basic feature scales plus two emotion scales.

Speed (slow-fast)

The general speed of the music disregarding any deeper analysis such as the musical tempo.

Rhythmic clarity (flowing-firm)

Indication of how well the rhythm is accentuated disregarding the rhythm pattern (c.f. pulse clarity, [7]).

Rhythmic complexity (simple-complex)

This is a natural companion to rhythmic clarity and presumably an independent rhythmic measure.

Articulation (staccato-legato)

Articulation is here only related to the duration of tones in terms of *staccato* or *legato*.

Dynamics (soft-loud)

The intention was to estimate the played dynamic level disregarding listening volume. Note that the stimuli were normalized using an equal loudness model.

Modality (minor-major)

Contrary to music theory we treat modality as a continuous scale ranging from minor to major.

Overall Pitch (low-high)

The overall pitch height of the music.

Harmonic complexity (simple-complex)

A measure of how complex the harmonic progression is. It might reflect for example the amount of chord changes and deviations from a certain key scale structure. This is presumably a difficult feature to rate demanding some knowledge of music theory.

Brightness (dark-bright)

Brightness is possibly the most common description of timbre.

Energy (low-high)

Valence (negative-positive)

These are the two dimensions of the commonly used dimensional model of emotion (e.g [8]). However, the energy dimension is in previous studies often labeled activity or arousal.

3.2 Listening experiment

A listening experiment was conducted with 20 subjects rating the features and emotion descriptions on continuous scales for each of the 100 music examples (details given in [5]).

| <i>Feature</i> | <i>Mean inter-subject corr.</i> | <i>Cronbach's alpha</i> |
|-------------------|---------------------------------|-------------------------|
| Speed | 0.71 | 0.98 |
| Rhythmic complex. | 0.29 (0.33) | 0.89 (0.89) |
| Rhythmic clarity | 0.31 (0.34) | 0.90 (0.90) |
| Articulation | 0.37 (0.41) | 0.93 (0.93) |
| Dynamics | 0.41 (0.44) | 0.93 (0.93) |
| Modality | 0.38 (0.47) | 0.93 (0.94) |
| Harmonic complex. | 0.21 | 0.83 |
| Pitch | 0.37 (0.42) | 0.93 (0.93) |
| Brightness | 0.27 | 0.88 |
| Energy | 0.57 | 0.96 |
| Valence | 0.42 (0.47) | 0.94 (0.94) |

Table 1. Agreement among the 20 subjects in terms of mean inter-subject correlation and Cronbach's alpha. A value of one indicates perfect agreement in both cases.

Could the subjects reliably estimate the perceptual features? This was estimated by the mean correlation between all subject pairs, see Table 1. In addition, for comparison with previous studies (e.g. [9]) Cronbach's alpha was also computed. The Cronbach's alpha indicated a good agreement for all ratings while the inter-subject correlation showed a more differentiated picture with lower agreement for the more complex tasks like harmonic complexity.

| | Speed | Rhythmic complexity | Rhythmic clarity | Articulation | Dynamics | Modality | Harmonic complexity | Pitch |
|---------------------|----------|---------------------|------------------|--------------|----------|----------|---------------------|---------|
| Rhythmic complexity | -0.09 | | | | | | | |
| Rhythmic clarity | 0.51*** | -0.54*** | | | | | | |
| Articulation | 0.57*** | -0.06 | 0.56*** | | | | | |
| Dynamics | 0.66*** | 0.00 | 0.53*** | 0.57*** | | | | |
| Modality | 0.19 | -0.17 | 0.01 | 0.20 | 0.03 | | | |
| Harmonic complexity | -0.37*** | 0.51*** | -0.63*** | -0.49*** | -0.31** | -0.22* | | |
| Pitch | -0.03 | -0.04 | -0.17 | -0.09 | 0.05 | 0.46*** | 0.21* | |
| Brightness | 0.01 | -0.05 | -0.16 | -0.02 | 0.12 | 0.59*** | 0.15 | 0.90*** |

Table 2. Cross-correlations between rated features averaged over subjects. N=100, p-values: * < 0.05; ** < 0.01, ***<0.001.

A closer inspection of the inter-subject correlations revealed that for some features there was one subject that clearly deviated from the rest of the group. Numbers in parenthesis refer to trimmed data when these subjects were omitted. However, the original data was used in the subsequent analysis. We interpret these results as an indication that all the measures could be rated by the subjects. Although the more complex measures like harmonic complexity obtained lower agreement the mean value across subject may still be a useful estimate.

The interdependence of the different rating scales was investigated using cross-correlations shown in Table 2. As seen in the table, there were relatively few alarmingly high values. Only about half of the correlations were significant and did rarely exceed 0.6 (corresponding to 36% covariation). The only exception was ‘pitch’ and ‘brightness’ with $r=0.9$, which is discussed below.

It is difficult to determine the reason for the high cross-correlations in the ratings at this point since there are two different possibilities. Either there is a covariation in the music examples, or alternatively, it could be the listeners that were not able to isolate each feature as intended.

Finally, the extent to which the perceptual features could predict the emotion ratings was tested. A separate multiple regression analysis was applied for each of the emotion ratings energy and valence with all the nine perceptual features as independent variables. The energy rating could be predicted with an adj. $R^2 = 0.93$ (meaning that 93% of the variation could be predicted) with four significant perceptual features. The strongest contribution was by speed followed by dynamics, while modality and rhythmic clarity contributed with a small amount. The valence rating was predicted with an adj. $R^2 = 0.87$. The strongest contribution was by modality followed by dynamics (negative), brightness, articulation, and speed. These results were unexpectedly strong given the small number of perceptual features. However, since both the feature ratings and the emotion ratings were obtained from the same subjects this is just a preliminary observation that needs to be further validated in a future study.

3.3 COMPUTED FEATURES

Computational audio features were selected from existing toolboxes that were publicly available. Two hosts were used for computing the audio features: MIRToolbox

v. 1.3.1 [10] and Sonic Annotator¹ v. 0.5. MIRToolbox is implemented in MATLAB and Sonic Annotator is a host program which can run VAMP plugins.

A list of all extracted features is shown in Table 3. Audio features were selected that we *a priori* would expect to predict a perceptual rating. Within these toolboxes we could only find *a priori* selected audio features for a subset of six perceptual ratings, namely speed, rhythmic clarity, articulation, brightness, and energy. In Table 4 below, the corresponding selected audio features are marked in grey color.

| Abbreviation | Meaning | Parameters |
|---|------------------------|----------------------------|
| <i>EX - VAMP Example plugins</i> | | |
| EX_Onsets | Percussion Onsets | Default |
| EX_Tempo | Tempo | Default |
| <i>MT - MIRToolbox</i> | | |
| MT_ASR | Average Silence Ratio | Default |
| MT_Bright_1.5k | Brightness | Default |
| MT_Bright_1k | Brightness | Cutoff: 1000 Hz |
| MT_Bright_3k | Brightness | Cutoff: 3000 Hz |
| MT_Event | Event Density | Default |
| MT_Mode_Best | Modality | Model: Best |
| MT_Mode_Sum | Modality | Model: Sum |
| MT_Pulse_Clarify_1 | Pulse Clarity | Model: 1 |
| MT_Pulse_Clarify_2 | Pulse Clarity | Model: 2 |
| MT_SC | Spectral Centroid | Default |
| MT_SF | Spectral Flux | Default |
| MT_Tempo_Auto | Tempo | Model: Autocorr |
| MT_Tempo_Both | Tempo | Model: Autocorr & Spectrum |
| MT_Tempo_Spect | Tempo | Model: Spectrum |
| MT_ZCR | Zero Crossing Rate | Default |
| <i>MZ - VAMP plugins ported from the Mazurka project.</i> | | |
| MZ_SF_Onsets | Spectral Flux Onsets | Default |
| MZ_SRF_Onsets | Spectral Reflux Onsets | Default |
| <i>QM - VAMP plugins from Queen Mary.</i> | | |
| QM_Mode | Modality | Default |
| QM_Onsets | Onset detection | Default |
| QM_Tempo | Tempo | Default |

Table 3. Overview of all computed audio features.

¹ <http://www.omras2.org/SonicAnnotator>

| | Speed | Rhythmic complex. | Rhythmic clarity | Articulation | Dynamics | Modality | Harmonic complex. | Pitch | Brightness | Energy | Valence |
|----------------|----------------|-------------------|------------------|--------------|----------|----------|-------------------|---------|------------|----------------|----------|
| MT_Event | 0.65*** | 0.08 | 0.33*** | 0.52*** | 0.47*** | -0.01 | -0.27** | -0.08 | -0.01 | 0.57*** | -0.01 |
| MT_Pulse_cla1 | 0.61*** | -0.22* | 0.73*** | 0.69*** | 0.56*** | 0.09 | -0.40*** | -0.13 | -0.07 | 0.67*** | 0.03 |
| MT_Pulse_cla2 | -0.08 | -0.34*** | 0.16 | 0.04 | -0.12 | 0.04 | -0.11 | -0.01 | -0.01 | -0.07 | 0.06 |
| MT_ASR | 0.21* | -0.03 | 0.44*** | 0.62*** | 0.28** | -0.03 | -0.26** | -0.09 | -0.13 | 0.33*** | -0.04 |
| MT_Bright_1k | 0.26** | -0.04 | 0.33*** | 0.18 | 0.53*** | -0.03 | -0.19 | 0.15 | 0.20* | 0.34*** | -0.13 |
| MT_Bright_1.5k | 0.31** | -0.06 | 0.42*** | 0.28** | 0.55*** | -0.05 | -0.22* | 0.08 | 0.16 | 0.38*** | -0.13 |
| MT_Bright_3k | 0.37*** | -0.07 | 0.52*** | 0.40*** | 0.47*** | -0.08 | -0.26** | -0.02 | 0.04 | 0.41*** | -0.15 |
| MT_Mode_best | 0.04 | -0.09 | -0.11 | -0.11 | -0.1 | 0.67*** | -0.01 | 0.41*** | 0.51*** | 0 | 0.69*** |
| MT_Mode_sum | -0.04 | 0.09 | -0.05 | 0.04 | 0.11 | -0.47*** | 0.15 | -0.19 | -0.25* | -0.03 | -0.43*** |
| MT_SC | 0.31** | -0.12 | 0.45*** | 0.34*** | 0.34*** | -0.1 | -0.23* | -0.03 | 0.03 | 0.31** | -0.15 |
| MT_SF | 0.72*** | -0.03 | 0.66*** | 0.67*** | 0.66*** | -0.03 | -0.39*** | -0.15 | -0.08 | 0.75*** | -0.07 |
| MT_Tempo_both | -0.11 | 0.17 | -0.1 | -0.06 | 0.03 | -0.21* | 0.15 | -0.09 | -0.08 | -0.09 | -0.22* |
| MT_Tempo_auto | -0.08 | 0.02 | -0.01 | 0 | -0.02 | -0.11 | 0.13 | -0.03 | 0.02 | -0.08 | -0.13 |
| MT_Tempo_spect | 0.02 | 0.12 | 0.04 | 0.08 | 0.07 | -0.17 | 0.08 | -0.05 | -0.05 | 0.03 | -0.16 |
| MT_ZCR | 0.43*** | 0.04 | 0.27** | 0.17 | 0.53*** | -0.02 | 0.01 | 0.14 | 0.15 | 0.45*** | -0.14 |
| QM_Onsets | 0.73*** | 0.24* | 0.15 | 0.38*** | 0.50*** | 0 | -0.13 | -0.06 | 0 | 0.62*** | -0.01 |
| EX_Onsets | 0.55*** | 0.08 | 0.36*** | 0.52*** | 0.34*** | -0.09 | -0.24* | -0.13 | -0.06 | 0.45*** | -0.06 |
| EX_Tempo | 0.15 | -0.12 | 0 | -0.05 | 0.08 | -0.05 | -0.01 | -0.03 | -0.04 | 0.06 | -0.1 |
| MZ_SF_Onsets | 0.61*** | 0.17 | 0.04 | 0.27** | 0.41*** | 0.06 | -0.17 | -0.02 | 0.06 | 0.51*** | 0.05 |
| MZ_SRF_Onsets | 0.64*** | 0.15 | 0.24* | 0.32*** | 0.40*** | -0.06 | -0.16 | -0.05 | -0.02 | 0.55*** | -0.01 |
| QM_Mode | 0 | 0.09 | 0.1 | 0.08 | 0.08 | -0.58*** | 0.02 | -0.26* | -0.39*** | 0.02 | -0.55*** |
| QM_Tempo | 0.09 | -0.21* | 0.04 | 0.01 | -0.03 | 0.18 | -0.05 | 0.04 | 0.05 | 0.02 | 0.08 |

Table 4. Correlations between all perceptual ratings and computed features. Dark grey areas indicate those audio features that *a priori* were selected for predicting the perceptual ratings. N=100, p-values: * < 0.05; ** < 0.01, ***<0.001.

Each feature was computed using the default settings and in certain cases using different available models. For each sound example one feature value was obtained. All the onset measures were converted to onsets per second by counting the number of onsets and dividing by the total length of each music example. For a more detailed description, see [11].

4. COMPARISON

4.1 Correlations

The correlation between all the perceptual ratings and the computed features are shown in Table 4. There is a large number of features that correlates significantly as indicated by the stars in the table. This may serve as an initial screening where we can sort out all non-significant relations. Then the size of the correlations should be considered. According to Williams [12] a correlation coefficient between 0.4-0.7 should be considered a substantial relationship and coefficients between 0.7-0.9 should be considered a marked relationship. Following this rather *ad hoc* rule-of-thumb we note that there were only four features with a marked relationship, three of them included in the list of *a priori* selected features. These were speed and one onset model (QM_Onsets, $r=0.73$), speed and spectral flux (MT_SF, $r=0.72$), rhythmic complexity and the pulse clarity model 1 (MT_Pulse_cla1, $r=0.73$), and energy and spectral flux (MT_SF, $r=0.75$). Many of the expected relations do in fact correlate with rather high

values but there are also a number of correlations that are more difficult to interpret.

As seen in Table 4, Speed is significantly correlating with many audio features. All the onset features have rather high correlations but note that none of the tempo features were significant. This result indicates that the perceived speed has little to do with the musical tempo. The results verify that the number of onsets per second is the most appropriate equivalent for perceptual speed. This was recently also verified by Bresin and Friberg [13] and Madison and Paulin [14].

Rhythmic clarity is highly correlated with pulse clarity model 1 which confirms that it is a similar measure. The pulse clarity model was developed using similar perceptual ratings [7]. Note that the second pulse clarity model is not significant and instead correlates somewhat with rhythmic complexity.

The spectral flux is an interesting case as it is correlating with almost all perceptual ratings. The high correlation with speed is not surprising since it is a measurement of spectral changes over time.

The rating of dynamics is also puzzling. As mentioned, all sound examples were normalized for equal loudness. Thus, one would possibly expect rather small variations in the ratings. Since dynamics is associated with spectral changes, the correlation with spectral features is natural. However, the strong correlations with temporal features are more difficult to interpret.

The rating of brightness had rather low correlation with any audio feature. One would have expected better correlation with the spectral features. The largest correlation is with the function for modality, using the method choos-

ing the best major and minor key. The correlation is positive, meaning major songs sound brighter. This can be due to the uncontrolled stimuli; songs in the stimuli with major key might be brighter. Another possibility is that people perceive major keys as brighter than minor, even with the same timbre. In addition the rated brightness correlated strongly with rated pitch ($r=0.9$). All this indicates that the brightness rating did not work the way we intended. Rather than rating the spectral quality of the sound the subjects seem to have rated a more complex quality possibly related to pitch and mode.

4.2 Regression analysis

To find out how well the perceptual features could be predicted we performed separate multiple regression analyses with each perceptual feature as the dependent variable and all the audio features as independent variables. Since the number of independent variables (22) were too high in relation to number of cases (100) we applied a step-wise multiple regression. However, this procedure is questionable and the results should be considered as preliminary and without consideration of details. The multiple regression coefficient R^2 determines how well the regression model fits the actual data. A summary of the result is shown in in Table 5. Also shown is the number of variables that were selected by the step-wise procedure in each analysis.

| Dependent variable | Adjusted R^2 | Number of variables |
|---------------------|----------------|---------------------|
| Speed | 0.76 | 8 |
| Rhythmic complexity | 0.14 | 2 |
| Rhythmic clarity | 0.52 | 1 |
| Articulation | 0.62 | 5 |
| Dynamics | 0.67 | 6 |
| Modality | 0.54 | 5 |
| Harmonic complexity | 0.23 | 5 |
| Pitch | 0.16 | 1 |
| Brightness | 0.29 | 2 |
| Energy | 0.68 | 5 |
| Valence | 0.50 | 2 |

Table 5. Summary of the step-wise regression analysis. Features in grey were predicted *a priori*.

All the regressions were significant but as seen in the table, the amount of explained variance (R^2) was rather modest. The regression results were in general similar to the correlations in Table 3. For example, speed could be rather well predicted as expected and the analysis included eight variables.

5. CONCLUSIONS AND DISCUSSION

The initial results of the perceptual ratings indicate that there was a rather good agreement among the listeners and that they could reliably assess the different musical aspects. The only scale that seemed to be problematic was the rating of brightness, also indicated by the high correlation between brightness and pitch. The emotion ratings could be well estimated by the perceptual features

using multiple regression with adj. $R^2 = 0.93$ and 0.87 , respectively.

The computed audio features correlated often with the perceptual ratings that were *a priori* expected. However, the audio features could only to a rather limited extent predict the perceptual ratings. Using multiple regression the best prediction was of speed with an adjusted $R^2 = 0.76$.

The selection of music examples is likely to have a strong effect on the results. It sets the variation of each feature and thus indirectly influences the judgment. It also influences the accuracy of the computed features. In addition, the current examples, which were converted from MIDI, had a rather limited timbral variation since they were all produced using the same synthesizer. Thus a future goal is to replicate this experiment using a different music set

The present selection of audio features only included a small subset of all previously suggested algorithms. Certainly, a broader selection of audio features would yield better results. Nevertheless, we think that these results point to the need for further development of audio features that are more specifically designed for these perceptual features. The only exception here was pulse clarity. It is likely that a small selection of such audio features would efficiently predict also higher-level semantic descriptions as indicated in Figure 1.

ACKNOWLEDGEMENTS

We would like to thank Erwin Schoonderwaldt who prepared the stimuli and ran the pilot experiment. This work was supported by the Swedish Research Council, Grant Nr. 2009-4285.

6. REFERENCES

- [1] J. J. Burred, and A. Lerch, "Hierarchical Automatic Audio Signal Classification," in Journal of the Audio Engineering Society, 52(7/8), 2004, pp. 724-738.
- [2] K. Hevner, "The affective value of pitch and tempo in music," in American Journal of Psychology, 49, 1937, pp. 621-30.
- [3] A. Friberg, "Digital audio emotions — An overview of computer analysis and synthesis of emotions in music," In Proc. of the 11th Int. Conference on Digital Audio Effects (DAFx-08), Espoo, Finland 2008, pp. 1-6.
- [4] A. Friberg, E. Schoonderwaldt, & P. N. Juslin, "CUEx: An algorithm for extracting expressive tone variables from audio recordings," in Acta Acoustica united with Acta Acoustica, 93(3), 2005, pp. 411-420.
- [5] A. Friberg, E. Schoonderwaldt, and A. Hedblad, "Perceptual ratings of musical parameters," In H. von Loesch and S. Weinzierl (eds.) Gemessene Interpretation - Computergestützte Aufführungs-

analyse im Kreuzverhör der Disziplinen, Mainz: Schott 2011 (Klang und Begriff 4). (forthcoming)

- [6] L. Wedin, "A Multidimensional Study of Perceptual-Emotional Qualities in Music," in *Scand. J. Psychol.*, 1972, 13, pp. 241-257.
- [7] O. Lartillot, T. Eerola, P. Toiviainen, and F. Fornari, "Multi-Feature Modeling of Pulse Clarity: Design, Validation and Optimization," In *Proceedings of the International Conference on Music Information Retrieval (ISMIR 2008)*, 2008, pp. 521-526.
- [8] J. A. Russell, "A circumplex model of affect," in *Journal of Personality and Social Psychology*, 1980, 39, pp. 1161 - 1178.
- [9] V. Alluri, and P. Toiviainen, P. "In Search of Perceptual and Acoustical Correlates of Polyphonic Timbre," *Proc. of the Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM)*, Jyväskylä, Finland, 2009.
- [10] O. Lartillot, and P. Toiviainen, "A MATLAB toolbox for musical feature extraction from audio," in *Proc. Of the 10th Int. Conference on Digital Audio Effects*, 2007, (DAFx-07).
- [11] A. Hedblad, *Evaluation of Musical Feature Extraction Tools Using Perceptual Ratings*. Master thesis, KTH, 2011, (forthcoming).
- [12] F. Williams, *Reasoning With Statistics*. Holt, Rinehart and Winston, New York, 1968.
- [13] R. Bresin, and A. Friberg, "Emotion rendering in music: range and characteristic values of seven musical variables," *Cortex*, 2011, in press.
- [14] G. Madison, and J. Paulin, "Relation between tempo and perceived speed," in *J. Acoust. Soc. Am.*, 128(5), 2010.

INVESTIGATION OF THE RELATIONSHIPS BETWEEN AUDIO FEATURES AND INDUCED EMOTIONS IN CONTEMPORARY WESTERN MUSIC

Konstantinos Trochidis

LEAD-CNRS

Université de Bourgogne

Konstantinos.Trochidis@u-bourgogne.fr

Charles Delbé

LEAD-CNRS

Université de Bourgogne

Charles.Delbe@u-bourgogne.fr

Emmanuel Bigand

LEAD-CNRS

Université de Bourgogne

Emmanuel.Bigand@u-bourgogne.fr

ABSTRACT

This paper focuses on emotion recognition and understanding in Contemporary Western music. The study seeks to investigate the relationship between perceived emotion and musical features in the fore-mentioned musical genre. A set of 27 Contemporary music excerpts is used as stimuli to gather responses from both musicians and non-musicians which are then mapped on an emotional plane in terms of arousal and valence dimensions. Audio signal analysis techniques are applied to the corpus and a base feature set is obtained. The feature set contains characteristics ranging from low-level spectral and temporal acoustic features to high-level contextual features. The feature extraction process is discussed with particular emphasis on the interaction between acoustical and structural parameters. Statistical relations between audio features and emotional ratings from psychological experiments are systematically investigated. Finally, a linear model is created using the best features and the mean ratings and its prediction efficiency is evaluated and discussed.

1. INTRODUCTION

The expressive aspects of music are the most difficult to analyze structurally, inducing a large variety of emotional responses in humans. The richness of these responses is what motivates an engagement with music [1]. Many studies indicate the important distinction between one's perception of the emotion(s) expressed by music and the emotion(s) induced by music. Studies of the distinctions between perception and induction of emotion have demonstrated that both can be subjected to not only the social context of the listening experience, but also to personal motivation [2].

Modeling the perception of expressive musical content is highly useful in MIR applications such as emotion based classification and recommendation systems, radio and TV broadcasting programs, and music therapy defining appropriate musical repertoires for research in patients suffering from Alzheimer or Bibromilagic. Due to the highly conceptual elusiveness of emotions and the

limitation of computational methods mainly based on low level features, modeling and prediction of emotion in music remains a particularly difficult task. Research on music and emotion has always focused in music genres such as Western Popular or Classical music.

To the best of our knowledge limited research has been conducted in the field of Contemporary Western music. The term "Contemporary art music" is used for Western art-tradition music written since 1945. Characteristic structures in Western music like systematic variation of tempo, mode and timbre, which are identified in Classical or popular modern music, do not necessarily exist in Contemporary art music. This raise questions such as:

- 1) Can an emotional response be triggered when the mentioned features are not present?
- 2) Which other features contribute to emotional reaction?
- and 3) Are the same features contribute to emotion generation in both musicians and non musicians?

The present paper deals with the above issues. Section 2 provides background material on previous research on music studies, while Section 3 presents the ground truth and the experiments carried out with musicians and non-musicians volunteers. Section 4 describes the audio feature extraction and representation while, Section 5 discusses the statistical selection of features and modeling of music emotional regression. Discussion and conclusions are drawn in Section 6.

2. RELATED WORK

Many psychological models have been used in studies concerning music and emotion. The main approaches existing in the literature are the discrete and the dimensional models [3]. A comparison of the discrete and dimensional models of emotions in music can be found in [4]. According to the categorical approach, emotions are conceptualized as discrete unique entities and contain a certain basic number of emotion categories from which all the emotional states are derived. There is an agreement towards researchers representing this approach as to five basic emotions: happiness, sadness, anger, fear and disgust [5], [6].

In the dimensional approach, emotions are expressed on a Cartesian coordinate system according to two axes those of valence and arousal. The model depicted in figure 1 shows Russell's [7] circumplex model of affect, where the axes measure activation and pleasure. Happiness and Anger are located at the top of the vertical

(arousal) axis, Serenity and Sadness are located at the bottom. On the horizontal axis (pleasure), Happiness and Serenity are more positive emotions than Anger and Sadness, and so these pairs are located on the right and left side respectively of this axis of Russel's space. In the late 1990s, Thayer [8] proposed a two-dimensional mood model that uses individual adjectives which collectively form a mood pattern. This dimensional approach adopts the theory that mood is entailed from two factors: Stress (happy/anxious) and Energy (calm/energetic), and divides music mood into four clusters: Contentment, Depression, Exuberance and Anxious/Frantic.

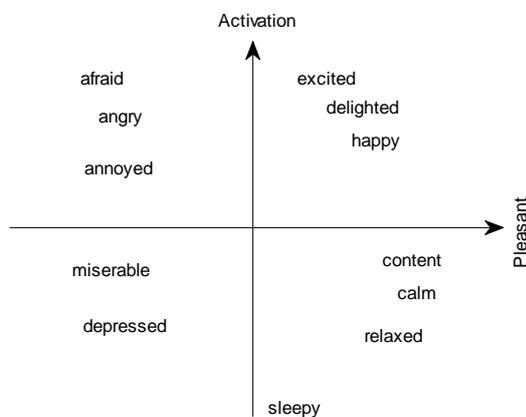


Figure 1. Russell's Circumplex model of emotion.

Music emotion representation in an affective space gained lots of interest among researchers. Hevner's study [9] arranges music emotion in a circle of eight clusters. Each cluster represents a certain emotion and contains seven to eleven adjectives grouped together. Shubert [10] argues that emotional reaction depends on the interaction between different factors such as musical mode, tempo, loudness and pitch. In [11] the time course of emotional responses to music is investigated in both untrained and trained musicians using musical excerpts of increasing duration ranging from 250 ms to 20 seconds. Findings show that less than 1s of music is enough to instill elaborated emotional responses in listeners

The four main emotion classes of Thayer's model were used as the emotion model in [12]. Three different feature sets were adopted for music representation, namely intensity, timbre and rhythm. Gaussian mixture models were used to model each of the four classes. An interesting contribution of this work was a hierarchical classification process, which first classifies a song into high/low energy (vertical axis of Thayer's model), and then into one of the two high/low stress Classes.

Emotion recognition is modelled as a regression task in [13]. Volunteers rated a training collection of songs in terms of arousal and valence in an ordinal scale of values from -1 to 1 with a 0.2 step. The authors then trained regression models using a variety of algorithms and a variety of extracted features.

A model predicting perceived emotions based on a set of features extracted from soundtrack music is given in [14]. Three separate data reduction techniques, namely stepwise regression, principal component analysis, and

partial least squares are compared.

The effectiveness of current music understanding processes and music intelligent systems is mainly hampered by the so called semantic gap between human perception and cognition and on the other hand by the low level music features which are mostly statistics of spectral and temporal characteristics in the signal.

Therefore, many researchers [15], [16], [17] try to bridge the semantic gap between the low level features and high level semantics, which humans perceive and understand, by merging different modalities such as low level acoustical features and social data including lyrics, tags and web logs.

3. EXPERIMENTAL SETUP

We decided to analyze and explore Western Contemporary art music because several of its features are shared differently by other musical genres. The main concept that best describes contemporary music is confusion in listening. Harmony does not necessarily play an important role. Thus, there is a difficulty of extracting and interpreting harmonic information because of the lack of tonal reference system. Composition contains clustered sounds and disharmonic intervals very different compared to the ones found in Western Classical music or modern popular music. Instrumentation is very complex and musicians use their instruments for producing sounds very different from those encountered in the Classical repertoire. Mixed Electro acoustic and traditional instruments raises the problem of which information of inharmonic sounds related to timbre can be captured and represented.

3.1 Method

The data used in this paper are based on a previous study [11]. Twenty participants without musical training (referred to as non musicians) and 20 with an average of 10 years of musical training and instrumental practice participated in this experiment. A set of 27 musical excerpts of Contemporary music is selected by music theorists and psychologists according to several constraints. All excerpts were expected to convey a strong emotional experience. They were chosen to illustrate a large variety of emotions, and to be representative of key musical periods of Contemporary Western music. The excerpts showed a great variation in musical structure including harmony, rhythm, tempo, timbre and instrumentation. The participants were asked first to listen to all excerpts and then focus their attention on their private emotional experience. Next, they were asked to look for excerpts that induced similar emotional experience based on arousal and valence dimensions (whatever that may be) and to drag the corresponding icons in order to group these excerpts. They corresponded either to the beginning of a musical movement, or to the beginning of a musical theme or idea. An average duration of 30s sounded appropriate. We adopted a dimensional approach for emotional labeling because it avoids a strict classification and accounts for similarity and dissimilarity of emotions. Linguistic

labels remain problematic and may simplify the emotional reaction and further disregard the difference between induced and perceived emotions.

The groupings of participants were then converted into a 27x27 matrix of co-occurrence. Each cell of the matrix indicated the average number of times that two excerpts were grouped together. The subtraction of the average matrix of occurrence from 1 resulted in a matrix of dissimilarity. The matrices obtained were highly correlated for musicians and non-musicians, ($r = .65$, $p < .001$). The resulting matrices were analysed with MDS and cluster analysis methods. The locations of the 27 excerpts along the two principal dimensions are presented in Figures 2 and 3. The vertical axis represents musical excerpts that varied obviously by their arousal level (with low arousal pieces at the bottom, and high arousal pieces at the top). The horizontal axis represents presumably musical excerpts that differ by their emotional valence (with positive valence on the right and negative valence on the left).

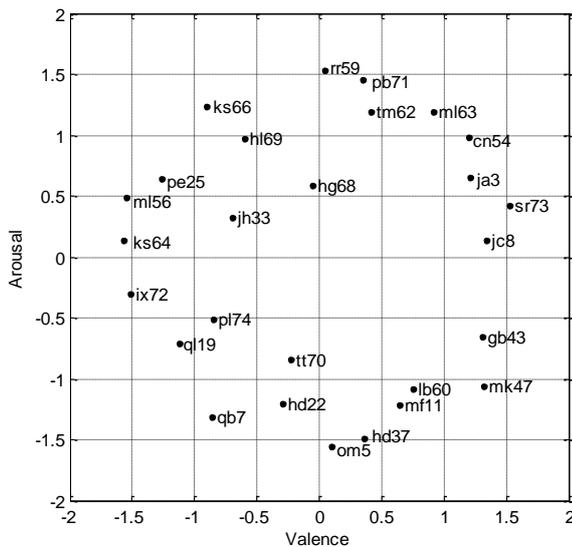


Figure 2. Geometrical representation for the 27 Contemporary music excerpts in musicians, resulting from the MDS analysis.

4. AUDIO FEATURE EXTRACTION

4.1 Low-level acoustical features

A theoretical selection of musical features was made based on music characteristics such as timbre, harmony, rhythm and dynamics. A total of 324 features were extracted from the music excerpts representing information related to the above concepts. The MIR Toolbox for MATLAB was used to compute the various low and high level descriptors [18].

4.1.1 Rhythmic features

A rhythmic analysis of the music signals was performed. Descriptors such as the fluctuation (the rhythmic periodicity along auditory frequency channels), the estimation of notes onset times and the number of onsets per second were computed. Finally, the tempo was estimated.

4.1.2 Timbre features

Mel Frequency Cepstral Coefficients (MFCCs) are used for speech recognition and music modeling. To derive the MFCCs, the signal was divided into frames and the amplitude spectrum for each frame was calculated. Next, its logarithm was taken and converted to Mel scale. Finally,

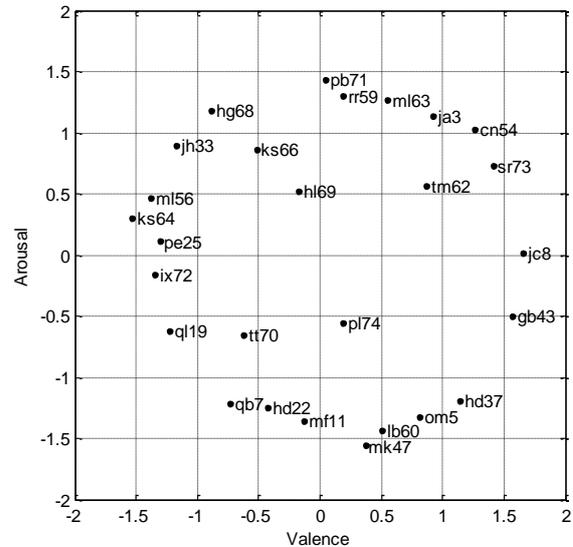


Figure 3. Geometrical representation for the 27 Contemporary music excerpts in non-musicians, resulting from the MDS analysis.

the discrete cosine transform was implemented. We selected the first 13 MFCCs. Another set of 4 features related to timbre textures were extracted from the Short-Term Fourier Transform: spectral centroid, spectral rolloff, spectral flux and flatness which indicate whether the spectrum distribution is smooth or spiky. The size of the frames used to compute the timbre descriptors was 0.05 sec half-overlapped.

4.1.3 Tonal features

The signals were also analyzed according to the tonality context. Descriptors such as the Chromagram (energy distribution of the signals wrapped in the 12 pitches), the key strength (i.e. the probability associated with each possible key candidate, through a cross correlation with the Chromagram and all possible key candidates), the tonal Centroid (a six dimensional vector derived from the Chromagram corresponding to the projection of the chords along circles of fifths or minor thirds) and the harmonic change detection function (flux of the tonal Centroid) were extracted.

4.1.4 Dynamic features

We computed information related to the dynamics of the music signals such as the RMS and the percentage of low energy frames to see if the energy is evenly distributed throughout the signals or certain frames are more contrasted than others. For all features a series of statistical descriptors were computed such as the mean, the standard deviation and the linear slope of the trend along frames,

i.e. the derivative. For all Mel Frequency Cepstral Coefficients the first and second derivative were computed. Also the maximal periodicity detected in the frame-by-frame evolution of the values was estimated through the computation of the autocorrelation sequence and the amplitude of this periodicity.

4.2 High-level Contextual features

While low-level descriptors such as loudness (perception of sound intensity) or pitch (perception of fundamental partials) account for perceptual aspects of music, higher-level ones, such as pulse, harmony or complexity account for contextual aspects, i.e. they refer to the cognitive perception and aspects of music. Many models using low level features successfully predicted the dimension of Arousal. The retrieval, however, of Valence has proved to be difficult to measure by using only low level information [19]. In order to tackle this problem, we used a set of five high level features in conjunction with the low level descriptors which are described above.

4.2.1 Pulse Clarity

This descriptor measures the sensation of pulse in music. Pulse can be described as a fluctuation of musical periodicity that is perceptible as “beatings” in a sub-tonal frequency band below 20Hz. The musical periodicity can be melodic, harmonic or rhythmic as long as it is perceived the listener as a fluctuation in time.

4.2.2 Articulation

Articulation usually refers to the way in which a melody is performed. If a pause is clearly noticeable in between each note in the melodic, the articulation of the melody is *staccato*, which means “detached”. On the other hand, if there is no pause in between the notes of the melody then the melody is *legato*, meaning “linked”. This feature attempts to estimate the articulation from musical audio signals by attributing to it an overall grade that ranges continuously from zero (*staccato*) to one (*legato*).

4.2.3 Mode

This feature refers to a computational model that detects between major and minor excerpts. It calculates an overall output that continuously ranges from zero (minor mode) to one (major mode).

4.2.4 Event density

This descriptor measures the overall amount of simultaneous events in a musical excerpt. This can be melodic, harmonic and rhythmic, as long as they can be perceived as independent entities by the human cognition.

4.2.5 Brightness

This descriptor measures the sensation of how bright of a music excerpt is felt to be Attack, articulation, or the unbalance or lacking of partials in other regions of the frequency spectrum can influence its perception.

5. FEATURE SELECTION

From the selected features, only those whose correlation with the ratings is sufficiently statistically significant (with a p-value lower than .05) are selected. The selected features are ordered from the most correlated to the least correlated ones. Features that are not sufficiently independent with respect to the better scoring ones (with a normalized cross correlation exceeding 0.6) were removed as well. In order to see how these acoustic features may account for the present data, a normalized step-wise regression of the coordinates of the pieces on the two axes was performed using the best features. Table 1 and 2 provide the outcome of the multiple linear regression analysis of the acoustic features over the coordinates of the pieces for musicians and non musicians. The resulting model provides a good account of the arousal for musicians (adjusted $R^2 = 0.72$, see Table 1), with the periodicity amplitude of flatness ($\beta = 0.60$) and the entropy of the magnitude of the highest peak in the chromagram ($\beta = -0.43$) contributing the most, followed by the flux of the tonal centroid and the mean derivative of the 3d mfcc band. On the other hand, the regression model provided a moderate account of valence with $R^2 = 0.57$, with the mean of pulse clarity ($\beta = 0.74$) (i.e. the perceived sensation of pulse) contributing the most, followed by the mean of articulation ($\beta = 0.68$) and brightness ($\beta = -0.40$).

| Valence | β | Arousal | β |
|---------------|---------|--------------------|---------|
| Pulse_clarity | 0.74 | FlatnessPeriodAmp | 0.60 |
| Articulation | 0.68 | chromagramPeakstd | -0.43 |
| brightness | -0.40 | Tonal_hdcf | 0.33 |
| Event_density | -0.19 | Dmfcc_mean_3 | -0.18 |
| Mode_Mean | 0.17 | FlatnessPeriodFreq | -0.06 |

Table 1. Outcome of the multiple linear regression analysis of the acoustic features over the coordinates for musicians.

The outcome of the multiple regression analysis of the acoustic features over the coordinates of the pieces for non musicians is presented in Table 2. One can see that the results are very similar to that of the musicians. The resulting model provides a good account $R^2 = 0.67$ of the arousal for the non musicians, with the periodicity amplitude of flatness ($\beta = -0.58$) and the entropy of the magnitude of the highest peak in the chromagram contributing the most ($\beta = -0.46$) followed by the flux of the tonal centroid. The regression model provided a moderate account of valence with $R^2 = 0.62$, with the mean of pulse clarity and the mean derivative of the 10th mfcc band contributing the most, followed by the mean of brightness.

| Valence | β | Arousal | β |
|---------------|---------|---------------------|---------|
| Articulation | 0.83 | FlatnessPeriodAmp | 0.58 |
| Pulse_clarity | 0.68 | chromagramPeakstd | -0.46 |
| brightness | -0.50 | Tonal_hdcf | 0.25 |
| Event_density | -0.19 | FlatnessPeriodFreq | -0.19 |
| Mode_mean | 0.05 | RoughnessPeriodFreq | -0.009 |

Table 2. Outcome of the multiple linear regression analysis of the acoustic features over the coordinates for non musicians.

6. CONCLUSIONS

In the present paper the relationships between music features and emotion perception in the case of Contemporary Western music are investigated. A systematic analysis of the musical stimuli shows that low level spectral and temporal features such as flatness and chroma features are efficient in modeling the emotion perception of arousal dimension, while high-level contextual information such as articulation, pulse clarity, mode and brightness succeed to measure the more cognitive nature of valence. The results contradict the widespread opinion that understanding of contemporary western music is restricted to highly trained listeners. It is shown that the emotion processing mechanism is quite similar for musicians and non musicians with the same low level spectral, temporal features correlated with arousal and high level contextual features correlated with valence dimension.

Contemporary Western music can serve successfully as stimulus for studying the emotional processing mechanism in music. An emotional response can be still triggered when characteristic structures and features of Western popular or Classical music are not present.

Future work will explore the effectiveness of new features extracted from physiological signals such as EEGs to bridge the semantic gap between high level knowledge related to the cognitive aspects of emotion and low level acoustical features. Furthermore, a larger Contemporary music dataset will be constructed and new audio features will be designed and tested to allow for better statistical results.

7. REFERENCES

- [1] E. Bigand, S. Vieillard, F. Madurell, J. Marozeau & A. Dacquet, "Multidimensional scaling of emotional responses to music: The effect of musical expertise and of the duration of the excerpts", in *Cognition & Emotion*, 2005, 19(8), 1113–1139.
- [2] P. N Juslin and P. Luukka, "Expression, perception, and induction of musical emotions: A review and questionnaire study of every day listening", in *Journal of New Music Research*, 2004, 33, 217–238.
- [3] P. Juslin and J. Sloboda, *Music and emotion: Theory and research*. Oxford, England: Oxford University Press, 2001.
- [4] T. Eerola, & J.K Vuoskoski. "A comparison of the discrete and dimensional models of emotion in music", in *Psychology of Music*, 2011, 39(1), 18–49.
- [5] R. Plutchik. *The psychology and biology of emotion*. Harper Collins, New York, 1994.
- [6] T.D. Kemper. How many emotions are there? Wedding the social and the autonomic components. *American Journal of Sociology*, 93:263–289, 1987.
- [7] J. A. Russell, "A circumplex model of affect," in *Journal of Psychology and Social Psychology*, 1980, 39, 6, 1161-1178.
- [8] R. E. Thayer. *The biopsychology of mood and arousal*. Oxford University Press, 1989.
- [9] K. Hevner, "Expression in music: a discussion of experimental studies and theories" in *Psychological Review*, 1935, 42, 186-204.
- [10] E. Schubert. "Measuring emotion continuously. Validity and reliability of the two-dimensional emotion-space", in *Australian Journal of Psychology*, 1999, 51(3), 154-165.
- [11] E. Bigand, S. Filipic, & P. Lalitte. "The time course of emotional responses to music", in *Annals of the New York Academy of Sciences*, 2005, 1060, 429-437.
- [12] L. Lu, D. Liu, and H.-J. Zhang. "Automatic mood detection and tracking of music audio signals", in *IEEE Transactions on Audio, Speech, and Language Processing*, 2006, 14(1):5–18.
- [13] Y.-H. Yang, Y.-C. Lin, Y.-F .Su, and H.-H. Chen. "A regression approach to music emotion recognition" in *IEEE Transactions on Audio, Speech and Language Processing*, 2008, 16(2):448–457.
- [14] T. Euroala, O. Lartillot, P. Toivaiainen. "Prediction of Multidimensional Emotional ratings in Music from Audio Using Multivariate Regression Models", in *Proc. Int. Conf. in Music Information Retrieval*, Kobe, 2009.
- [15] C. Laurier, M. Sordo, J. Serra, and P. Herrera, "Music mood representation from social tags," in *Proc. of the Int. Society for Music Information Conf.*, Kobe, 2009.
- [16] X. Hu, J. S. Downie, and A. F. Ehmann, "Lyric text mining in music mood classification," in *Proc. of the Int. Conf in Music Information Retrieval*, Kobe, Japan, 2009.
- [17] Y.E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, A. Speck and D. Turnbull. "Music emotion recognition: A state of the art approach", in *Proc. of the Int. Conf. in Music Information Retrieval*, Utrecht, 2010.
- [18] Lartillot, O., and P. Toivaiainen. "MIR in Matlab (II): A Toolbox for Musical Feature Extraction From Audio", *Proceedings of the International Conference on Music Information Retrieval*, Wien, Austria, 2007
- [19] J. Fornari, & T. Eerola. "Computer Music Modeling and Retrieval. Genesis of Meaning in Sound and Music", in *Lecture Notes in Computer Science*, chapter *The Pursuit of Happiness in Music: Retrieving Valence with Contextual Music Descriptors*, 2009, 5493, 119-133. Springer.

HUMANITIES, ART AND SCIENCE IN THE CONTEXT OF INTERACTIVE SONIC SYSTEMS – SOME CONSIDERATIONS ON A CUMBERSOME RELATIONSHIP

Pietro Polotti^{1,2}

¹Department of New Musical Technologies and Languages
Conservatory “Giuseppe Tartini”, Trieste, Italy

²Interaction Research Unity
IUAV University of Venice, Italy
pietro.polotti@conts.it

ABSTRACT

The theme of this conference, “creativity rethinks science” involves a radical epistemological challenge with respect to a classical view of science and it is an extremely hot topic of speculation within the scientific community, at least for what concerns computer sciences. In this paper, we propose some considerations about the role that artistic research could have within science, where science is meant in the wide sense of *knowledge*, including, thus, humanities as a one of the partners together with natural sciences. After a more general discussion focused mainly on the field of Information and Communication Technology (ICT), we will restrict the scope to the case of sound art involving new technologies and sound design for Human Computer Interaction (HCI), namely Sonic Interaction Design (SID). In our discussion, the concepts of design have a particular relevance, since they provide a connection between fields traditionally far away one from the other such as natural sciences, art, engineering and humanities. In the last part of the paper, we provide some examples about what we mean by doing artistic research guided by a design practice. We envisage this as one of the possible ways to make a dialogue between artistic research and scientific research more feasible at a methodological level.

1. INTRODUCTION

In this work, we discuss our point of view about the role of art as research activity in relation to scientific research. We conceive art as a way for discovery and definition of new perspectives of comprehension of reality. We think as well that artistic investigation and the “artistic means” can give an original contribution to knowledge and collaborate with science in terms of construction of evidences, study cases, counterexamples and criticisms to

Copyright: © 2011 Pietro Polotti. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

questioned ideas. In particular, we take into consideration the fields of HCI and SID by tackling the concept of interaction as an exemplary point of reference. In our opinion, interaction provides a privileged subject for a dialogue among humanities, art, science and technology. We also believe that the adoption of design methodologies in art could be an effective way to facilitate a dialogue between art and science in order to bring them closer together.

The paper has the following structure. In Section 2, we recall how computer sciences and HCI in particular are undergoing a process of inclusion of knowledge and ways of thinking that are not strictly based on scientific inductive methodologies. In Section 3, we consider this aspect for the particular case of the argumentative thinking as complementary to the deductive/inductive reasoning proper of natural sciences. Section 4 is devoted to the recovery of a temporal dimension in science and the consequent relativization of a visual paradigm of knowledge as opposed to an acoustic one. In Section 5, we propose some issues about the contribution that art could give to an overwhelming technology development. In Section 6, we envisage the integration of design methodologies in the artistic practice as a way of making artistic research more systematic and closer to a scientific research praxis. In Section 7, we expose some brief considerations about the modalities of a dialogue between artistic creativity and science. In Section 8, we concentrate on the particular case of sonic interactive arts and SID, and we briefly review some team works by the author with different artists and researchers in order to provide some examples of application of the points discussed in the paper. Section 9 is devoted to a final discussion, in which we draw our conclusions.

2. HCI EPISTEMOLOGICAL REVOLUTION

In order to face the new challenges stated by the ICT society, the necessity of promoting a wide scope interdisciplinary effort ranging from the humanities to the natural sciences is clearly emerging (see for example [1]). The epistemological revolution introduced by HCI [2] and Artificial Intelligence (AI) [3] provides examples in this

sense. When technology was employed only within the scope of the industrial revolution under the epistemological framework of natural sciences, the complexity of the human psychology could be almost neglected or oversimplified. On the contrary, with fast-evolving digital means and the computer “tool” undergoing what Brey calls the passage from an epistemic role to an ontic role [2] a conception and development of a “psychologically and anthropologically ergonomic” technology becomes necessary. In his paper, Brey argues that “contemporary computer systems perform two broad classes of functions: epistemic functions and ontic functions. Epistemic functions are what have traditionally been called information processing functions,” however “the computer is no longer just a cognitive device, it is now also a simulation device” and “the functional role of computer systems in their role of simulation devices may be termed ontic, because their role is to generate or represent objects and environments that form an addition to the physical world.” In other words, digital technology is not just a tool for knowledge but something rather related to being.

In this sense, it is also remarkable as the name of a discipline born in a purely computer science environment, such as HCI, is gradually changing its denomination with Interaction Design (ID), dropping the term “machine” and introducing a term that is not easy to define as “design” [4], [5]. Design can be thought of as a discipline, which is in between natural sciences and humanities, technology and art. As Stolterman says [6] “interaction design research has for some decades developed theoretical approaches, methods, tools, and techniques aimed at supporting interaction designers in their practice” and “many of them have intellectual roots in other academic areas, such as science, engineering, social science, humanities, and in the traditional art and design disciplines”.

In the debate within the HCI/ID community one of the questions that people ask themselves is if a quantitative validation of the results is always non-renounceable and even meaningful for the kind of studies and subjects involved by the discipline. For example, talking about GOMS (Goals, Operators, Methods, and Selection rules), a popular evaluation methodology in HCI, Rogers [7] asserts that “A problem with predictive models, ..., is that they can make predictions only about isolated, predictable behavior. Given that individuals often behave unpredictably and that their activities are shaped by unpredictable external demands, the outcome of a GOMS analysis can be only a rough approximation and may sometimes be inaccurate. Furthermore, many would argue that carrying out a simple user test, such as heuristic evaluation, can be a more effective approach and also require much less effort.” In general, if we need to rehabilitate qualitative and heuristic evaluation, it becomes meaningful to go beyond the strictly logic thinking peculiar of natural sciences and to take into consideration humanities research paradigms, as discussed in the next section.

3. ARGUMENTATIVE THINKING AND THE NEW RHETORIC

Heuristic thinking and a holistic approach mainly based on qualitative argumentation find more and more space in the scientific world beside a strictly logic, deductive and/or inductive thinking (see for example the foundational text by Perelman and Olbrechts-Tyteca [8] on the subject of argumentation theory and the work by Jekosch [9] for what concerns the SMC field specifically related to product sound design). The reductionistic paradigm inherent physics research, pursuing the definition of general and simple (simplistic?) models and the reproducibility of results as grant of objectivity, shows its relativity as soon as the object of study and its environment involve human factors either psychological, social or cultural. Examples of this crucial issue can be found in the dialectic relationships between neurosciences and the phenomenological approach in experimental psychology or between a sociological practice founded upon quantitative methods and ethnography, the former based on tests aiming at an objectivity of scientific character and the latter founded on the observation of the phenomena in their complexity (see for example [10] for what concerns the qualitative character of the ethnographic research). A research, thus, closer to the thinking of the humanities emerges as a need.

In an already mentioned paper [3], cogent argumentations in the direction of recovery of humanities techniques into the field of computer science were recently introduced. Specifically, the subject of the paper is the present role of rhetoric and argumentative reasoning in AI. The author recalls how “Computing, especially in artificial intelligence and multi-agent systems, has moved away from exclusive use of deductive logic and inductive reasoning and has now accepted argumentation as a method of modeling defeasible reasoning.”

In tune with these argumentations, we envisage the reinvention and adaptation of communication techniques developed within pure humanistic frameworks to the context of ID as a powerful strategy for the development of the discipline. We consider this as a coherent and promising research subject from a methodological point of view for both ID and SID, being the latter a subdiscipline of the former. In the context of ID, an effort of exploiting the nature of rhetoric as art of convincing can be found in an article by Grasso et al. [11]. In a recent paper [12], we argue that rhetoric can provide a breakthrough in the definition of sound design for Auditory Display (AD) and SID. In the second part of the paper, we present the results of a first case study concerning the evaluation of a rhetoric-based design procedure of a set of earcons, i.e. of short melodic fragments used as audio iconic representations [13]. The effectiveness of the design was assessed in terms of memorization of the earcon-function associations by groups of subjects. The outcomes of the experiment provided a first positive result encouraging a more general and extensive investigation about the utilization

of rhetorical techniques in support of sound design for AD.

The fields of AD and SID are good examples about what is happening in the HCI revolution. In those disciplines, the issue of defining guidelines in order to go beyond particular cases and solutions is a hot topic. A keen critic about the necessity of development of solid methodologies for the discipline appeared in a paper of some years ago by Barrass [14]. In another paper by the same author [15], the problem is faced from a perceptual point of view. However, perceptual aspects are only one of the facets of SID research. A more comprehensive work [16] appeared in a special issue of the International Journal of Human Computer Studies devoted to SID. In that paper, Frauenberger and Stockman propose to consider design pattern analysis as a point of reference for the discipline. Their article offers an overview of the available methodologies, and points out the lack of a unitary and robust framework for the discipline. In particular, the authors highlight how researchers in the SID field usually do not reveal the rationale of their design decisions. As an alternative, they introduce a method based on pattern mining in a context space. They are aware of the pro and con of a pattern-based approach, and propose to use context as an organizing substrate. In their opinion, this will provide designers facing new problems with useful reference patterns of already existing ID practices. The method promotes the growth of the discipline by means of a process of building upon previous knowledge. The approach presents strong similarities with one of the keystones of rhetoric that is the so-called *loci*. The *loci* are the result of a classification of the argumentation types according to specific criteria, and they provide the orator with a well-organized database of formulas, paradigms, examples and strategies that she or he can browse, select and employ in order to build her/his discourse and promote her/his theses.

We are strongly convinced as well that providing AD and SID with robust operational guidelines is of extreme importance for our forthcoming social life and environment. In fact, we can easily envisage a future, in which a multitude of technological devices will own expressive and listening capabilities in the frame of speech and non-speech audio communication. The acoustic scenario will include thousands of new artificial sounds that will pervade our everyday life, and consume the available "environmental acoustic band". In particular, we expect a huge proliferation of non-verbal sounds. Such an acoustical hypertrophy requires an adequate action aiming at defining ways to optimally exploit the communication potentialities of non-speech audio, while avoiding the degeneration of the acoustic environment into an overstuffed soundscape. Strategies for designing artificial sounds in a concise and effective way tackle both these aspects at once by optimizing the communication process, and reducing the sonic impact in terms of psychological and physiological fatiguing. The opposite of what an ambulance siren is in the context of alarm design. These ideas are compliant with an ergonomic design of technological

means, and tackle the more general issue of a sustainable technology.

4. TOWARDS A RECOVERY OF A TEMPORAL/SONIC DIMENSION OF THE WORLD

When dealing with sound, time is a crucial aspect. Sound is essentially a temporal phenomenon both from the physical point of view and from the perceptual one. In ID as a whole, dealing with temporal aspects means taking into consideration continuous interaction. This is another important aspect of the above discussed epistemological change of perspective.

In science in general, time has gained new insight during the last quarter of the 20th century, marking the end of the supremacy of a static and spatial view of the world by physics [17]. Classical physics is related to a visual and geometric conception of the world: images, as well as written words, allow the eye to gaze indefinitely and to move forth and back in an unified, homogeneous and divisible world, as if time were suspended [18]. On the contrary, time related phenomena as sounds are irreversible. Changing perspective from a visual to an acoustic one has radical consequences on the conception of the world, whereas its acoustic aspects represent the irreversibility of time that involves energy transformation, vitality, complexity, opposite to the ideas of reversibility, immobility, symmetry, simplicity of the classical visually-oriented world conception.

In this fast evolving cultural framework, embodiment and gesture are acquiring more and more importance in HCI (see, for example, the series of workshops on "Gesture in Embodied Communication and Human-Computer Interaction" [19]). Body and gesture are related not only to space, and, therefore, to the visual world, but also to movement in time, and they are intrinsically related to sound: sounds are produced by movements and interactions and constitute one of the modalities of their appearance.

A further consideration is that ID and interactive arts share common digital means. We think that this is a crucial aspect that makes the two fields more related one to the other than traditional arts and design. ID and interactive arts together provide a great opportunity of synergy between humanities, science and technology. We believe that SID, together with the design of new interfaces for music and sound art, constitutes a particular case of this potential synergy, and we share Bill Verplank et al.'s conviction, when they write: "We believe that the direct engagement in an expressive realm like music can generalize to a wide range of human-machine controllers." [20]. Even more, we are strongly convinced that the sound art and musical workbenches can be a privileged place, where it is possible to explore and develop paradigms for a subversion of the scheme of a mankind running after technological development into that of a technology recovered to its primary scope of means for human life, as discussed in the next sections.

5. AN ARTISTIC RESEARCH PERSPECTIVE FOR A SUSTAINABLE RELATIONSHIP WITH TECHNOLOGY

Along the 20th century, the philosophical debate was actively engaged into the comprehension and definition of technology in its socio-cultural implications in terms of modifications produced in our existence. The awareness of a technological development living of its own life as a consequence of deep cultural roots of the western world emerged through the century [21]. Technology is not any more reducible to the dimension of a tool or, more in general, a means. Rather, the primary goal of technology is the technological development itself: the mankind is somehow culturally compelled to develop and exploit everything technology allows/offers him. At the same time, the imagination cannot stand the power and the rate of technological production and innovation, but can only run after it. Technology progresses in a tautological way according to an inner logic, and the human goals and concerns can only adequate and draw consequences according to the latest technological advancements.

The question is if we can go beyond this perspective, and if we can still define goals starting from the state of the art of technology but independently from its inner, non-human logic. One could argue that since its first appearance in the second half of the 19th century, design in general can be seen as a huge effort in the sense of imposing creativity and human-concerns over technology by reflecting on and planning the industrial production.

The digital era emphasizes this uncontrollable technological growth, however at the same time, it opens chinks for placing again human aspects before technology. This seems, for example, the attitude and source of inspiration of the hacking approach to digital creativity for what concerns hardware or the open source philosophy for what concerns software: “anybody” can adapt and reinvent technology according to personal purposes. More in general, as discussed in Section 2, HCI puts technology explicitly in contact with the complexity of the human being that is with the non univocal nature of the human thoughts, emotions and behaviors. As a consequence, HCI becomes a natural laboratory for an epistemological revolution, and, at the same time, provides the playground of the great challenge for the development of a technology devoted to humans (and not the opposite). In other words, the goal is to go beyond the “computer metaphor and the related Cartesian mind-body dualism [that] have resulted in a fairly mechanical comprehension of the human being using a technical device” [22], in order to strive for a technology meaningful and sustainable for the human beings from many points of view: physical, psychological, social and environmental.

We believe that art can play the role of a laboratory for developing evidences and letting new perspectives of comprehension of the world emerging. A kind of research crosscut following the path of intuition and creativity instead of that of strictly logic thinking and providing knowledge from a different however valuable perspec-

tive. In particular, we believe that interactive arts are a fundamental actor for the development of a sustainable relationship with technology. They can constitute the workbench for a free experimentation of new ways of conceiving, employing, analyzing, interpreting, and criticizing technology in a complementary and synergic way with respect to ID.

Another aspect somehow related to art as investigation is that of art as didactic means. This is nothing new as art has always had a pedagogical role within societies. However is worthy to underline, how a portion of the artistic production since the 70's, i.e. the public art and body art (see for example the work by Dennis Oppenheim, and other artists of that period [23]), has its focus on the experience of the audience and its active participation. This idea can be found in many present-day interactive installations in form of public art, where a didactic-explorative valence of the artistic work emerges explicitly. The idea of participation in interactive arts strongly enhances the didactic valence of art.

In general, we have the impression that, when dealing with interactive contexts, the distinction between art and design tends to fading out. In our field, a pregnant case of being on the borderline between science, design and art is that of AD and SID. In particular, the former community that studies how to display information through the auditory channel by means of non verbal sounds considers itself as a scientific community in a pretty strict sense. The latter community is, on the contrary, more hybrid and multidisciplinary by its nature, in the same way as ID is. Furthermore, art is explicitly accepted among the contributors to the discipline [24]. On the other side, we believe that the interconnection between art and design can proceed also along the inverse direction that is a designerly approach can be fruitfully adopted in interactive arts (and art in general), as argued in the next section.

6. A DESIGN-ORIENTED ARTISTIC PARADIGM

SMC is a highly interdisciplinary field and a relatively new one [25]. The profile of the SMC researcher is not always easy to define from a disciplinary point of view. For example, the question if a person, who faces the challenge of creating a new interface or a new sonic/musical system, is necessarily supposed to be an electronic engineer or a computer scientist, is a cumbersome one [26]. Given the accessibility of the present technologies, we believe that this is not always a must. On the other side, when this type of work takes place within and for a musical and artistic domain, we think that the designer denomination would not be the most appropriate. Often, what we are in front of is the work of technologically skilled artists and musicians. At the same time, we pose ourselves another question: could design methodologies act as point of reference for an artist that works with digital technologies?

We think that making art inspired by a designerly approach, where ideas are followed by multiple and alterna-

tive realizations that can be compared in order to reveal the multifaceted and critical points of questioned ideas from different perspectives, can be a profitable strategy for art as well. Also, the principle of cyclic iterations, realization, evaluation and redefinition of the realization (or of the idea itself) according to the evaluation results could constitute a strong paradigm to establish an artistic practice structured and biased towards a systematic investigation of an idea. Third, the fundamental design principle of going through rapid sketching and/or realization of mock-ups provides a powerful operating praxis in front of the unavoidable rapidity of technological evolution. In fact, the impossibility of being anchored to standards in the sense of solid reference tools and methodologies, on the top of which to build something valuable and durable/repeatable because sufficiently independent from the particular technology state of the art is a severe drawback. Fourth, a design approach involves as a connate praxis working in team: the artwork becomes a product of a group of people jointly contributing to it and used to share ideas, plans and goals as in a research group.

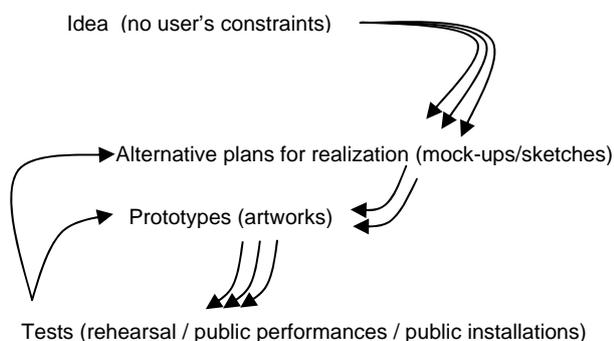


Figure 1. A design-oriented artistic practice.

The evolution rate is one of the main aspects that determine a lack of stable and long lasting technological means, through which to develop new traditions and solid cultural and stylistic peculiarities. In case of our discipline, this corresponds for example to the difficulty of creating new musical instruments that people are interested in exploring musically and technically for a significant while (let's say, at least some decades?). Design is thought of as guideline for a sustainable relationship with technology evolution, from one side, and methodological instrument for a structured artistic research, from the other. The framework of an artistic practice inspired by the procedures of design is represented in the diagram in Figure 1. In the artistic case, the starting idea is not the outcome of a brainstorming about consumer/stakeholder needs and/or requirements, but a free artistic investigation. In the same way, the validation/evaluation phase by means of user-tests is substituted by the rehearsals with the performers or by the observation of the reaction of the visitors to a system exhibited in public spaces (finally, a sort of qualitative user test), without any quantitative or logically rigorous evaluation as usually required in product design.

7. FOR A COLLABORATION BETWEEN ARTS AND SCIENCES

In this section, we figure different kind of forms of collaboration between arts and science. A preliminary issue is the necessity of working in a research group. Working in team on an artistic project in a design fashion can be very similar to the activity of a scientific research group. Nowadays, interactive artists are more and more trained to work with other people, contrarily to the model of artist as isolated creator. This means a shared capacity among artists for an integrated work with people having different disciplinary skills, methodologies and goals.

A first form of collaboration could regard art as a means for the development of the representation of the world in an epoch and a culture, by means of the display (in a multimedia sense) of results and concepts produced by sciences. In this sense, art can become a stimulus for scientific imagination and thinking and, vice-versa, science can be a source of inspiration for arts. The latter is actually a common practice. Moreover, collaboration between arts and sciences is nothing new in history, during epochs of minor disciplinary specialization. Anyhow, in this kind of relationships, the risk of being pretty superficial or even vague is high.

In the context of some disciplines, the relationship could be tighter and more effective. For example, psychology could collaborate with art at any level, from perception to cognition up to emotions, in order to provide data, study cases, alternatives, counterexamples, exceptions. At least since the Bauhaus experience, there are many contemporary artists that conceive their work as an explicit investigation of perception and cognition.

The case of computer science is also a key research area, able to involve disciplinary fields related to humanities too, as outlined in this paper. In particular, we discussed the AD and SID domains (see also [27]).

Concerning sound, it is possible to consider architecture, city and landscape planning as potential partners. We hope that as a consequence of the emergence of the soundscape as an important aspect of life quality, sound artist will be able to contribute to the development of a consolidated discipline of soundscape design.

In the next section, we briefly review some team works by the author and others as particular implementations of the principles previously discussed. The first work, the *Gamelunch*, is a sonically augmented dining table in between a public art work and a product design project. In 2007, it was presented as an interactive art installation at *Enaction in Arts* in Grenoble [28]. At the same time, it was developed in a design context and contextualized in the general theme of continuous sonic interaction within ID [29], providing, in our opinion, an example of strict relationship between themes and tools that can be developed both in artistic and design contexts.

The others results are related to the Elementary Gestalt for Gesture Sonification (EGGS) project [30]. The works are both in the form of public art installations and of interactive performances. All of them are inspired by a

common idea that is developed and questioned in different alternative realizations in the spirit of what depicted in Figure 1.

8. EXAMPLES FROM RECENT WORKS

In this section, we provide an overview of some team works by the author and other researchers and artists, as preliminary results about what discussed in the present paper.

The already mentioned *Gamelunch* was an interactive installation based on a dining table. The work aimed at investigating the enactive loop connecting action, sound and sensation. By means of various sensors, the continuous gesture interactions of a dining context were captured and transformed into energetically coherent data driving a set of physically-based sound synthesis algorithms, provided by the Sound Design Toolkit (SDT) package [31]. One of the key points of the installation was to raise the user attention and awareness about the importance of the sound response in any human physical action by means of unexpected and sometimes contradictory sonic feedbacks. The work was also intended as an investigation about the potentialities of sound as an augmenting element in interactive systems for everyday life, tackling aspects such as the sonic identity of materials (in an enhancing or contradicting fashion) and the alteration of the user proprioception during the action (in a prompting or retaining way).

Sound as means of gesture representation in the time domain combined with the principle of simplicity based on the use of elementary sonic and movement units are the leading ideas of another set of works related to the already mentioned EGGs project. In the spirit of Bauhaus, one of the principles we adopted is to consider straight and circular trajectories as elementary gestalts for gesture analysis and segmentation (see [31] for more details). Some of them were conceived as interactive installation in the form of public art as Visual Sonic Enaction (VSE) [32] and Sonic Walking (SW) [33], others as interactive performances as Swish 'n' Break (SnB) [34] and Body Jockey (BJ) [35].

VSE, is a multimodal and interactive installation that allows to generate audiovisual representations of one's gestural expressivity. The installation is usually presented as if it were an interactive graffiti painting system, where one can "listen" to her/his gesture. The visitors are encouraged to paint on a large wall by means of an "electric torch/spray can" controlling different graphic and sound processing algorithms. The sound elicits and guides the movements of the user and immerses she/him in a bodily-visual-auditive experience, providing gesture with a multimodal and continuous feedback. Indeed, sound plays the role of connective element of the three components of VSE. In fact, the EGGs principles of simplicity are applied to the visual domain as well. The final aim is not to paint. Rather, what appears on the wall or on the computer screen is a visualization of the expressivity of gesture guided and prompted by the sonic feedback. At the same time, in an enactive way, the visual feedback spurs

the user to modify and control her/his own gesture also according to different type of visualized graphic. A further subject of investigation offered by this kind of works and planned for the future is to question, if the definition of abstract (gestural) categories and of effective, however independent, mappings of gesture onto sounds and graphics generation will reveal unexpected cross-modal relations between visual and sonic structures.

In another public installation, Sonic Walking (SW), we moved the focus on gait expressiveness, therefore, shifting the focus from the upper part of the body to the lower part and taking into consideration gesture in an everyday context. Visitors had just to walk freely along a straight path in an ordinary indoor space, while their gait was sonified by means of ecological sounds related to nature. Similarly to VSE, the users were told to "listen" to her/his walk. Since natural soundscape sounds were employed, the didactic valence of SW regards as well a sort of soundscape listening training through the use of the body.

When working with a professional performer/dancer on a stage, as in SnB and BJ, the premises are different. Sound is meant as an effect of the choreographic gesture and a representation of her/his gestural expressiveness. EGGs becomes what we call a "choreophone": the performer/dancer does neither follow a musical piece, nor controls the execution of a musical piece, and not even generates music with her/his movement. Rather, (s)he listens to her/his gesture, enactively, modifying and controlling her/his performative action according to the produced sound [36]. The sounds, thus, are a representation of the movement, a sonic consequence and a continuous feedback, in no way external to the gesture itself. In this fashion, sound is intended as augmenting the proprioception of the performer.

SnB was presented in the context of the latest SMC conference. The employed sounds were retrieved from the Freesound Project [37]. Beside the specific composition process, the choreographic choices and the decision of using everyday-life sound samples, the EGGs principles were the main focus of our attention in the development of the work. The main guideline was how to map different sound sets to different movements/gestures categories on the basis of a elementary straight/circular-clockwise/circular-counterclockwise trajectory discrimination (see [30] for more details).

The same holds for the last work, BJ, presented at NIME 2011. The idea was to introduce embodiment in club culture and musical styles. The technical setup is the same as the one in SnB, but including also a video projection according to the same principles of VSE. The three performers act as if being in a DJ and VJ set. The dancer triggers and modulates sounds by mean of her body, while the laptop performers change sounds, graphics and mappings, following a predetermined score. With respect to SnB, the system can be applied to non-trained bodies that is non professional dancers according to what disco dancing is: a way of moving that is totally personal and has none of the established barriers of dance language.

9. CONCLUSIONS

In this paper, we have presented some considerations about artistic research in interactive contexts and its actual or potential synergies with scientific research. We have underlined as the necessity of a broader and more inclusive way of thinking is emerging within various disciplines related to ICT and to computer science. We discussed how this widened epistemological scope involves humanities and its methodologies as well. When in the study of some subject, the environment plays a fundamental and non-renounceable role, the reductionistic and modeling paradigms of natural sciences become inadequate, since they always involve decontextualization as a premise.

In this framework, a significant example is provided by the employment of rhetorical strategies in AI and HCI. We argue that the recovery of argumentative thinking in AI and HCI is coherent with the epistemological revolution discussed, for example, by Brey [2]. We have recalled some results in this sense concerning the case of AD and the design of earcons.

In order to foster an enlargement of the interdisciplinary scope to art as well, we proposed an integration of properly adapted design methodologies within the praxis of artistic research. The idea that investigation through the artistic means can be an interlocutor of scientific research is supported by some examples produced by the author in team with various artists and researchers, and concisely reviewed in Section 8. In those examples the aim was providing evidences about the potentialities of non verbal sounds in terms of representation and comprehension of reality focusing on the particular case of human gesture/action sonification. Such works can be viewed both from and artistic and an ID perspective.

The proposal of framing an artistic activity into a cyclic process involving evaluation and rethinking of an (artistic) idea in a systematic way appears to us as a potential way to open a renewed and profitable dialogue between artistic research and scientific research. Also, becoming comfortable with the development of mock-ups in interactive contexts, from one side, and with team work, from the other, seems to be a promising strategy to make interactive arts and ID close enough to engage fruitful relationships.

10. REFERENCES

- [1] J. Löwgren and E. Stolterman, *Thoughtful interaction design: A design perspective on information technology*. New York: MIT Press, 2004.
- [2] P. Brey, "The Epistemology and Ontology of Human-Computer Interaction," in *Minds and Machines*, 2005, Vol. 15(3-4), pp. 383-398.
- [3] D. Walton "Computational dialectic and rhetorical invention," in *AI & Society*, 2011, Vol. 26, pp. 3-17.
- [4] P. Ralph and Y. Wand, "A Proposal for a Formal Definition of the Design Concept," in *Design Requirements Engineering: A Ten-Year Perspective. Lecture Notes in Business Information Processing*, Springer Verlag Berlin Heidelberg, 2009, Volume 14, Part 2, pp. 103-136.
- [5] K. Krippendorff, *The Semantic Turn; A New Foundation for Design*. Boca Ratan, London, New York: Taylor & Francis CRC, 2006.
- [6] E. Stolterman, "The Nature of Design Practice and Implications for Interaction Design Research," in *International Journal of Design*, 2008, Vol. 2 No. 1, pp. 55-65.
- [7] Y. Rogers, "New Theoretical Approaches for Human-Computer Interaction," in *Annual Review of Information Science and Technology*, 2004, Vol. 38, pp. 87-143.
- [8] C. Perelman and L. Olbrechts-Tyteca, *The New Rhetoric, A Treatise on Argumentation*. University of Notre Dame Press, Indiana. 1969 (original edition in French. Presses Universitaires de France, 1958).
- [9] U. Jekosch, "Assigning Meaning to Sounds – Semiotics in the Context of Product-Sound Design," in J. Blauert, ed. *Communication Acoustics*, Berlin, Springer, 2004, pp.193-221.
- [10] J. Clifford and G. E. Marcus (Editors), *Writing Culture: The Poetics and Politics of Ethnography*, University of California Press, 1986.
- [11] F. Grasso, A.Cawsey, and R. Jones, "Dialectical argumentation to solve conflicts in advice giving: a case study in the promotion of healthy nutrition," *Int. Journal of Human Computer Studies*, 2000, Vol. 53 (6), pp. 1077-1115.
- [12] P. Polotti and G. Lemaitre, "Rhetorical strategies for sound design and auditory display: A case study," in *Int. Journal of Human-Computer Studies*, under review, submitted Jun. 2010.
- [13] M. Blattner, D. Sumikawa and R. Greenberg, "Earcons and icons: Their structure and common design principles," in *Human Computer Interaction*, 1989 Vol. 1 (4), pp. 11-44.
- [14] S. Barrass, "A comprehensive framework for auditory display: Comments on Barrass, icad1994," in *ACM Trans. Appl. Perception*, 2005, Vol. 2 (4), pp. 403-406.
- [15] S. Barrass, "A perceptual framework for the auditory display of scientific data," in *ACM Trans. Appl. Perception*, 2005, Vol. 2 (4), pp. 389-402.
- [16] C. Frauenberger and T. Stockman,. "Auditory display design: An investigation of a design pattern approach", in *International Journal of Human Computer Studies*, 2009, Vol. 67 (11), pp. 907-922.

- [17] I. Prigogine, *From Being to Becoming - Time and Complexity in the Physical Sciences*. W. H. Freeman and Co., New York, 1980.
- [18] M. McLuhan, *The Gutenberg Galaxy: The Making of Typographic Man*, University of Toronto Press, 1962.
- [19] S. Kopp and I. Wachsmuth (Eds.), *Gesture in Embodied Communication and Human-Computer Interaction*, LNAI 5934, Springer Verlag, Berlin Heidelberg, 2010.
- [20] B. Verplank, C. Sapp and M. Mathews, "A Course on Controllers," in *Proceedings of the 1st Workshop on New Interfaces for Musical Expression (NIME01)*, ACM SIGCHI, 2001.
- [21] U. Galimberti, "Man in the age of technology". In *Journal of Analytical Psychology*, 2009, Vol. 54, (1), pp. 3-17.
- [22] A. Pirhonen, "Gestures in Human-Computer Interaction – Just Another Modality?" In S. Kopp and I. Wachsmuth (Eds.), *Gesture in Embodied Communication and Human-Computer Interaction*, Springer Verlag Berlin Heidelberg, 2010, pp. 281-288.
- [23] R. L. Goldberg, *Performance Art. From Futurism to the Present*, Thames & Hudson, pp. 156-163, 2001.
- [24] <http://sid.soundobject.org>
- [25] The S2S² Consortium, *A Roadmap for Sound and Music Computing*, Version 1.0, 2007, ISBN: 978-9-08-118961-3.
- [26] <http://smc2010.smcnetwork.org/panel.htm> (Mar. 30, 2011).
- [27] G. Eckel and D. Pirrò, "On Artistic Research in the Context of the Project Embodied Generative Music", *Proceedings of the International Computer Music Conference*, Montreal, Canada, August, 2009, pp. 541-544.
- [28] S. Delle Monache and P. Polotti. "Gamelunch - the sonic dining table," interactive sound installation presented at *Enaction in Arts*, Grenoble, Nov. 2007 (<http://acroe.imag.fr/enactive07/gamelunch.php>).
- [29] D. Rocchesso, P. Polotti, S. Delle Monache, "Designing Continuous Sonic Interaction," in *International Journal of Design (IJD)*, 2009, Vol 3, No 3, pp. 13-25.
- [30] P. Polotti and M. Goïna, "EGGS in Action," to appear in the Proceedings of the 2011 International Conference on New Interfaces for Musical Expression (NIME-2011). Oslo, Norway, May 30 - Jun. 1, 2011.
- [31] S. Delle Monache, P. Polotti, and D. Rocchesso, "A Toolkit for Explorations in Sonic Interaction Design," in *Proceedings of Audiomostly '10*, Pitea, Sweden. Sept. 15-17, 2010, pp. 7-13.
- [32] <http://visualsonic.eu/vse.html> (Mar. 30, 2011).
- [33] <http://visualsonic.eu/sw.html> (Mar. 30, 2011).
- [34] M. Goïna, P. Polotti, and S. Taylor, "Swish & Break - Geschlagene-Natur," in *Concert around Freesound*, SMC 2010, 7th Sound and Music Computing Conference, Universitat Pompeu Fabra, Sala Polivalent, Barcelona, Spain, 22 July 2010.
- [35] S. Taylor, M. Goïna and P. Polotti, *Body Jockey (BJ)*, interactive performance to be presented at NIME 2011, Oslo, Norway, May 31, 2011.
- [36] http://visualsonic.eu/eggs_in_action.html (Mar. 30, 2011).
- [37] www.freesound.org (Mar. 30, 2011).

EXPLORING THE DESIGN SPACE: PROTOTYPING “THE THROAT V3” FOR THE ELEPHANT MAN OPERA

Ludvig Elblaus

KTH Royal Institute of Technology
elblaus@kth.se

Kjetil Falkenberg Hansen

KTH Royal Institute of Technology
kjetil@kth.se

Carl Unander-Scharin

University College of Opera
carl.unander-scharin@telia.com

ABSTRACT

Developing new technology for artistic practice requires other methods than classical problem solving. Some of the challenges involved in the development of new musical instruments have affinities to the realm of *wicked problems*. Wicked problems are hard to define and have many different solutions that are good or bad (not true or false). The body of possible solutions to a wicked problem can be called a *design space* and exploring that space must be the objective of a design process.

In this paper we present effective methods of iterative design and participatory design that we have used in a project developed in collaboration between the Royal Institute of Technology (KTH) and the University College of Opera, both in Stockholm. The methods are outlined, and examples are given of how they have been applied in specific situations.

The focus lies on prototyping and evaluation with user participation. By creating and acting out scenarios with the user, and thus asking the questions through a prototype and receiving the answers through practice and exploration, we removed the bottleneck represented by language and allowed communication beyond verbalizing. Doing this, even so-called tacit knowledge could be activated and brought into the development process.

1. INTRODUCTION

It is common practice when working with new musical instruments, new media art or other artistic practices that rely heavily on new technology, to work interdisciplinary. Engineers, technicians or instrument makers work together with artists, composers or musicians towards a common goal. We will here denote these groups as *developer* and *artist* respectively, well aware that this is an oversimplification.

The developers often know the technology well but have less insight into the intended context and usage than the initializing artist, while the artist may have less knowledge of the technology or other priorities. Many projects are therefore close collaborations between these professionals (and commonly, the cross-fertilization is strong, thus smearing out the borders between the roles), as have been reported

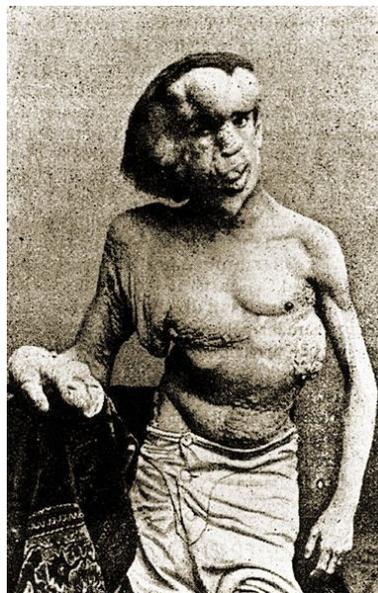


Figure 1. Portrait of Joseph Merrick (1862–1890), also known as *The Elephant Man*.

in for instance the Sound and Music Conference [1] and the New Interfaces for Musical Expression conference proceedings [2].

Communication during the development process is very important. The more of the artist’s relevant information that can be available to the developer, the higher the probability is of the project being successful. This knowledge is however not always easy to communicate. It can be embedded in practice, so-called tacit knowledge that can be hard to verbalize. The communication can also be hindered by questions that the developer fails to ask and details the artist fails to recognize as being important [3].

When dealing with technology intended to create or enhance experiences both from a performer and an audience perspective other practices than traditional hardware–software development models can be used to open the necessary channels of communication. During recent work with prototype development for a new musical instrument, ideas from the fields of interaction design and participatory design were used to bridge the artistic and technological divide.

In this article these ideas will be briefly outlined as they are theorized within their respective fields and their application described in three specific cases. Furthermore, the divide between the artistic and the technological is bridged

by a hands-on approach using prototyping and involvement of both artist and developer in a team. Instead of only using the artist's expertise as a starting point and to evaluate the result of the process, the artist has been immersed in the development process so all available resources could be exploited in the project.

1.1 The project

In this project, Elblaus was approached by Unander-Scharin to develop a gesture controlled signal processing device for stage use. The project, called "The Throat v3" is still ongoing by the time of writing. The Throat v3 is to be used in an opera entitled "The Elephant Man", currently being composed by Unander-Scharin. Although the roles as artist and developer were conditioned from the start, the developer has documented artistic experiences, and the artist has documented experiences in development. Therefore, interchange of experiences was well supported and the use of participatory design patterns was a natural direction in the development.

Frederick Treves notes in his autobiography [4] regarding Joseph Merrick, popularly known as *The Elephant Man*, that "the fact that his face was incapable of expression", and "his attitude that of one whose mind was void of all emotions" (see Figure 1). When conceptualizing The Throat v3, these aspects were thoroughly considered. A microphone was used to capture the singers' portrayal of the limited vocal sounds that were possibly produceable by Joseph Merrick, due to his severe physical disability. The smaller components of speech and singing, which are normally inaudible in applied operatic (italianate) technique, could be utilized to create soundscapes and accompaniment for arias.

A suggested term for this practice could be "deformed vocal technique"—as opposed to "extended vocal technique". Keywords suggested by the artist to the developer, when designing the sound processing modules were: mucus, inflammation, coughing etc.

The prototype uses a computer with an audio interface, a microphone, an Arduino microcontroller [5], and pressure and flex sensors. The sensors are varistors so some simple voltage divider circuitry is needed to let the microcontroller read the varying resistance corresponding to the measured pressure or bending of the sensors.

The software, written in the SuperCollider language [6], is a modular environment that offers a wide assortment of processing types that can be modified, combined and collected into scenes which in turn can be arranged into sequences. This way longer structures of scenes can be prepared for a performance so that a performer can focus on stepping through the structure and modify parameters expressively in each scene.

A system of morph groups is available, where one sensor can be mapped to any number of signal processing parameters on a per scene basis, allowing both *one-to-one* and *one-to-many* mappings to be constructed [7]. Thus, scenes can both sound very differently and also offer individual types of interaction.

2. THEORY

2.1 Wicked problems

Horst Rittel and Melvin Webber [8] describe what they call *wicked problems* as problems that defy the standard problem solving method. Rittel and Webber were concerned with complex large scale problems such as public policy, but the reasoning behind wicked problems also motivates why some tasks in general benefit more from a design methodology than from problem solving.

Rittel and Webber formulated a set of distinguishing properties that showed how a wicked problem differs from what they call *tame* or *benign* ones. The theory is quite extensive, but in short, there are some properties explaining why design is not problem solving. In general wicked problems

- have no definitive formulation,
- have no stopping rule,
- have good-or-bad (not true-or-false) solutions,
- have no ultimate tests for solutions,
- have no finite number of potential solutions, and no defined set of permissible operations.

The last distinguishing property is very important for the process of dealing with wicked problems. Firstly, it states that the set of potential solutions is not known explicitly, which means that we can never try all the possible solutions to find the best one. Secondly, it states that each solution might contain any operation or element, and that we can never go through all the combinations of a defined set of operations since that set is not known.

2.2 The Design Space

The design space is the sum of all possible solutions to a design problem. The question is how to approach this space and how the design process should be structured with this in mind to produce good results.

The design space will never be fully known or fully understood but at the same time knowledge about the design space is needed to evaluate the solutions that are discovered during the design process. Therefore the goal of the design process can never be to fully define the design space but to get as much knowledge of it as possible so as to find the best possible solution given the constraints of the process itself, e.g. budget and time.

Accepting the process implied by the many different solutions in the design space means that the design process will be more of a gradually narrowing search than a journey to a predefined goal. It is clear that several intermediary solutions must be explored before the process is finished and that these solutions must be evaluated in some way. Every solution that the development process produce will chart a small subset of the design space so it is by putting forward solution suggestions and testing them that information about the design space is discovered.

Notice that the goal from the beginning is to find *several* solutions to the problem and not to first try figure out the

optimal solution. For such work, prototypes will be used to explore and map the design space, and each prototype will generate knowledge about the design space. By making prototypes and mock-ups of the proposed solutions, they can be evaluated and discussed from an interaction perspective and not just a theoretical one.

3. METHOD

3.1 Participatory Design

Knowledge of the product's intended user and the context can be acquired in many different ways. Combining methods that include observational field studies, interviews with users or other stakeholders, statistics, and surveys, lead to a better understanding of user needs and practices [9]. These methods can be problematic as they rely on intermediary observers or mediums like language and statistics, necessitating interpretation. Personal background, preformed views, and prejudices color the observers interpretation [10]. There can also be a discrepancy between what the respondents think they do, and what they actually do [11]. Many experiences are also very hard to accurately describe across modalities. For the subtle experiences of musical instruments, this is very much true.

Another approach is to invite the user to partake in the development process to directly access user response and feed that back into the development. Directly involving the user in the process can be challenging, but it will provide an abundance of information that is relevant since the information springs from the interaction between user and process [12].

With user participation, the information one gets is not filtered by the questions one ask, i.e. answers to questions that are never verbalized or thought to be relevant by the developers can still be given. Similarly, answers that can not be verbalized, so-called tacit knowledge, can still be used in the development process by letting the user participate and show practices, act out scenarios or by other means communicate what is hard or impossible to reduce to words [3].

Leman [13] uses the term *embodied cognition* to describe how the body is intimately engaged both musical performance and perception. Getting participants involved in the design process in a physical way is therefore very important to get an understanding of all the mechanisms involved in a musical context. These activities must have a solid support in the design philosophy used.

3.2 Iterative Design

Iterative design is a prototype-driven way of structuring the development process that is well suited to the needs of participatory design, as well as exploration of the design space.

It is a well established development method, see for instance Gould and Lewis [14] who focus on the designer's perspective or Nielsen [15] who discusses usability from a practical economic perspective.

Iterative design uses a cyclical work flow that for each cycle further refines the design. This model is a good match

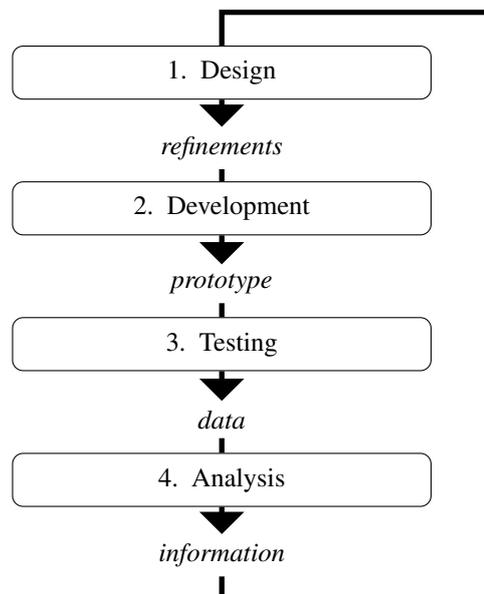


Figure 2. The cyclical flow of the iterative design process.

for participatory design, where maximizing the user involvement and quick reactions to user input are crucial. As soon as a first sketch or idea is formed the cyclical iteration can begin.

At its simplest, a design cycle contains four stages, as shown in Figure 2. First, a design stage where refinements to the design, based on available information, are proposed.

Second is a development stage where the design changes are carried out, which is the only stage where user involvement can be unnecessary. If hi-fi prototypes are to be constructed or if the product is nearing completion, user may have very little to contribute.

Third is a testing stage where it is vital to involve the users, to learn from the prototype. All cycles involve testing, a central requirement of participatory design.

Fourth is an analysis or evaluation stage where the data collected during testing is processed and refined into usable information. This is an important step as the experiences from the testing stage can be ambiguous and difficult to interpret. The information produced in this stage is used as input to the next cycle, as basis for the decisions taken in the next cycle's design stage.

3.3 Prototyping

When working with technology that is used in an artistic context developers are often in need of information on diffuse, hard to measure qualities regarding user experience. Whether something works or not in a technical sense is easy to determine but how the user experiences the interaction is a very different matter [16].

Language might be a bottleneck in communicating these kinds of questions and their corresponding answers. Simply asking someone how they would act in a certain situation might yield very different answers compared to actually observing them experience that situation and act in it [11]. This is why asking the question through a prototype

and receiving the answer through practice and exploration can be so valuable to a design process.

A prototype from a user interaction perspective is anything that can provide information on an interaction scenario. As Westerlund has shown [3], the prototype does not have to be an advanced piece of technology, but it must be able to successfully put the user in the desired scenario. A prototype does not have to be functioning on its own, it does not even have to be built with the same material as the intended product of the design process, as long as it answers a question or provides information through real or simulated interaction (for instance, a Wizard of Oz experiment).

Another aspect to consider when prototyping is to what extent the prototype is to evolve into the following generation and to what extent it will be thrown away. This is a dichotomy known as Evolutionary / Throwaway prototyping. When creating artworks, it is common practice to let the artifact evolve from prototype to a finished work of art. However, in iterative design, the use of a prototype that will be abandoned is common.

4. USE IN DEVELOPMENT OF MUSICAL INSTRUMENTS

Interaction design has emerged to fill a function where earlier methods have been less effective. The measurable and quantifiable aspects of design and development such as ergonomics and efficiency are all captured in more traditional development processes [17].

The field of new musical instruments and other artistic uses of technology are completely saturated with these very types of problems and questions of soft values and sometimes even subliminal experiences [18]. How does the performer experience her instrument? How is the instrument perceived by the audience? Is the instrument experienced as expressive and if not why?

Using the well researched tools of interaction design and other similar models can not only help to find these questions but also provide a method to learn from them and work with them. The idea of the design space is a good metaphor for projects that are not searching for an ideal solution to a well defined problem, but rather exploring a myriad of possible designs that can only be evaluated aesthetically.

4.1 Limits of user participation

With technology intended for an artistic context, one might assume that getting the technology to work and to make it a work of art is two separate tasks. This leads to the assumption that the artist primarily has artistic concerns, and the developer mainly technical. In reality, the roles are rarely so explicit.

A definition of purely technical aspects is to view them as black boxes. The input and output from the black box might have aesthetic ramifications but the inside workings are obscured from the user and indeed the system as a whole. Given that the same input leads to the same output, the mechanics inside are bereft of artistic relevance

Case 1: The Experimental Environment

| | |
|----------------------------|--|
| Participation | Incorporation of artistic vision |
| Prototyping | Non-verbal communication in the form of concrete sound exploration |
| Software-modularity | Reduction of developer bias |

Case 2: Lo-fi sensor workshop

| | |
|---------------------------------|---|
| Participation | Activation of tacit knowledge of practice |
| Low-fidelity prototyping | Unhindered exploration |
| Workshop | Open form that supports experimentation |

Case 3: Concert test

| | |
|---------------------------------|--|
| Wizard of Oz-prototyping | Testing without fully functional prototype |
| Context exploration | Simulation of the intended final context |
| Participation | Use of the composer's stage experience |

Case 4: External artist

| | |
|---|--|
| Studio test with external artist | Testing fully working prototype |
| Simplified context | Elimination of audience interaction |
| Externalized cognition | To explore the recontextualization of The Throat |

Table 1. Methods and goals for each of the case studies.

for the system.

Elements that fit that description can be handled with regular problem solving, user participation is not necessary and aesthetics can be disregarded. These black boxes are the only elements that the developer can design alone, leaving out the artist. It can be mutually beneficial to conserve responsibility for the technical details to the most knowledgeable party, especially when for instance the artist could direct efforts where it is more needed.

5. THE THROAT V3

For *The Throat v3* participatory design was used extensively. This was helped by the fact that the developer in this case had an artistic background and the artist had already developed early versions of The Throat.

However, the developer had however no previous experience of working with opera or singing voice, while the artist was unfamiliar with SuperCollider and could therefore not partake in the source code. Thus, the situation re-

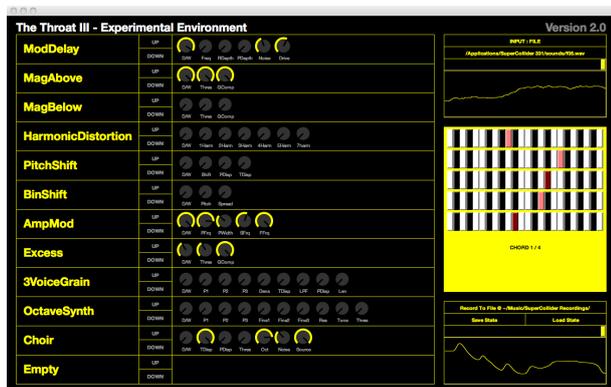


Figure 3. The Throat Experimental Environment is a modular signal processor written in the SuperCollider programming language. It offers a wide array of effects that can be used in real time.

sembled that of a traditional divide between project roles, and although in practice the participants were experienced in both roles, many details still needed to be shared and communicated. The design methods used needed to supply the tools for this communication to take place.

In the following, we outline how applied design methodology corresponding to stages in the project exemplifies the use of participatory design method, transmission of tacit knowledge through workshops, and partially working prototypes used with a Wizard of Oz approach. A summary of the methods and goals used in each case can be seen in Table 1.

5.1 Case 1: The Experimental Environment

When the foundations of the signal processing in the prototype were to be laid out, an approach was needed to ensure that no design decisions with artistic ramifications were taken without the artist’s involvement. For this purpose an experimental environment was constructed. It contained modular building blocks of signal processing. The processing performed could be controlled by a set of parameters unique to each module. The signal chain could use any selection of modules in any order. The graphical user interface is shown in Figure 3.

While it fell on the developer to code the actual software, the preliminary selection of processing types could be agreed on through discussions with the artist. Here, language was no bottleneck since well-known unambiguous terms exist for operations in signal processing (such as *ring modulation* or *low pass filter*).

Different kinds of timbres achievable through these techniques lack a well defined terminology. Even within a specific vocal tradition, terminology that transcends the technical aspects of vocal pedagogy can be ambiguous [19]. Precisely describing the desired timbres of singing through complex, layered signal processing would have led to unusable specifications. Instead, to give some guidance, a scenario was written by the artist to deepen the developers’ understanding of the intended use of the prototype.

In working with the environment all or a subset of the



Figure 4. Paper replicas of sensors used in a workshop. The positions of the replicas were found through experimentation, unhindered by risk of damage to the actual sensors.

available modules were connected in series. An audio signal was sent through this chain of modules and the result could be listened to in realtime. The incoming audio signal could be read from an audio file or taken from an audio interface.

The parameters of the signal processing of each module were available for manipulation. The mapping of the parameters and the choice of which parameters to offer were done to offer too much rather than too little. This way, the developer imposed as little artistic influence as possible, while leaving the artist room for creativity.

The artist explored this environment and saved noteworthy configurations, whether they were aesthetically pleasing or simply undesirable. It was not required to test the finer details of the signal processing together at that stage. The saved configurations were also practical for expressing aesthetic preference without the limitations and interpretations of language. It was simply easier to show than to tell.

The modular, open-ended concept worked so well that it was carried over into the later prototypes. It was a request from the artist to keep all parameters available from the experimental environment—to be able to return to all possible combinations of processing that had been used during the development.

5.2 Case 2: Lo-fi sensor workshop

One of the most important design decisions was how to position, attach and interpret the gesture-reading sensors placed on the hand of the performer. After discussions on different placements the need for testing the ideas in practice arose. Using the actual sensors would have been restricting and potentially expensive, as care would have to be taken not to damage or stress the sensors in any way.

To be able to freely explore the possibilities, paper replicas of the sensors were constructed. The replicas were made to have the same weight and bending characteristics as the actual sensors. The replicas attached to the hand can be seen in Figure 4.

Soon, a wealth of the artist's tacit knowledge of stage work was uncovered, catalyzed by the experience of the different sensor placements. Different placements affected the hand's stiffness in different ways, having a significant effect on the artist's ability to act on stage, that had not been anticipated.

In the conceptual work with the sensor placements focus had been on function and having as much control as possible. Considering that the performer in the opera would wear an elaborate mask to resemble the elephant man, including an abnormally large arm that would hide the sensors, the strain of the sensors seemed small in comparison. This mistake should possibly, in hindsight, have been noticed and avoided, but in the context surrounding the discussions there were too many other factors that received attention.

Here prototyping worked as a safeguard directing focus to an aspect that might otherwise have been overlooked. It turned out to be relevant to the development process and it surfaced naturally through a hands-on workshop with lo-fi prototypes. Simple discussions had not been able to come to the same important conclusions, and neither would a test with the developer using paper replicas have.

When preparing for the workshop, the developer had devised a set of hand gestures that could be used with the sensors. It became clear however that some hand positions that were trivial to the developer were uncomfortable for the artist, and vice versa. This is naturally especially important if the design is addressing more than one performer.

5.3 Case 3: Concert test

At one point in the development process an opportunity to perform an informal concert was presented. While the prototype at that point was not fully functional there were still many things that could be learned by testing it live, so the work focused momentarily on preparing a performance-ready version.

An attempt was made to integrate all the parts of the system, but the result, while functional, did not seem stable enough to meet the demands of a performance however informal. The concern was the connection between hardware and software that failed at one point and the time that was left before the concert did not allow for debugging to find the problem. Not knowing the cause of the failure, if it would happen again and if so, how often and under what circumstances, the risk of the prototype failing mid-performance could not be assessed, and so a fully-functional prototype was unattainable.

Next, a decision was made to only test a part of the system and to simulate the rest of the functionality in a Wizard of Oz approach. The software was used to process the artist's voice but the hardware control was substituted by an additional person on stage. The artist's hand gestures were not represented by sensor readings, but simulated by the person sitting directly by the computer running the software, watching for predefined hand cues. For this purpose a simple interface for manual control was added to the software and used for the performance.



Figure 5. A singer performing an aria using gesturally controlled signal processing in the final test in the prototyping process of designing the Throat V3.

Much was learned from that test and canceling the performance because of prototype failure or instability would have been a missed opportunity.

5.4 Case 4: Test with external artist

The development process was concluded with a test where a singer, external to the process, tested the prototype in a studio environment. Material from the upcoming opera was used, and the test therefore provided an understanding of how the prototype would function in the artistic process in which it was to be used. The artist could explore the process of preparing signal processing suited to the set material, testing both the usability of the software as well as the possible artistic expressions that could be achieved.

The singer had little problem moving while wearing the prototype and immediately began incorporating the control-gestures in larger gestures, masking their true function and making them a part of a larger stage presence. The singer, using the prototype can be seen in Figure 5. This provided important information about the prototypes ability for interaction and that the interaction itself could be integrated into operatic stage practice.

The singer performing with the prototype can be seen in Figure 5. The test was also filmed and an excerpt is available online.¹

6. CONCLUSIONS

Learning from the process of developing The Throat v3, it is clear that other tools than problem solving are needed to work with the complexities of constructing prototypes of musical instruments. Methods for design and development exist that are specialized in dealing with the unquantifiable values of subjective user experience. These are commonplace in many fields of commercial design, and development and should be so also in the practice of development

¹ <http://www.electronic-opera.com/node/774>

of new musical instruments and other similar academic and artistic projects.

Involving the users in the design and development process is beneficial and by creating a context where user participation is maximized and users are empowered by short development cycles and prototype-driven test based design, the users can become a great resource to the process. In a project with artistic goals, only that which can be considered as a technological black box can be developed without an artistic perspective.

The users provide not only what they can verbalize. Involving the users with hands-on, open exploration brings tacit knowledge and practice to light. Exploring situations and scenarios to learn together with the users can be much more revealing than asking specific questions: Sometimes the important questions are answered without ever being asked.

Acknowledgments

A special thanks to the acclaimed opera tenor Håkan Stårkenberg for a great performance with the prototype in the final test.

7. REFERENCES

- [1] Sound and Music Computing Conference, <http://smc2010.smcnetwork.org/>.
- [2] The International Conference on New Interfaces for Musical Expression, <http://www.nime.org/>.
- [3] B. Westerlund, "Design space exploration - cooperative creation of proposals for desired interactions with future artifacts," Ph.D. dissertation, Royal Institute of Technology (KTH), 2009.
- [4] F. Treves, *The Elephant Man and Other Reminiscences*, 1923.
- [5] Arduino, <http://www.arduino.cc>.
- [6] SuperCollider, <http://supercollider.sourceforge.net>.
- [7] A. Hunt and M. Wanderley, "Mapping performer parameters to synthesis engines," *Org. Sound*, vol. 7, pp. 97–108, August 2002.
- [8] H. Rittel and M. Webber, "Dilemmas in a general theory of planning," *Policy Sciences*, no. 4, 1973.
- [9] C. Bornand, A. Camurri, G. Castellano, S. Catheline, A. Crevoisier, E. Roesch, K. Scherer, and G. Volpe, "Usability evaluation and comparison of prototypes of tangible acoustic interfaces," in *Proceedings of ENACTIVE05*, 2005.
- [10] S. Lindquist, "The researchers role at stake the meeting between the objective researcher and the subjective individual," *CID-307 Technical report CID/KTH*, 2005.
- [11] J. Singer, T. Lethbridge, N. Vinson, and N. Anquetil, "An examination of software engineering work practices," in *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research*, ser. CASCON '97. IBM Press, 1997, pp. 21–.
- [12] S. Bødker and Y. Sundblad, "Usability and interaction design - new challenges for the scandinavian tradition," in *Behaviour and Information Technology, Volume 27, Number 4*, 2008, pp. 293–300.
- [13] M. Leman, *Embodied Music Cognition and Mediation Technology*. The MIT Press, 2007.
- [14] J. Gould and C. Lewis, "Designing for usability: key principles and what designers think," *Commun. ACM*, vol. 28, pp. 300–311, March 1985. [Online]. Available: <http://doi.acm.org/10.1145/3166.3170>
- [15] J. Nielsen, *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
- [16] E. McNutt, "Performing electroacoustic music: a wider view of interactivity," *Organised Sound*, vol. 8, 2003.
- [17] S. Jordà, "Instruments and Players: Some Thoughts on Digital Lutherie," *Journal of New Music Research*, vol. 33, no. 3, pp. 321–341, 2004.
- [18] D. Overholt, "The musical interface technology design space," *Organised Sound*, vol. 14, 2009.
- [19] D. Prem and R. Parncutt, "The timbre vocabulary of professional female jazz vocalists," in *International Symposium on Performance Science*, 2007, pp. 347–352.

MARCO STROPPA'S COMPOSITIONAL PROCESS AND SCIENTIFIC KNOWLEDGE BETWEEN 1980-1991

Vincent Tiffon

Université de Lille-Nord de France (UDL3-CEAC) /
Laboratoire STMS (Ircam-CNRS-UPMC)
vincent.tiffon@univ-lille3.fr

Noémie Sprenger-Ohana

Université de Lille-Nord de France (UDL3-CEAC) /
Laboratoire STMS (Ircam-CNRS-UPMC)
sprenger@ircam.fr

ABSTRACT

The purpose of this paper is to show the creative relationship that can be established between scientific knowledge and musical innovation, through the example of Marco Stroppa's work performed between 1980 and 1991 in three specific places: Padova CSC (and the Conservatory of Venice), Ircam (Paris) and MIT (USA).

The following methodological tools allow to understand the links between Stroppa's technico-scientific innovation, and musical invention: an analysis of his training years from 1980 to 1983 and of the main sources of cognitive models; a genetic study of the work *Traiettoria* (1982-1988), that is, the systematic study of traces, sketches, drafts, *computer jotters* and other genetic documents; written work published by Stroppa between 1983 and 1991; multiple interviews with the composer and witnesses of the period; a partial reconstitution under *Open-Music (OMChroma workspace)* of the electronic part initially performed under *Music V*.

In fact, *Traiettoria* constitutes what can be labelled a laboratory of Marco Stroppa's "workshop of composition".

1. INTRODUCTION

Marco Stroppa's musical composition process can be traced by means of a genetic-type inquiry relative to the sketches and other initial documents of the work *Traiettoria* (1982-1988), a piece for piano and computer generated sounds, which is the essential moment of this art-and-science articulation. The eighties were the time when this dual formation crystallized into a spirit of invention framed by the dual reference to scientific and musical worlds.

After highlighting the advantages and drawbacks of the bivalence of Marco Stroppa's competence, this discussion will describe how suitable were the conditions for him to assimilate scientific knowledge at Padova CSC (Italy), Ircam (France) and MIT (USA). These conditions led Stroppa to define a "workshop of composition" comprising, first the physical location of his work, second, the technological tools, and third, the experimental methods

and intellectual scientific surroundings. For each of these parts of his workshop, it is worth describing how the worlds of scientific research and musical invention can enrich each other.

2. STROPPA, BOTH COMPOSER AND SCIENTIST

Marco Stroppa's personality provides a perfect basis for the study of the interactive relationship between scientific and art research. This interaction stands out in Stroppa's work, in particular in *Traiettoria* (1982-1988)¹, a work written for piano and computer generated sounds. As regards the man himself, beyond his work, and following the example of some famous predecessors such as John Chowning or Jean-Claude Risset, Stroppa is often considered as a scientist by the artistic community, such his competence has been proven in the field. This artistic researcher is thus equally a skilled science and technology researcher.

Before approaching the genealogy of this dual competence, it is worth pointing out the remarkable advantage brought by this bivalence for a creator. The conclusion fed back from composition-related problems and solutions afforded by scientific and technical means led young Stroppa to develop his very affirmative and powerful creativeness. Although *Traiettoria* is only his second piece of work, it is already classed as a major work of the mixed repertoire. On the other hand, bivalence may have some drawbacks. The relations nurtured by artists and scientists may produce profitable accidents which cannot happen if the two types of competences are possessed by the same person. The discussions and requests of musical assistants or producers of computer-generated music and other scientific researchers may produce new ideas likely to generate compositional innovations. Unlike Stroppa's predecessors Risset and Chowning who made a dual career, Marco Stroppa finally chose to take the privileged road of composition, without however leaving aside scientific aspects (through reading and daily contacts with engineers, computer staff and researchers).

Copyright: © 2011 Tiffon et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

¹ Scores published by Ricordi [1] and recording published by Wergo and Stradivarius [2].

3. IDEAL TRAINING CONDITIONS

3.1. Training sites: CSC and Conservatories (Italy)

The dual competence of Marco Stroppa is primarily linked to the institutional context of the articulation between the Conservatory of Venice and the University of Padova CSC. Toward the final years of his training as a highly confirmed pianist and composer at the conservatories of Milan and Verona, Stroppa simultaneously followed musical training at the conservatory of Venice and at Padova CSC (1980-1983), where Alvisé Vidolin helped him acquire scientific knowledge related to the domain of sounds: signal processing, acoustics, psychoacoustics, sound synthesis, fundamental computer science, etc. On the occasion of interviews made with Stroppa, he declared that this simultaneous training led him to build a composition method which does not differentiate work on notes and work on sounds.

When his training was partially completed, Marco Stroppa embarked on a project with the aim of entering a music commission of the RAI. However, since he was under no obligation to enter the examination, the project was in fact stemming from his personal determination rather than being a mere answer to a proposal. This was indeed the first of Stroppa's personal projects, since *Metabolai*, composed in 1982 and the first of his works to be classified, was actually written to validate his composition studies. Thus, this scientific/technical and musical dual training led the way to the birth of the idea contained in the *Tre Studi per un progetto* which eventually became known as the three movements of the work named *Traiettoria*. Stroppa's initial intention was to work on the microscopic size of the sounds, as a follow-up to the scientific approach of his studies at the CSC. Stroppa's reflection focused first on a work for *pianoforte* only, and on the resonance capability of the natural acoustic characteristics of the instrument. He soon began to fully use his dual competence by writing a "synthetic orchestra" ("orchestre synthétique"), an expression by which he defined piano accompaniment by computer-generated synthetic sounds [3].

3.2 Laboratories and institutes: Ircam (Paris, France) and MIT (Cambridge, Mass., USA)

Pierre Boulez invited Stroppa to enter Ircam at the beginning of 1983 where his first intuition was soon confirmed. He remained until the end of 1984, further enhancing his relationship with the scientists. At Ircam, institute whose activity is precisely centered on the articulation between art (music) and science (acoustics), Stroppa could fully validate his first attempts at writing the first two movements of his project. He could thus refine his reflection, in particular through the decisive discovery of McAdams work [4, 5]. In addition to the fusion/separation of the aural stream², today, *a posteriori*, whenever Stroppa mentions those notions which were noteworthy to him, he also cites for example the "timbral space" [6]. These notions were under study at the time and they were

² Subject studied in part 4.3.3.

transmitted by means of a high number of characteristic sound samples which are unforgettable for Stroppa.

During his presence at MIT (Cambridge, Mass., USA) between 1984 and 1986, Stroppa followed high-level courses in domains rather far removed from the music field, but which were to boost his fertile imagination: computer music and cognitive psychology, and also artificial intelligence, structured programming, expert systems, etc. He was inspired with musically oriented reflections stemming from his knowledge of professorial work such as *Cognition and categorization* [7] which induced him to imagine his Musical Information Organisms [8].

These various places, where it was easier to meet scientists and to work on machines, constituted a part of the "workshop of composition" in gestation at the very beginning of Marco Stroppa's composer career.

4. TRAIETTORIA COMPOSITION WORKSHOP

In order to explain how the scientific knowledge and the musical creation are articulated, it is necessary to minutely describe the composer's workshop. This word "workshop" must be understood as the work environment in a wider sense than is the case for artistic crafts or painters' studios. For clarity's sake, the following brief typology describes in three points the composer's workshop as designed by Marco Stroppa for *Traiettoria*.

4.1 Physical location

As mentioned above, Stroppa designed his work both theoretically and practically in three different places: at Padova CSC for the premises, and also for the generation of the synthetic sounds (*Music V³* and *ICMS⁴*) of the different movements; at Paris Ircam for the composition of the first two movements, *Deviata and Dialoghi* (creating score for piano, and *computer jotters*); and finally at Cambridge (Mass.) MIT for the continuation of the project with *Contrasti* (again creating a score for piano, and *computer jotters*). Cross-checking his notebooks and sketches with the archives of the different research centers allow to trace his itinerary.

At that time, composers led a nomadic existence to find research centers which were the only places where high-performance computers could be found. This also provided composers an opportunity to meet researchers in particular at the CSC and the LIMB [9]. Stroppa always generated his synthetic sounds at the CSC. Therefore, Stroppa did not go to Ircam or MIT to use computers but indeed to look for places at the heart of research combining music and science. These encounters with the researchers and their intellectual surroundings constitute a third aspect of the workshop (ref. part 4.3). His first tests on the natural resonance of the piano were performed at his place of residence for *Deviata*, then at Ircam for *Dialoghi*, but the latter part of the writing work (the piano

³ Ref. part 4.2.1.

⁴ *Interactive Computer Music System* is a construction and mixing program written by Graziano Tisato (CSC).

part as well as the synthetic sounds) is achieved whenever possible depending on his successive trips and places of residence.

4.2 Technological tools

The expression « Technological tools » covers, in addition to the computers and software programs, all the technical prostheses at the disposal of the composer. Part 4.3.1 expounds the tests performed on instrumental writing. As for sound synthesis composition, this incited Stroppa to opt for an original computerized environment dedicated to composition – or better said, to his own composition method. Hence, sound synthesis used Fortran supported routines under *MusicV* in the eighties, *Carla* at the end of the eighties and beginning of the nineties for Computer Assisted Composition (CAC), then a *Chroma* library under *OpenMusic* (in control of *CSound*) in the nineties and the next decade, and *Antescofo* from 2007 on for real-time electronics.

4.2.1 Music V

To further explore the above-mentioned resonance logics, Stroppa felt the need to prolong the natural resonance of the piano with computer generated sounds, not only to cover the acoustic sound with an electronic veneer, but also to provide the piano with a wider range of resonance. As for the selection of *Music V* versus real time (with the 4i entrusted by Di Giugno to the CSC in 1982 – shortly after the project began), it would not be surprising if this were directly due to the rigorous training he received from his teacher Alvisé Vidolin. The texts written during the genesis show that Stroppa had highly precise ideas about the types of artificial sounds and resonance he wished to add to the piano sound. It was because Stroppa mastered *MusicV* in the same manner as a composer would for orchestration that he could compose computer generated sounds in a “synthetic orchestra” [3] which was capable of accompanying the piano much more efficiently in terms of perception than real-time computer-generated sounds. Moreover, *MusicV* is capable of efficiently rendering the most intimate and microscopic sound components. These ideas were directly derived from his training in acoustics and psychoacoustics and also from his earlier work on the piano, in particular the pianistic touch which enables musicians to enter the microscopic properties of sound. Jean-Claude Risset, another qualified pianist, was to live the same experience. This is why Stroppa actually “composed the sound itself” [10], within this new paradigm created almost two decades before.

It must also be noted that *Music V* offers a software architecture designed for and according to the wishes of the composers [11, 12], in the sense that the *score* and *orchestra* ergonomic functions are designed to offer musicians a direct link between acoustics and music. This results from pooling scientific thought and musical requirements at the *Bell Labs* under the sponsorship of Mathews and Risset. It is worth noting that in this case, tools were

common both to scientists and composers. The aim was to both acquire new scientific knowledge (*Bell Labs*) and generate musical inventions (Risset, Stroppa, Harvey, etc.).

4.2.2 Carla

Another software, *Carla*, was designed following Stroppa’s work on *Traiettoria*. Stroppa settled on two types of matrix-based chords in *Traiettoria*: one was more specifically used in *Deviata* and the other in *Dialoghi*, and both were used jointly in *Contrasti*. In addition, he frequently had to manually perform simple operations on these chords such as inversions and complements, so as to dispose of a range of different chords connected by common notes or intervals. After *Traiettoria*, according to pre-defined requirements, Stroppa continued to compose “by hand” this type of harmonic aggregate he named “Vertical Pitch Structure” (VPS) [13], for example for his piece of work *Spirali* (1989). Toward 1988, he entered Ircam with Francis Courtot [14] for the development of a Prolog 2-written CAC tool called *Carla*, which enabled the production of harmonic material under requirements to be specified. This requirement-ruled programming tool was used for *Elet... Fogytiglan* (1989, 1997, -) and for *Miniature Estrose* (1992, 1995, 2000, -), then it was re-written in LISP.

4.2.3 OMChroma

Upon the creation of *Traiettoria*, Stroppa’s “software workshop” was closely dependent on the requirements of the studios, in terms of utilization of the machine time-shared by the composers and the computer and/or science researchers, the extremely long computation time, and the unadaptable configuration of the software programs. *OpenMusic* is a CAC software which enables Stroppa to implement synthesis control patches. The synthesis control support to be loaded in *OpenMusic* (1999-2000) was rewritten by Stroppa on his own, then with the help of Carlos Agon *et al.* [15, 16, 17, 18], Serge Lemouton for the *OMChroma* library and more recently Jean Bresson for the latest *OMChroma* spinoffs [19]⁵. Incidentally, *OMChroma* library was widely shared by the community of composers, notwithstanding Stroppa’s composing practices. On this support, Stroppa wrote a Workspace entitled *Traiettoria* (2002) more pliable as a graphic interface which enabled him to re-create identical original sounds. The way this synthesis was made in delayed time was reproduced, in particular thanks to the proposed choice between 1/ entering data in Lisp language or 2/ specifying these data directly in the graphic interface. Hence, the procedures used in *Music V* and inscribed in Stroppa’s *computer jotters* upon composition could be thoroughly reproduced (and partially reproduced by Stroppa himself with his Workspace) in *OMChroma*.

⁵ Notwithstanding the reference [19], the most recent developments are published in C. Agon, J. Bresson and M. Stroppa, *OMChroma: Compositional Control of Sound Synthesis*, *Computer Music Journal*, Summer 2011, Vol. 35, No. 2, pp. 67-83.

4.2.4 Antescofo

Antescofo is an example of software created to fulfill the wishes of a composer. Two decades before, Philippe Manoury and Miller Puckette had innovated in the instrument/electronics interaction with the score follower concept under *Max/MSP*. Nevertheless, the relationship between the musician and the electronics was still somewhat unbalanced. Stroppa had always shown some hostility to real time [20] because this can sometimes produce anti-musical conditions, or more precisely a technological rather than musical relationship. This time, Stroppa decided to follow this path in 2007 in collaboration with Arshia Cont by envisaging a man/machine predictive device, which would be more in phase with the cognitive reality of the time synchronization between two performing musicians in chamber music conditions. Stroppa then fitted in the continuity of several pieces of work “for chamber electronics”, i.e., an autonomous electronic portion with artificially intelligent reactions. Crossing Marco Stroppa’s and Arshia Cont’s⁶ testimonies confirms the existence of “mutual inspiration” between the two protagonists, thus resulting in a software program at the meeting point of scientific and musical researches.

4.3 Experimental practice and intellectual scientific surroundings

4.3.1 From the tests on piano resonance to the “Treaty of resonance”

The *Traiettorie* notebooks show that he performed numerous tests on piano resonance. Direct work on piano was inspired to him by the acoustic phenomenon approach of sound he acquired while training at the CSC. Indeed, his aim was not to test harmonic sequences or chords (even when the work was based on two “manually” produced chords). He did not attempt to reproduce a piano spectrum by notes as would a spectral musician, but rather, using only the natural resources of the piano (the three pedals, the keyboard touch), to generate incredible resonances generally outside the range of standard piano writing. In view of the extreme difficulty in fully concretizing his musician wishes, Stroppa used electronic writing to create resonances. There again, direct practice on the piano without the aid of computers seems widely nurtured by his very extensive knowledge of the physical and psychoacoustic phenomena.

This preliminary work on the piano was then the subject of another piece for piano solo, *Miniature estrose* (1992, 1995, 2000, -). It can be noted that the piece brings the final touches to the initial project of *Traiettorie* which was meant to be written for a piano solo. Less than a quarter of a century later, Stroppa had Ricordi [22] republish the scores, to which he added the “treaty on resonance”, the end result of his above-mentioned former experiences.

⁶ See [21], at 4’.

4.3.2 Sub-routines (PLF)

In the pre-compositional phase, Stroppa wrote PLF sub-routines. To illustrate the correspondence between the types of syntheses and the families of sounds at the beginning of the compositional process, here are three examples drawn from *Deviata* :

- PLF 10 produces additive synthesis to generate cluster type sounds (family C, see Figure 1)
- PLF 20/21 produces what is called granular synthesis nowadays, to generate glissandi in particular (family A)
- PLF 33 produces FM synthesis to generate attack/resonance type sounds, or more globally the sustain of complex sounds (family B)

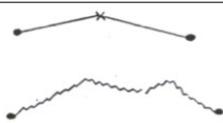
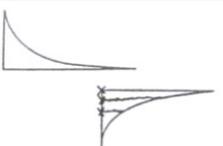
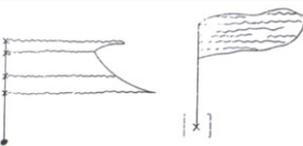
| | Feature | Score notation |
|----------|---|--|
| Family A | Glissando (from smooth to granular) |  |
| Family B | Percussive |  |
| Family C | Resonant cluster (with or without perceptible fundamental) |  |

Figure 1. The 3 sound families in *Deviata*⁷

4.3.3 Tversky’s contrast model, McAdams’ aural stream fusion/separation

As regards Marco Stroppa’s intellectual and scientific surroundings, it is worth mentioning the “mandatory readings” of his recent training at the CSC, in particular the work of Roederer [23]. In this context, *Cognition and Categorization* by Rosch and Lloyd [7] is a bedside book for Stroppa. In 1982, basic but nonetheless essential psychoacoustics principles were also laid out in his CSC courses of that time. For Stroppa, psychoacoustics for sounds synthesis was tantamount to what the orchestration treaty is for orchestral writing. The knowledge of Roederer’s [23], McAdams’ [4, 5] or Wessel’s [6] work scientifically validated his initial intuition and at the same time helped him to systemize his musical thoughts.

Based on this knowledge of the cognitive and psychoacoustic science, Stroppa built up the concept of Musical Information Organisms (MIO), in the midst of the constructions of sound families (ref. next paragraph). What mattered for Stroppa was the perceived result, taking a precise account of observations drawn from cognitive and psychoacoustic sciences. It is clear that numerous techniques for local or global writing [8] are more or less direct transpositions of these psychoacoustic “rules”. But

⁷ Transcription of the sounds families, from [3]

there again, the knowledge of these concepts was contemporary with his writings, often providing a confirmation of his first musical pages as in the first two movements *Deviata* and *Dialoghi*, and sometimes in anticipation as in the third movement *Contrasti*. In the latter case, deeper knowledge of the latest discoveries in cognitive and psychoacoustic sciences enabled Stroppa to make the most of the dual concept of similarity/contrast model [24, 25] and fusion/separation.

The "Contrast model" was taken up by Stroppa in an example which he used as a support for composition courses dealing with *Traiettorie* (see Figure 2).

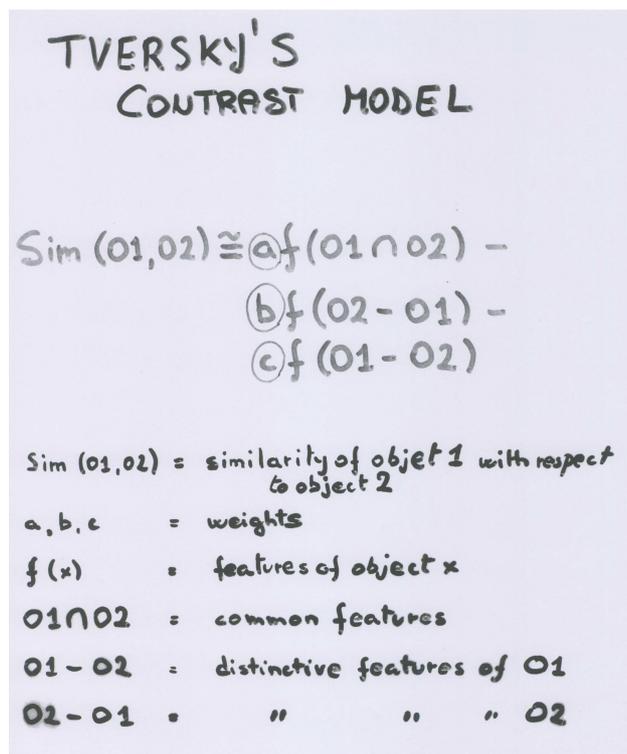


Figure 2. Tversky's contrast model reproduced by Stroppa

To explain the link existing between these different scientific concepts and Stroppa's composition processes, it is necessary to detail how he designed the MIO. Each of these consists of several sound objects, in turn containing more basic musical characteristics. Stroppa's compositional process classified these sound objects according to Tversky's model similarity index. Then he could identify organisms formed by the association of these objects: the MIO, which have a strong identity recognizable on perception.

Upon assessment of the similarity index between two sound objects, the contrast model combines three terms provided with weighting coefficients linked to the auditory judgment and closely associated to Stroppa's decision: 1) characteristics common to both objects, 2) characteristics specific to the first object, 3) characteristics specific to the second object. Thus for an MIO consisting of two superimposed sounds, either of the following cases may occur:

1) the two sounds are discriminated when the similarity index is low. This separation phenomenon is frequent in *Deviata* with its highly characteristic synthetic sounds as indicated in fig. 1.



Figure 3. Page 4 of *Contrasti* original working scores, size between A3 and A2.

2) the two sounds cannot be discriminated when the similarity index is high. This fusion phenomenon is frequent in some *Contrasti* MIO perceived as single although they are made of two superimposed sounds. As revealed by the *computer jotters*, the most frequent case consists of a high-pitch part generated by additive synthesis, and a low-pitch part generated by formant FM synthesis (see Figure 3: D3A et D3B). In this large format, mixed scores used by Stroppa as a working document to compose the mixed passages of *Contrasti*, the content of the synthesis part can be reviewed finely. Each sound family (itemized with a letter) appears as an instance (numbered in increments) which can often be divided into two parts (A et B, as mentioned above) generated by PLF 10 and by PLF 33 (in this case, sounds C3, D3, F9). This writing technique per MIO implemented for the electronic part also operates as an extension of the piano part. Indeed, the harmonic content of aggregate D3A (synthetic sound at 1'36"25) is played on the piano at 1'34" and, in the same way, the harmonic content of aggregate D3B is played on the piano at little before

1'36"25 (see Figure 3). Thus in this passage, the recurrent doubling of the piano by the electronic sound (synchronous or offset) infers that an outstanding musical passage (chords, stressed notes, etc.) is most often perceived globally as an organism (whether it is instrumental or electronic).

4.3.4 Structured programming

Between September 1984 and September 1986, Stroppa completed his training at the MIT. He then devoted himself to fundamental computer science, making attempts at structured programming (basically Fortran language used by *Music V*). Readings such as *Structure and Interpretation of Computer Programs* [26], relative to these domains obviously little connected to composition, provided a rational explanation of how to perform compositional practice. The main idea is the decomposition of a problem in primitives. Stroppa declared in an interview: "From a compositional point of view, what is my primitive? Where am I to start the definition of a material from the viewpoint of a composer (material that may be complex but that I use as a unit)? Is it a chord, a note, a note spectrum, a rhythm, a 3-minute long process?"⁸.

In addition, he suggested that, for the compositional work, he preferred using a Top-down concept (man language to machine language): "Because at the top is the problem as I understand it and not as it is presented by the machine. And this way of working, very well explained by Sussman, is imperative epistemology. It is a study of the knowledge structures, although not a declarative one (not defining what it is), but giving a clue as to how the problem can be solved. This imperative epistemology scope – actually, I found this later -, is exactly the type of problem we composers are confronted with"⁹. He finished by saying that: "I have discovered that writing a program was like writing a fugue"¹⁰. For a composer, decomposing problems into primitives (Top-down) is a permanent act. Then, the creation process is completed by combining several primitives (Bottom-up) in sequences of primitives¹¹. For example, to create an MIO *via* sound synthesis, Stroppa had to decompose abstractly the MIO into primitives, that is, in different sound objects. After making a step-by-step synthesis of each of these sound objects, he must recombine them to make up the MIO. He repeated the process for other MIO until a trajectory (*Traiettorie*) was gradually traced by these MIO.

4.4 Personal background of the composer

Without intending to be exhaustive about this final aspect of the composer's workshop, it can be noted that

⁸ 28/09/2009, interview of Marco Stroppa by Noémie Sprenger-Ohana and Vincent Tiffon at Ircam (Paris)

⁹ 08/01/2010, interview of Marco Stroppa by Noémie Sprenger-Ohana and Vincent Tiffon at Ircam (Paris).

¹⁰ 26/02/2010, interview of Marco Stroppa by Noémie Sprenger-Ohana and Vincent Tiffon at Ircam (Paris).

¹¹ This is called "functions" or "abstractions" in programs such as Max/MSP.

Stroppa's thoughts were experimental. Music composition is equivalent to musical research. In other words, Stroppa's works were never finished. He liked to keep the work as a sketch which could be updated and improved later, and above all used as a matrix or support for novel ideas for the next work. For example the compositional process by organisms (MIO) and the attempts at piano resonance were repeated in *Miniature Estrose* (1991). Stroppa followed a thought process akin to scientific experimental reasoning.

5. CONCLUSION

Structured programming, cognitive sciences, acoustics and psychoacoustics boosted the thought process of Marco Stroppa relative to his language, his tools, in short his "workshop of composition" in the eighties. By reading scientific texts and acquiring scientific knowledge, Stroppa found confirmation of a number of musical practices. Simultaneously, this scientific knowledge contributed to the delineation of an autonomous mind, free from the restrictions imposed by esthetic trends. Finally, Stroppa's dual scientific and musical competence, and his closeness to researchers have also contributed to developing his knowledge and to the designing of software programs dedicated to musical research.

Acknowledgments

This research on the composition processes subjacent to Marco Stroppa's *Traiettorie* was performed within the frame of a contract with the ANR (French state research agency) entitled "MuTeC" (Musicology of contemporaneous composition techniques) between 2009 and 2011, under the responsibility of Nicolas Donin (Ircam) (<http://apm.ircam.fr/mutec/>). This is one of the nine "fields" under study, among which the pieces of work from Pierre Boulez, Stefano Gervasoni, Gérard Grisey, Jean-Luc Hervé, Charles Koechlin, Mickaël Jarrell, Tristan Murail, and Bernd-Alois Zimmermann.

We warmly thank translator Xavier Lannuzel, Pierre-Laurent Aimard, Jean Bresson, Sergio Canazza, Serge Lemouton, Jean-Claude Risset, Alvis Vidolin, and above all Marco Stroppa for his ever continuous availability.

Reproduction of genetic documents kindly authorized by Marco Stroppa.

6. REFERENCES

- [1] *Traiettorie... deviata*, Ricordi, 1984, 133770; *Dialoghi*, Ricordi, 1985, 134015; *Contrasti*, Ricordi, 1985, 134261.
- [2] M. Stroppa, *Traiettorie*, Pierre-Laurent Aimard, piano, Marco Stroppa, sounds projection, CD Wergo, recorded in 1991 and published in 1992, WER 2030-2. New edition update in: *Stradivarius STR 57008*, 2009.

- [3] M. Stroppa, "Un orchestre synthétique : Remarques sur une notation personnelle" in *Le timbre, métaphore pour la composition*, Jean-Baptiste Barrière ed., Bourgois/IRCAM, 1991, pp.485-538.
- [4] S. Mc Adams, "Spectral Fusion and the creation of auditory images", in M. Clynes, ed. *Music, Mind, and Brain: The Neuropsychology of Music*. New York: Plenum, 1982, pp. 279-298.
- [5] S. McAdams, *Spectral Fusion, Spectral Parsing, and the Formation of Auditory Images*, Ph.D. Dissertation, CCRMA Stanford University, 1984.
- [6] D. Wessel, "Timbre space as a musical control structure", *Computer Music Journal*, summer 1979, Vol 3, n°2, pp. 45-52.
- [7] E. Rosch and B. Lloyd eds., *Cognition and categorization*, Hillsdale, New Jersey, Lawrence Erlbaum Associates, 1978.
- [8] M. Stroppa, "Musical Information Organisms: An Approach to Composition", *Contemporary Music Review*, vol. 4 ; in *Music and the Cognitive sciences*, S. Mc Adams and I. Deliège eds., Harwood Academic Publishers, London, 1989, pp. 131-163 ; "Les Organismes d'Information Musicale : une approche de la composition", *La Musique et les sciences cognitives*, S. McAdams Stephen and I. Deliège eds., Bruxelles, Pierre Mardaga, 1989, pp. 203-234.
- [9] A. Vidolin and R. Doati eds., *LIMB (Laboratorio permanente per l'Informatica Musicale della Biennale di Venezia)*, quaderno 1-5, Venezia, 1980-1985 (in collaborazione con il C.S.C. dell'Università di Padova).
- [10] J.-C. Risset, "Composer le son : expériences avec l'ordinateur, 1964-1989", *Contrechamps*, 11, 1990, pp. 107-126.
- [11] M. Mathews, J.E. Miller, F. R. Moore, J. R. Pierce and J.-C. Risset, *The Technology of Computer Music*, Cambridge, Massachusetts, MIT Press, 1969.
- [12] J.-C. Risset, *Catalog of computer-synthesized sounds*, Bell Telephone Laboratories, 1969 (reprinted in *An Introductory Catalogue of Computer Synthesized Sounds*), *Computer Music Currents* n° 13, "The Historical CD of Digital Sound Synthesis", Wergo, Germany.
- [13] M. Stroppa, "Structure, Categorization, Generation and Selection of Vertical Pitch Structures: a Musical Application in Computer-Assisted Composition", IRCAM Document, Paris, 1988.
- [14] F. Courtot, "Carla : Knowledge acquisition and induction for computer", *Interface*, Vol1, n°3-4, 1992.
- [15] C. Agon, M. Stroppa, G. Assayag, "High Level Musical Control of Sound Synthesis in OpenMusic", *Proc. International Computer Music Conference*, Berlin, 2000 pp. 332-335.
- [16] M. Stroppa, "Paradigm for the high-level musical control of Digital Signal Processing", *Proc. Int. Conf. on Digital Audio Effects (DAFx-00)*, Verona, Italy, 2000, addendum.
- [17] J. Bresson, M. Stroppa, C. Agon, "Symbolic Control of Sound Synthesis in a computer-assisted composition environment", *Proc. International Computer Music Conference*, Barcelona, September 2005, pp. 303-306
- [18] J. Bresson, M. Stroppa, C. Agon, "Generation and Representation of Data and Events for the Control of Sound Synthesis", *Proc. Conf. Sound and Music Computing*, Lefkada, Greece, July 2007, pp. 178-184.
- [19] C. Agon, S. Lemouton, M. Stroppa, "omChroma : vers une formalisation compositionnelle des processus de synthèse sonore", *Journées d'Informatique Musicale*, 9th edition, 2002, GMEM, Marseille, pp. 51-57.
- [20] M. Stroppa, "Live electronics or ... live music ? Towards a critique of interaction", *Contemporary Music Review*, Vol. 18, Part 3, 1999, pp. 41-77.
- [21] N. Donin and B. Martin, *Images d'une oeuvre n°7, "L'électronique de chambre de Marco Stroppa"*, IrCam/Centre Pompidou, http://www.ircam.fr/images_d_une_oeuvre.html#c3847.
- [22] M. Stroppa, *Miniature Estrose, Ricordi n°136804* (including "treaty on resonance").
- [23] J. Roederer, *Introduction to the Physics and Psychophysics of Music*, Springer, 1979 (first ed. 1973).
- [24] A. Tversky, "Features of similarity", *Psychological Review*, 84, 1977, pp. 327-352.
- [25] A. Tversky and I. Gati, "Studies of similarity", *Cognition and categorization*, E. Rosch and B. Lloyd eds., Hillsdale, New Jersey, Lawrence Erlbaum Associates, 1978, pp.79-98.
- [26] H. Abelson and G. Sussman, *Structure and Interpretation of Computer Programs*, The MIT Press, 1985.

LIMITS OF CONTROL

Hanns Holger Rutz

Interdisciplinary Centre for Computer Music Research (ICCMR) – University of Plymouth

hanns.rutz@plymouth.ac.uk

ABSTRACT

We are analysing the implications of music composition through programming, in particular the possibilities and limitations of tracing the composition process through computer artefacts. The analysis is attached to the case study of a sound installation. This work was realised using a new programming system which is briefly introduced. Through these observations we are probing and adjusting a model of the composition process which draws ideas from systems theory, the experimental system of differential reproduction, and deconstructionism.

1. INTRODUCTION

«But words are still the principal instruments of control»*

The term "computer music" can be used to denote a musical praxis and musical research that reflect the «profound influence of computer science on music»[1], and thus they substantially depend on the medial implications of the computer. The most crucial implication, as Loy and Curtis point out in a 1985 survey, is the formalisation of concepts through the use of programming languages. The involvement of programming languages distinguishes this form of computer music from other forms in which the musician takes the role of the user of readily available applications «in which inputs of a simple structure produce effects (such as outputs) desired by the user»[1]. Coincidentally, the sound transformation plug-ins offered by commercial music software are often called "effects"—they offer fully prescribed, and often standardised, tools to achieve well known goals.

The term 'goal' was early introduced in cybernetics (e.g. [2]), and is linked to the concept of control which is a regulatory mechanism to direct the system towards a goal. The feedback from the system's output to the regulator's input can be replaced by the human being listening to the sound produced by an "effect", and the regulator is the knob in the interface which also goes by the name of a "controller". By rotating the knob, the imagined sound—the goal—is incrementally approached.

*All epigraphs from William S. Burroughs' essay *The Limits of Control*, but the last which is from the Jim Jarmusch movie of the same title.

Obviously, the application of these terms seems easy in a rather mechanical case like this, but how is goal directness translated when using a musical programming language, and how is control exercised with words instead of knobs? As Rosenblueth et al. carefully put it [2], purposefulness of a behaviour is an attribution resulting from an *interpretation*. In other words, an observer is needed to make this attribution.

2. DISSEMINATION

«A basic impasse of all control machines is this: Control needs time in which to exercise control»

When using systems thinking as an approach it is important to remind oneself of its abstract nature. Although some authors give systems an ontological status and then deduce (observer specific) symbolic representations in the form of models as homomorphic mappings of the real world systems [3], we follow Checkland here in that systems themselves are just abstractions and epistemological tools [4]. One of the main abstractions, perhaps the most fundamental one, is that of the boundary between a system and its environment (cf. [5]): Where is the boundary between the computer music composer taking the role of a programmer and any other programmer (where is the boundary between programming and composing, between a programme and a composition)? Where does the process of composition begin and where does it end?

Light is shed on these questions with the help of a case study: «Dissemination»¹ is the title of an audio-visual installation by Hanns Holger Rutz and Nayarí Castillo, in which both media create a space by relying on and reflecting upon each other. It consists of horizontally and vertically suspended glass panels functioning both as a body for sound resonance and diffusion—through the attachment of sound transducers—as well as specimen holders carrying petri dishes filled with seeds (see fig. 1). Several tableaux are generated which unwind the temporal development of the generative sound composition in space. Flying seeds, which due to their natural features can be dispersed by wind, symbolise motion, traveling and migration. The act of dissemination is interpreted as an act of re-writing, leaving traces, producing movement instead of a fulfilment, strategy instead of finality.

Without going into many details, the concept of the sound installation was to create a generative real time composition with different temporal layers, some of which are

¹http://www.sciss.de/texts/ins_dissemination.html



Figure 1: Photos from *Dissemination*. On the left, the site (Gallery ESC Graz) is seen with the suspended glass panels pervading the space. The daylight is filtered with yellow gels. The right photo shows the pairing of a horizontal and vertical plate, the vertical plate being excited by the black sound transducer, the horizontal plate holding an ensemble of petri dishes. The middle photo shows a close up of a petri dish holding flying seeds.

cyclic and others which span an ongoing thread over the duration of the exhibition. Although the setup had been conceptualised back in September 2009, it was only in August of the following year that the sound composition was carried out. Within this period a framework for the description and connection of sound processes had been developed, and this was the second project in which it was put to practice.

Thus, when looking at the programming, an arbitrary line must be drawn between the preparatory work and the composition in the narrow sense. Since both the framework and the composition have been managed using a versioning system, we are able to look at the development over time. The repository containing the actual composition was opened in August 2010 and is visualised in figure 2. The lines of code written in the Scala programming language have been manually categorised as belonging either to the technical infrastructure of the system or carrying actual musical meaning. The piece was exhibited twice, with the premiere taking place on September 18, and the second exhibition beginning on October 20.

It can be seen that in the beginning more time is spent with the programming of the infrastructure than the actual composition. Since the framework was rather new, various adaptations were required to realise the ideas for the project. Infrastructure code generation decreases to nearly zero towards the end of the first composition cycle with another final spike due to preparing the work for autonomous operation during the exhibition².

The other distinction made here is between newly written

²A bug was found that caused the system to become unstable after a couple of hours running, so several measures had to be taken to work around it.

code and code derived from previous projects or from older stages of the same project. This follows from a recursion model of the composition process developed in [6] which proposes that this process is driven by material injected from outside the system as well as (re-)transformation of material already inside the system. Finding this pattern in manifest condensations—computer artefacts such as the code—could indicate that something similar is happening in the psychic system of the composer. This is confirmed by other studies, for example Collins in his case study of a composer working on instrumental staff based music identifies a combination of both linear and recursive motions in the development of the piece [7].

Infrastructure code copied from previous projects indicates lack of modularisation, but can be mostly explained by the time pressure factor in the realisation of a work³. On the other hand, the adaptation of existing musical code mostly amounts to sound synthesis and transformation instruments which are reused. In the second half, self-referential composition code—code which is derived from code within the same project—begins to increase and finally amounts to roughly half of the additionally produced code volume. Musically, this can be interpreted as variation: More sound processes are introduced which share structural similarities with previously created processes but are then differentiated—for example by using other sound files, other parametrisation, probabilities, spatialisation etc. Figure 2 therefore seems to support the model of an increasingly recursive behaviour of the composition process.

³Copying code in the short term is a much faster measure than refactoring into reusable external modules which pays off in the long term.

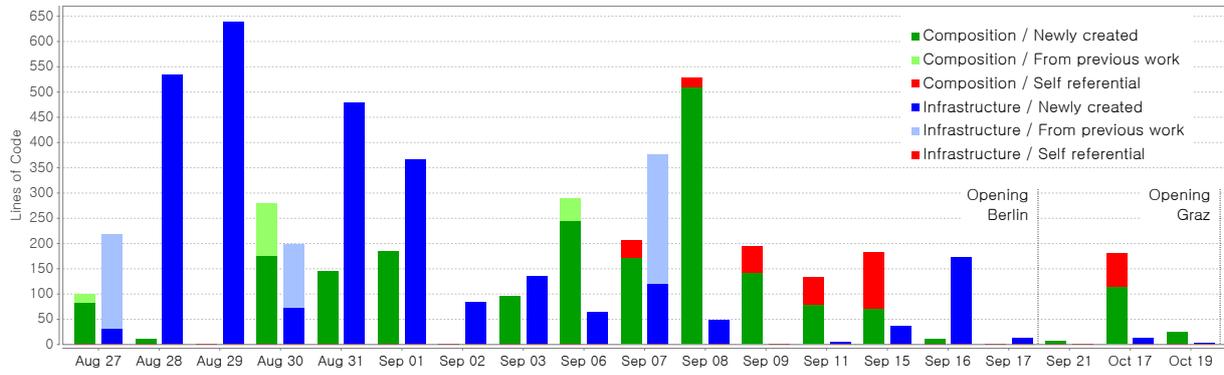


Figure 2: Code commits to the GIT repositories of the composition and frameworks it depends on. Line count includes lines created, lines edited and lines deleted. Multiple commits per day are integrated. Code carrying musical meaning in the narrow sense is shown in green and contrasted with code used to build the infrastructure, used for debugging, and so forth, which is shown in blue. Bars are split to distinguish newly created parts from parts derived from previous work.

3. PERMEABLE BOUNDARIES

«Concession is another control bind»

The ambitious goal of "strategy instead of finality" was to reflect the recursive model of human activity in the generative structure of the piece itself. After all, «Programming is not about doing; it's about causing the doing»[8]. Following the technical agnosticity of a frameworks's application programming interface (API) as to whether a call is issued by direct human action or deferred action in the form of an algorithm (the "caused doing"), both direct and indirect actions can be collapsed in the notion of an abstract writing process as outlined by Derrida [9] (cf. [6]). A data structure which would allow the generative parts of the piece to inscribe their actions into the piece itself was proposed in [10], but unfortunately had not been ready for production by the time *Dissemination* was created.

As a result, we can observe how the practice of the art production ignores "impossibilities" imposed by the systems at hand and transcends their boundaries: The idea of a persistent trace was upheld by another type of observer: One of the main algorithms, named *Plates* (see figure 3), instead of being able to read traces left inside a data structure, uses audio signal feature extraction to gain insight into the instantaneous state of the composition. It tries to maintain a sort of energy balance, reacting by generation of new material or withdrawal of current material. The production of new material is accomplished by recording parts of the installation's output signal and feeding it into a set of signal transformation processes, eventually re-injecting the transformed material into the piece. The material thus generated remains on the hard disk of the installation computer and leaves a persistent trace over the period of the exhibition. This pool of material is periodically thinned out so that the second function of memory, forgetting, is included.

However, the assumption that, had the persistence layer been developed to planned extent, the whole composition would have been confined to the boundaries of this layer and hence be fully traceable, is illusory. This is be-

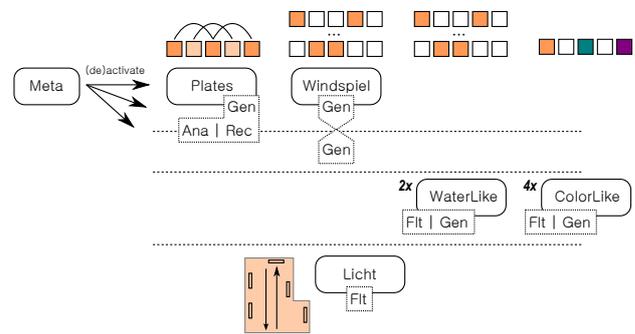


Figure 3: Schema of the constituent sound processes in *Dissemination*. Vertical lines partition sound layers where lower layers can filter or shadow upper layers. Orange squares indicate spatialisation modes, corresponding to the five channel diffusion in the first exhibition, and with the mode for *Licht* showing the floor plan in Berlin. Each process provides different components which can act as sound generators, filters or analysing and recording stages.

cause our framework is a designed system and as such has a prescribed model and a prescribed purpose (allowing a traceable form of composition), and it would consequently fall into the category of a "taciturn system" according to Pask [11]. On the other hand, we offer the composer the programming language which falls into the second category of "language oriented systems"—these exhibit contingency since the composer can change her mind and instruct it to do other things than before. The composer-observer resides inside the boundaries of the language oriented system, so to say, but outside the boundaries of the technological observer, the domain specific language (DSL), the framework subset of the language.

This is illustrated in the top most diagram of figure 4. The framework's language contains a notion of sound processes which are described in terms of input and output signals, control parameters, resources such as sound buffers and sound files, as well as a function which generates a graph of unit generators responsible for the actual sound analysis, transformation and synthesis, using the SUPER-COLLIDER server. A simplified example from the *Plates*

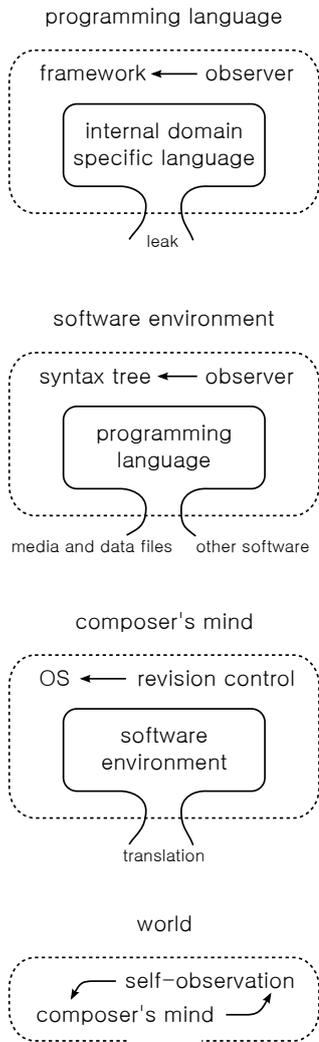


Figure 4: The idea of a composition system as observing (tracing) the composer's traces, and how it is undermined.

algorithm is shown in figure 5.

One of the problems—defining a stable reference point in the sound stream which could be analysed despite all processes appearing and disappearing dynamically—was solved within the framework: Special collector nodes are created which are maintained throughout the installation. The two collectors *pColl1* and *pColl2* correspond to the two upmost dotted horizontal lines in figure 3. This allows for example another process, *Windspiel*, to decide whether it wants its sound output be picked up by *Plates*' analysis (using level *pColl1*) or not (using level *pColl2*).

Notwithstanding, the real problem is the highlighted line in figure 5: Once every second the smoothed out analysis data is sampled and a subroutine *newAnalysis* called, and this is where the code escapes the intended traceability of the persistence framework. One might well be able to catch and represent the graph function within this framework, but the client code looking at the analysis data and making decisions about starting or stopping sound processes leaks outside. Code and data fall apart, and there are only two possible solutions to this dilemma: Either the approach of developing the composition within a DSL em-

```

val fColl = filter("+") { graph { in=>in }}
val pColl1 = fColl.make
val pColl2 = fColl.make
pColl1 ~> pColl2

val pAna = (diff("ana") { graph { in =>
  val bufID = bufEmpty(1024).id
  val chain = FFT(bufID, Mix(in))
  val loud = Loudness.kr(chain)
  val centr = SpecPcile.kr(chain)
  val flat = SpecFlatness.kr(chain)
  val compound= List(loud, centr, flat)
  val smooth = Lag.kr(compound, 10)
  1.react(smooth) (plate.newAnalysis(_))
}}).make

pColl1 ~> pAna
pColl1 ~> pRec
pAna.play
    
```

Figure 5: Example of an analysis sound process code from the *Plates* component. Also shown is the creation and connection of collector nodes which establish stable references in the conceived level structure.

bedded in a general purpose language is given up in favour of a more rigid environment—perhaps one in which it is still possible to write functions for the synthesis graph, but such that they are embedded in hidden glue code that prevents the composer from escaping a given scope—, or we move the observer outside the language, dealing not directly with the framework anymore but the abstract syntax tree (AST) representation of the language. This latter solution corresponds with the second diagram in figure 4. In the argumentation of soft systems approaches—which seem adequate when dealing with language oriented systems—, any system is open by definition, since it effects and is affected by its environment. The "leak" can be recursively closed: «Any open system can always be reframed as closed by expanding the system boundaries to include its environment.»[12].

Figure 4 shows this recursive closure with the software observer eventually being replaced by the composer observing herself, meaning «the hermeneutic circle of interpretation-action, on which all human activity is based.»[13] Each stage bears new problems. In the AST analysis, it becomes unfeasible to trace musical intentions in a fine grained way, it also places a serious technical hurdle by the need to represent multiple versions of code fragments within the same class loader. On the other hand, the composition may easily integrate data which is not part of the host language. In the diagram, this is indicated by "media and data files" and "other software". In *Dissemination*, several sound files were incorporated and meta files in the form of segmentation data for particular sounds. Furthermore, a separate software *FSCAPE* was used from various processes to render sound transformations while the installation is running. The next logical boundary would thus be a file revision control system that could observe all data involved. Again, if the fine levels of granularity shall not be lost, this must be combined with observers of the previous levels. The third stage definitely escapes the ability to monitor this purely in software, as the actual piece in-

volves decision making in the composer's head which may not be directly projected onto the software. A simple reminder is that this is an audio-visual installation, so the composition of the physical components, the arrangement and conditioning of the space are not covered, neither are notes taken in sketchbooks and so forth.

4. SOLUTION SPACES

«All control systems try to make control as tight as possible, but at the same time, if they succeeded completely, there would be nothing left to control»

It can only be concluded that any observation system is limited and should be modest and candid about its limitations. It goes without saying that this does not imply that such systems are not useful, but a clarification of the purpose of this observation is required. It is not about the establishing of a trace as an empirical fact pointing back to an arche-trace, as there is no such thing as an arche-trace (cf. [9]). The transportation of code from previous projects and frameworks into a "new" project has already indicated that, and figure 6 underlines it even further, as it shows how DSP processes, sound files and concepts from previous projects form an important part of the piece. The DSP processes depicted have been developed by the author over a period of ten years, and some of them are highly idiosyncratic and unfold their potential when connected with the other processes, so they are considered essential compositional elements and not just "effects" (infrastructure). The sound files, too, span almost a decade. Some of them had been used in previous projects, some had never been used in a piece, others have been used in a completely different manner before. Finally «Kalligraphie» and «Amplifikation» are two sound installations which had a profound influence on this new one: They established the idea of a sound mobile constructed from semi-independent processes and driven by a dedicated meta process, the physical setup of glass plates, transducers and daylight colourisation, and even specific processes such as *Licht*—a process which takes the frequency response of the glass plates and constructs an inverse filter, imposing a moving gesture of immateriality onto the "unfiltered" sounds—which was enhanced from the original version in «Amplifikation».

The boundary between this piece and all previous pieces seems to correlate with the performances and exhibitions of the respective pieces, but it is just as conventional as any system boundary and may not help in the analysis of the composition process.

Observation as drawing-a-distinction is not so much different from writing, especially when the observer is located within the system, as this type of observer not only describes the system, but is also a relation, a determining component of the system, and the very description of the system dynamically changes the subject of description [13]. Thus, when we construct a computer music composition system with an observable, traceable data structure, we aim at changing the notion of composition altogether,

and by doing that we hope to contribute to a form of computer music which is truly depending on the computer as medium. By offering the trace of the composition process as an access point to recursive transformation, we are not supporting the closure of a problem space but the opening of a solution space (cf. [7]).

This is best illustrated with a classic concept from control theory, Ashby's law of requisite variety [14]. It states that goal-directed systems pursue a state of equilibrium, and that in order to shield them from disturbances from the environment—which would deter the system from its goal—the regulator must provide counter actions which have a variety that is at least as great as the variety of disturbances. The function of the regulator thus is to minimise variety in the output. Obviously this does not resonate with language oriented systems which deal with communication and wish to maximise the variety of possible expressions. These two types of systems do not contradict each other, as one can easily specify expressiveness as the goal of a system, so that the regulator would minimise disturbances which inhibit expressiveness. Heylighen and Joslyn portrait this duality as interaction between two systems with complementary goals [3].

Another way to see this complementarity is the experimental system described by Rheinberger which is based on differential reproduction [15]. It is the *modus operandi* of the scientific system but can be equally applied to arts. Differential reproduction is a process of repetition and not replication which means that the goal of identity is substituted for one of variation. In each iteration of an experiment a form of cohesion must be established that allows it to be compared to the previous experiment, but at the same time the system must be open for disturbances from the environment so that unforeseen things can happen. Unforeseen things *should* happen because the scientific system aims at creating new knowledge, similarly the arts aim at creating novel experiences. While technology serves as a background layer and is a form of answering-machine—problem closure, Pask's taciturn system—, the epistemic object is a question-generating machine—opening the solution space, the language oriented system.

5. CONCLUSIONS

«... musical instruments ... still sound even when not being played, have a memory; every note that has been played once with them, still there, inside, is resonating in the molecules of the wood»

We have identified computer music (a better term would be computer sound art) as a form which is intrinsically inspired by the computer as medium and not just a workbench with tools. Systems theory and cybernetics were evaluated for the usefulness and applicability of their concepts regarding the computer music composition process. The distinction of language oriented from taciturn systems and the identification of the former with computer music programming languages opened the discussion of specific terms such as 'goal', 'boundary' and 'observer'.

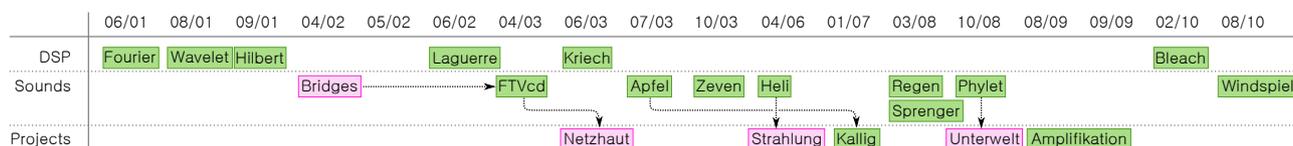


Figure 6: Selection of external references for the piece and their establishment over time: Special digital signal processing algorithms, sound files, and previous works. Pink colour items do not have a direct link with «Dissemination», arrows indicate dependencies between these references.

This composition approach was exemplified by an audio-visual installation work. Using data from a versioning system we found indications that the composition process can be modelled as a recursive system which takes input both from new ideas but equally from previous projects and then increasingly relies on transformed material fed back from previous iterations within the same process. Despite being reinforced by other studies of composer observation, further research is needed to turn these indications into generalised claims about computer music composition.

While the software framework used was not yet capable of tracing the writing of the composition, strategies have been employed to overcome this limitation. A memory was realised as recorded and rendered sound files and self observation was realised using audio feature extraction. Elaborating on the hypothetical implementation of the persistent observing system, it was found that any such observation system is inherently limited in scope and that it lies in the nature of the programming language approach—maximising expressiveness and communication with its environment—that system bounds become periferous.

Hitting the limits of objective or "unconcerned" observation, we introduced a turn in the purpose of observation. A "concerned" observation is actively transforming the compositional process by facilitating a recursive self-reflection and focusing on its creative potential rather than an archival idea of establishing the origins or roots of a composition. This self-reflection can be interpreted as a strategy for maintaining and increasing a solution space—for example by allowing the composer to establish new connections and relations between existing elements of the composition cutting across different versions in time (the 'meld' operation of confluent persistence)—while at the same time allowing for a cohesion between past and future states of a composition. The oscillation between cohesion and openness to disturbance from the environment is what classifies this process as a differential reproduction, supporting the emergence of new forms of expressions and thereby supporting a core interest of the arts.

6. REFERENCES

- [1] G. Loy and C. Abbott, "Programming Languages for Computer Music Synthesis, Performance, and Composition," *ACM Computing Surveys (CSUR)*, vol. 17, no. 2, pp. 235–265, 1985.
- [2] A. Rosenblueth, N. Wiener, and J. Bigelow, "Behavior, Purpose and Teleology," *Philosophy of Science*, pp. 18–24, 1943.
- [3] F. Heylighen and C. Joslyn, "Cybernetics and Second Order Cybernetics," in *Encyclopedia of Physical Science and Technology*, R. A. Meyers, Ed. Academic Press, 2001, vol. 4, pp. 155–170.
- [4] P. Checkland, "Systems Thinking," in *Rethinking Management Information Systems*, W. L. Currie and R. Galliers, Eds. New York: Oxford University Press, 1999, pp. 45–56.
- [5] R. L. Flood, "Unleashing the 'Open System' Metaphor," *Systemic Practice and Action Research*, vol. 1, no. 3, pp. 313–318, 1988.
- [6] H. H. Rutz, E. Miranda, and G. Eckel, "Reproducibility and Random Access in Sound Synthesis," in *Proceedings of the International Computer Music Conference*, 2011.
- [7] D. Collins, "A synthesis process model of creative thinking in music composition," *Psychology of Music*, vol. 33, no. 2, pp. 193–216, 2005.
- [8] D. Harel, "Can Programming Be Liberated, Period?" *Computer (IEEE)*, vol. 41, no. 1, pp. 28–37, 2008.
- [9] J. Derrida, *Of Grammatology*. Baltimore: Johns Hopkins University Press, 1997 (1967), trans. Gayatri Chakravorty Spivak.
- [10] H. H. Rutz, E. Miranda, and G. Eckel, "On the Traceability of the Compositional Process," in *Proceedings of the Sound and Music Computing Conference*, 2010, pp. 38:1–38:7.
- [11] G. Pask, "The meaning of cybernetics in the behavioural sciences (The cybernetics of behaviour and cognition; extending the meaning of 'goal')," *Progress of Cybernetics*, vol. 1, pp. 15–44, 1969.
- [12] A. Ryan, "What is a Systems Approach?" *Arxiv preprint arXiv:0809.1698*, 2008.
- [13] F. J. Varela, "Autonomy and Autopoiesis," in *Self-organizing Systems: An Interdisciplinary Approach*, G. Roth and H. Schwegler, Eds. Frankfurt and New York: Campus Verlag, 1981, pp. 14–23.
- [14] W. R. Ashby, *An introduction to Cybernetics*. London: Chapman & Hall, 1956.
- [15] H.-J. Rheinberger, *Toward a history of epistemic things: Synthesizing proteins in the test tube*. Palo Alto: Stanford University Press, 1997.

GENERATING MUSICAL ACCOMPANIMENT THROUGH FUNCTIONAL SCAFFOLDING

Amy K. Hoover

Dept. of EECS
University of Central Florida
Orlando, FL 32816-2362 USA
ahoover@eecs.ucf.edu

Paul A. Szerlip

Dept. of EECS
University of Central Florida
Orlando, FL 32816-2362 USA
paul.szerlip@gmail.com

Kenneth O. Stanley

Dept. of EECS
University of Central Florida
Orlando, FL 32816-2362 USA
kstanley@eecs.ucf.edu

ABSTRACT

A popular approach to music generation in recent years is to extract rules and statistical relationships by analyzing a large corpus of musical data. The aim of this paper is to present an alternative to such data-intensive techniques. The main idea, called *functional scaffolding for musical composition* (FSMC), exploits a simple yet powerful property of multipart compositions: The pattern of notes and rhythms in different instrumental parts of the same song are *functionally related*. That is, in principle, one part can be expressed as a function of another. The utility of this insight is validated by an application that assists the user in exploring the space of possible accompaniments to pre-existing parts through a process called *interactive evolutionary computation*. In effect, without the need for musical expertise, the user explores transforming functions that yield plausible accompaniments derived from preexisting parts. In fact, a survey of listeners shows that participants cannot distinguish songs with computer-generated parts from those that are entirely human composed. Thus this one simple mathematical relationship yields surprisingly convincing results even without any real musical knowledge programmed into the system. With future refinement, FSMC might lead to practical aids for novices aiming to fulfill incomplete visions.

1. INTRODUCTION

An interesting aspect of creativity is that people are often better at discerning and evaluating novelty than at creating it [1, 2, 3]. While researchers define creativity in many ways and assert that creativity *should* be computable, no consensus yet exists on how algorithms should compose creative works [4, 5, 6, 7]. A key challenge for computational models of musical creativity is thus to identify guiding principles that facilitate its creation. The insight in this paper is that it is possible to exploit general mathematical properties of musical relationships and to leverage the intuitive human ability to evaluate novelty in the service of generating accompaniment without musical expertise.

Interestingly, some composers intentionally and explicitly incorporate mathematical transformations into their works. By translating, inverting, and reflecting musical lines they create canons, fugues, suites, and other styles [8]. However, transformations relating one part of a piece to another are often implicit. For example, Hellegouarch suggests that musicians implicitly incorporated logarithmic and modular transformations long before such operations were formally defined [9]. Similarly, Harkleroad [10] describes the relationship of change ringing to group theory although the musical technique predates the mathematical by centuries. The implicit nature of such transformations thus suggests that the mathematical relationship between musical parts is often intuitive.

An intriguing implication of this *intuitive* aspect of musical transformation is that it may be possible to formulate a *simple* theory that captures some important musical relationships. While much progress in recent years has focused on elucidating abstract relationships [11, 12] or extracting rules from statistical analyses [13, 14], the aim in this paper is to highlight a fundamental and simple insight that requires little data or analysis, in either the construction of the system or its datasets.

The main idea is based on the fact that music is in effect a function of time. That is, the pattern of pitches and the pattern of durations and rests can be expressed together as a vector function of time $\mathbf{f}(t)$ that outputs both pitch and rhythm information. Thus the part played by each instrument in an ensemble piece is the output of such a function. By casting instrumental parts as functions, the problem of accompaniment is illuminated in a useful light: Given an existing part $\mathbf{f}(t)$, the problem of formulating an appealing accompaniment translates to the problem of searching for accompaniment $\mathbf{g}(t)$ such that $\mathbf{g}(t)$ complements $\mathbf{f}(t)$.

Computationally, this problem is difficult if $\mathbf{g}(t)$ is sought *independently* of $\mathbf{f}(t)$ because the space of possible such functions is infinite and unstructured. Thus one approach is to enumerate a comprehensive set of *rules* that in effect describe how $\mathbf{g}(t)$ is constructed [15, 16]. However, the insight in this paper is simpler and interesting because it requires no musical analysis: The function $\mathbf{g}(t)$ can be constrained to a promising set simply by searching instead for $\mathbf{h}(\mathbf{f}(t))$, where \mathbf{h} outputs the accompaniment instead of $\mathbf{g}(t)$. In other words, we can exploit the fact that there must be a *functional relationship* between the accompaniment and preexisting parts, which are thus exploited as a

kind of *scaffold*. For example, the scaffold might contain bass and vocal parts, and the functionally-related accompaniment could be a guitar part. By simply searching for a function of the scaffold instead of a raw function of time, the accompaniment becomes constrained by the structure and contours of what is already written. The result is a pattern in time that in effect inherits the human style and character already in the scaffold, without any further analysis or rules. This approach, introduced in this paper, is called *functional scaffolding for musical composition* (FSMC).

A first step toward demonstrating the potential of FSMC is implemented in an interface that allows a human user to direct a search through the space of functional transformations of preexisting scaffolds. Each such transformation generates an accompaniment that the user rates. These ratings then drive a process called *interactive evolutionary computation* (IEC) that in effect allows the user to *breed* new accompaniments. Because each candidate in the search exploits the functional relationship between melody and harmony, the search quickly yields plausible accompaniments that inherit the human essence of the scaffold. In fact, participants in a listener study could not determine whether accompaniment generated through this method is computer-generated or not, even though no other musical knowledge is provided to the system, suggesting the significance of the insight behind FSMC. This result complements that in a companion paper [17], which focuses instead on establishing that the progression of accompaniments evolved through IEC indeed improves in quality.

A further interesting result of the research in this paper is that the functions that express the most convincing accompaniments are often surprisingly simple, implying that the veneer of complexity in the interplay between different musical parts may often be misleading. Perhaps in some cases our appreciation of rich musical tapestry is in its hidden simplicity, which FSMC can expose explicitly. On a practical level, the relative success of such a simple insight represents a possible first step towards more effective computational assistance in musical composition.

2. BACKGROUND

This section places FSMC in the context of other approaches to generating music and reviews a predecessor to FSMC called NEAT Drummer, which established the general principle of functional scaffolding.

2.1 Approaches to Music Generation

An important aspect of many traditional approaches to music generation is that they exploit musical corpora to discover relationships in the data to guide decision-making. For example, Ponsford et al. [18] learn probabilistic n-grams through analyzing local harmonies in a corpus. MySong also uses a hidden Markov model to generate accompaniments [19]. Chuan and Chew [20] combine statistical extraction from a corpus with known musical rules to produce songs in a particular style. While these models yield notable results, they require a significant database that must be carefully constructed. The idea in this paper

is to show that accompaniment can be produced simply, without any prior data or analysis.

In particular, the approach in this paper is based on *interactive evolutionary computation* (IEC), whereby a human collaborates with the computer to explore a space of candidates [21]. An example of IEC in music generation is GenJam, a jazz improvisation tool, which learns melodic measures and phrases through IEC [22]. By incorporating human evaluators, IEC helps reduce the need for building and analyzing data from a corpus. By further adding the idea of scaffolding in this paper, IEC is constrained to promising candidates.

2.2 NEAT Drummer

FSMC builds on previous work by Hoover et al. [23] and Hoover and Stanley [24] on NEAT Drummer, a system that creates percussion patterns for existing compositions. The drum generator “hears” the original parts in a MIDI composition simultaneously, and transforms these parts into a drum pattern that follows the contours of the original song. This transformation occurs through a function represented as a compositional pattern producing network (CPPN), which is a special type of artificial neural network (ANN) [25] that can take an arbitrary topology and wherein each neuron is assigned one of *several* activation functions.

NEAT Drummer users can explore drum patterns through IEC with NeuroEvolution of Augmenting Topologies (NEAT), a method for growing and mutating CPPNs [26]. Unlike traditional ANN learning, NEAT is a policy search method, i.e. it *explores* accompaniment possibilities rather than optimizing toward a specific target. While NEAT Drummer showed that the idea of functional scaffolding (implemented through CPPNs) can produce credible percussion accompaniment, it left open the question of whether such an approach can produce complete harmonization, which is the aim of this paper.

3. APPROACH: FUNCTIONAL SCAFFOLDING FOR MUSICAL COMPOSITION

Extending the idea in NEAT Drummer beyond just percussion, FSMC generates complete harmonies from existing compositions. These compositions form the scaffold from which accompaniments are built. However, unlike in NEAT Drummer, these scaffolds include rhythmic information *and* pitch information, thereby providing the foundation for harmonization.

To understand the idea behind FSMC consider that if different instrumental parts in the same composition were not related to each other at all, they would sound inappropriate together. Thus there is some relationship between different parts in the same piece. In effect, this *relationship* can be conceived as a *function* that describes how one part might be transformed into another. That is, theoretically there exists a function that can transform one sequence of notes and rhythmic information into another. If that function is simple then the relationship between the parts is more easily discernible than if the function is complex. Yet in any case, the important point is that there is *some* function that

relates these parts to each other. The idea in FSMC is to exploit this fact by literally *evolving the function* that relates one part to another. That way, instead of searching for a sequence of notes, FSMC can search for a transforming function that bootstraps off the existing parts (i.e. called the scaffold) to generate the accompaniment. In effect, FSMC is the hidden function that relates different parts of a composition to each other.

FSMC thus represents accompaniment as a *function* that transforms pitches and rhythms from input tracks (called the scaffold) into a temporal pattern interpreted as the accompaniment. In particular, this transforming function is encoded in FSMC by CPPNs [25], as explained in the next section. Outputs from CPPNs are interpreted as accompaniments that thereby follow contours of the original song. Users then interactively explore the space of such functions for personalized accompaniments through IEC.

3.1 Functional Relationship Representation

FSMC divides each musical part into a pitch pattern and a rhythm pattern, both of which are represented by separate CPPNs (figure 1). While CPPNs themselves are not essential to FSMC, they serve as convenient representations for exploiting the functional relationships between parts of a piece. The particular idea of separating pitch and rhythm follows a tradition in other approaches to music generation [6, 27]. The rhythm network, which extends the CPPN representation in NEAT Drummer, is shown in figure 1a. It has a set of scaffold inputs from the original composition (i.e. before accompaniment is added) and two output nodes for each instrument in the accompaniment: *OnOff* and *NewNote*. OnOff decides volume and whether or not the note will play. If the OnOff output returns a value below a given threshold, the accompaniment line rests at that tick. If OnOff indicates that a note is to be played, NewNote decides whether the note will be re-struck or sustained. In partnership with the rhythm CPPN, the pitch CPPN in figure 1b sees the pitches of instruments in the scaffold and decides accompaniment pitch with a single output. Viable pitches are discretized into bins that correspond to the given key and the network thereby plays the pitch closest to its output. The CPPNs in figure 1 act just like ANNs with weighted connections and hidden neurons that transform the scaffold input at the current timestep into rhythm and pitch accompaniment.

The CPPN representation in figure 1 thus in effect implements the idea of functional scaffolding. The CPPN is itself just a formalism for specifying a function that can be artificially evolved. The inputs to the CPPNs are the pattern of notes and durations within the scaffold and the outputs are the accompaniment. In this way, the CPPN is literally a function of the scaffold that transforms it into a functionally-related accompaniment pattern.

Figure 2a shows an example of a temporal pattern that is input into the rhythm CPPN. This scaffold is four measures of a repeating quarter-quarter-half note motif. To impart a sense of time within a note, when a note begins, an attack spike is sent to the network for that particular instant in time. This spike decays linearly over time for the dura-

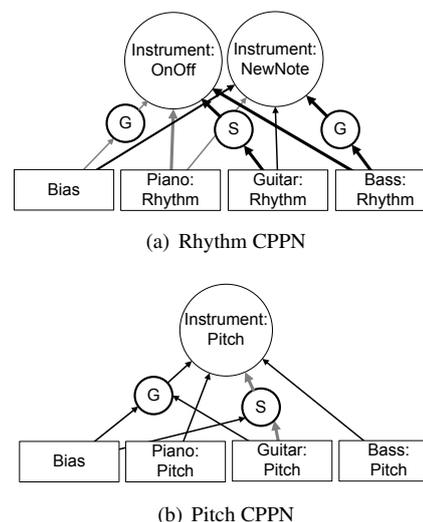


Figure 1. How CPPNs Transform the Input Scaffold.

The rhythm CPPN in (a) and pitch CPPN in (b) together form the accompaniments of FSMC. The inputs to the CPPN are the scaffold rhythms and pitches for the respective networks and the outputs indicate the accompaniment rhythms and pitches. Each rhythm network has two outputs: *OnOff* and *NewNote*. The OnOff node controls volume and whether or not a note is played. The NewNote node indicates whether a note is played or sustained at the current tick. If OnOff indicates a rest, the NewNote node is ignored. The pitch CPPN output decides what pitch the accompaniment should play at that particular instant of time. The internal topologies of these networks, which encode the functions they perform, change over evolution.

tion of the scaffold note. This spike-decay representation of time ensures that the position *within* the particular note is known to the rhythm network at any given time, thereby providing rhythmic context from the scaffold to the accompaniment. Thus it can output patterns based on the rhythmic information in the scaffold. Simultaneously, the pitch from the scaffold at each discrete instant in time is sent modulo 12 to the pitch CPPN (figure 2b), whose output is converted to one of eight pitches in the specified key.

The hidden nodes in the CPPNs depicted in figure 1 are added by mutations that occur over the evolutionary process. They in effect increase the complexity of the transforming function by adding intervening nonlinearities. For example, the Gaussian function (depicted as a “G”) introduces symmetry (i.e. such as the same sequence of notes ascending and then descending) and the sigmoid (depicted as “S”) is nonlinear yet asymmetric. As in a neural network, the connections are weights (i.e. coefficients) that are multiplied by their inputs. By accumulating such transformations, the relationship between scaffold and accompaniment can become more complex. In effect, the CPPN and its inputs provide the *functional scaffolding* in FSMC. The next section explains how such preferences are conveyed through the evolutionary process.

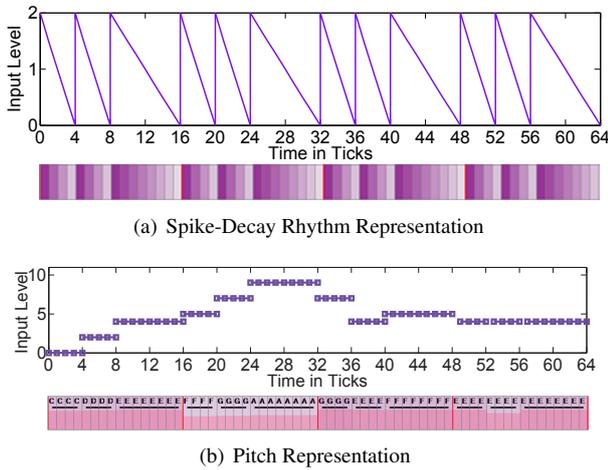


Figure 2. CPPN Input Representation. The spike-decay representation for rhythms is shown in (a) and the pitch representation is shown in (b). Both such inputs are depicted in two ways: The first is a continuous-time graph that shows decaying spikes for rhythm and the pitch level for pitch. The second is a discrete-time representation of what is actually input into the network at each discrete timestep, which is represented by darkness for rhythm and height for pitch. Because the network samples time discretely, results in this paper are also depicted in the discrete-time format. In this way, this figure gives a sense of exactly what the CPPN “hears” (for each instrument in the scaffold) as it generates accompaniment.

3.2 Exploring Functional Space

Through IEC, the FSMC accompaniment gains its musical character from a combination of the human breeder and the information in the scaffold, rather than through a corpus. In the iterative IEC process, the FSMC interface presents a population of candidate accompaniments to the user who then rates each candidate. Pieces are judged good, mediocre, or bad. The next generation then presents to the user accompaniment candidates from CPPNs derived from those judged best in the prior generation. The new generation is created by mutating and mating network weights from the parent CPPNs and occasionally through mutating network structure by adding or subtracting parts of the network, following the NEAT method [26].

Users also influence the accompaniment by choosing CPPN inputs, i.e. to which instruments in the scaffold the CPPNs should listen.

4. EXPERIMENT

The experiment is divided into two parts: First accompaniments are evolved for two songs and then a listener study assesses how convincing the accompaniments are.

4.1 Accompaniments

To demonstrate the capabilities of FSMC, accompaniments were evolved for two pieces in this paper. The program and accompaniment CPPNs will be released in summer of 2011 at <http://eplex.cs.ucf.edu/fsmc>. The

two pieces are folk songs (Nancy Whiskey and Bad Girl’s Lament) originally arranged in MIDI format by Barry Taylor and redistributed with his permission. The MIDI format is convenient because it is easy to convert directly into the FSMC input format (figure 2).

It is important to note that these pieces are chosen for this experiment because they exemplify entirely human compositions that meet a minimum standard of recognizable quality. While in the future FSMC can potentially generate accompaniment for incomplete compositions by amateur musicians, by starting with pieces that are convincing as complete compositions, it is possible to discern whether the generated accompaniments reduce the human plausibility of the work, or whether they complement it successfully, as would be hoped for such an approach.

The interactive evolutionary process for the two example pieces was guided by the authors. No musical knowledge was applied beyond simply choosing which candidates sounded best. The process proceeded as follows: A set of ten *random* CPPNs corresponding to an initial population of FSMC accompaniments was first generated by the program. Among these, those that sounded best were selected by the user. From the selected candidates a new generation of CPPNs was created that are offspring (i.e. mutations and crossovers) of the original generation. This process of listening to candidates, selecting the best, and creating new generations was repeated until a satisfactory accompaniment appeared. While user input is an important aspect of this process, no session lasted more than 12 generations (i.e. no more than 12 preference decisions were ever made), highlighting the overriding importance of the FSMC relationship to constraining accompaniments to a reasonable set of candidates. Thus, interestingly, in contrast to data-intensive approaches, the only knowledge needed to generate accompaniments through this approach is imparted in ten to 15 clicks of IEC.

Accompaniments are evolved with a CPPN mutation rate and crossover rate of 0.3. The NewNote threshold is also 0.3. Furthermore, when the OnOff output in the rhythm network (which also indicates volume) falls below 0.3, no note is played. The next section explains a study designed to assess the results.

4.2 Listener Study

To gain insight into the potential of the approach, the results in this paper are assessed through a listener study in which anonymous participants were asked to rate examples with and without FSMC accompaniments. The key focus in the study is on whether the fact that a computer is involved in generating some of the examples can be discerned by the listeners. Thus the survey is a kind of *musical Turing Test*. This perspective is interesting because FSMC is based on no musical principle or theory other than establishing a functional relationship; if such a minimalist approach can generate plausible accompaniment it suggests that the theory behind it is at least promising.

A total of 66 listeners, all of whom are students in a diversity of majors at the University of Central Florida, participated in the study. The full survey, including the

human compositions, is provided at <http://eplex.cs.ucf.edu/fsmc/smc2011/survey>. The aim is to discover whether the accompaniments sound either natural or computer-generated. Participants are asked to rate five different MIDIs by answering the following question:

Based on your impression, how likely is it that any of the instrumental parts in the musical piece found at the following link, <link>, were composed by a computer? “Composed” means that the computer actually came up with the notes, i.e. both their pitch and duration, on its own. (1 means very unlikely and 10 means very likely).

The participants rated a total of five MIDIs: (1) an obviously computer-generated control (which helps to establish that participants understand the question), (2) a version of Nancy Whiskey with a computer-generated accompaniment, (3) fully human-composed Chief Douglas’ Daughter, (4) fully human-composed Kilgary Mountain, and (5) a version of Bad Girl’s Lament with a computer-generated accompaniment. Thus the main issue is whether participants judge piece 2 and piece 5, which have accompaniments evolved with FSMC, as distinguishable from piece 3 and piece 4, which are entirely composed by humans.

5. RESULTS

This section begins with an analysis of the two evolved FSMC accompaniments and then presents the user study. All music discussed in this section, both with and without evolved accompaniments, can be heard at <http://eplex.cs.ucf.edu/fsmc/smc2011>.

5.1 Accompaniments

Figure 3 shows results after two generations of evolving accompaniment for Nancy Whiskey and 12 generations of evolving accompaniment for Bad Girl’s Lament. The low number of generations necessary to obtain these results is a result of the strong bias provided by FSMC towards generating accompaniments related to the scaffold. A key issue in understanding the results is the functional relationship between scaffold inputs and CPPN outputs over time, which gives a sense of the implication of linking these parts functionally. To help visualize this relationship, the top line of figure 3a and 3b contains a series of rectangles read from left to right that represent the CPPN output at that particular tick. Rectangle shading indicates pitch and volume: *Darker* color shading represents louder notes while *taller* shading indicates higher pitch. Note names are also written in bold at the top of each rectangle. Notes can be sustained from previous ticks, re-struck, or silenced. A thick horizontal line crossing the borders between ticks indicates a sustain while gaps with thin horizontal lines (slightly lower than the thick lines) indicate rests.

Figure 3 also shows both the rhythm and pitch *inputs* (i.e. the scaffold) to the CPPN. It is important to note that rhythm inputs represent the special spike-decay rhythm format introduced in figure 2a while pitch inputs are simply pitch levels, as in figure 2b.

Figure 3a shows four measures, numbered 3, 4, 5, and 6, of generated accompaniment for the MIDI scaffold, Nancy Whiskey. To also provide perspective in musical notation, figure 4 shows measures five and six of the score. The fiddle, steel guitar, and bass from Nancy Whiskey are input to both the pitch and rhythm networks; the output is a harpsichord accompaniment that inherits pitch and rhythm relationships from the scaffold. One salient such relationship is at the endings in measures four and six where a G is played in the fiddle part. Although the accompaniment does not always follow the fiddle, at this point the output accompaniment also plays G at the same time, although an octave lower. However, while it follows the pitch, the accompaniment varies the rhythm at this part, borrowing rhythmic elements from the other instruments in the scaffold, thereby further differentiating itself from the fiddle. In totality, the accompaniment incorporates pitch and rhythmic elements from all three scaffold instruments while also varying and combining them in novel ways, yielding an original pattern that complements the whole.

FSMC can be applied to any scaffold. An additional evolved accompaniment for a different scaffold, Bad Girl’s Lament, is shown in figure 3b. This image shows measures 1 and 2 of the accompaniment in the first section followed by measures 13 and 14. In measures 1 and 2, the pitches follow the harpsichord input in the pitch network exactly, but the rhythms are different. In the harpsichord part of the scaffold, each note is held until the next pitch sounds. Thus there are no rests or note rearticulations. However, the accompaniment adds new flair to these pitches by inventing a rhythm that is influenced by the rhythm of the harpsichord in the rhythm network. In the generated rhythm, the note is held for three ticks, rests on the fourth tick, plays on the fifth tick, and rests on the sixth tick. This pattern repeats throughout measures 1 and 2. Measures 13 and 14 in figure 3b show how pitches in the accompaniment sometimes also deviate from the scaffold input. Notice that when the harpsichord input plays an A, the accompaniment sounds a C#. Similarly, the accompaniment sounds D when a C# is heard in the scaffold.

Figure 5 shows the internal structure of the evolved CPPNs that produce these accompaniments. In the Nancy Whiskey rhythm CPPN (figure 5a), each input is connected to the two outputs with different weights. In the pitch network, each input is directly connected to the pitch output with the exception of the second steel guitar input, which connects through a hidden node with a Gaussian function. Recall that output nodes compute a sigmoid function of their input.

Interestingly, the CPPNs for generating Bad Girl’s Lament accompaniment are even simpler and did not even evolve any hidden nodes (i.e. additional nonlinearities beyond the sigmoid output functions). The simplest CPPN of all, which is the pitch network in figure 5d, has only a single connection. Such relationships encoded by the CPPNs can also be written mathematically. For example, the Nancy Whiskey rhythm CPPN (figure 5a) computes $\text{OnOff} = \sigma(-.35n_1 + 1.34n_2 - 1.76n_3 + 1.46n_4)$ and $\text{NewNote} = \sigma(1.01n_1 + 1.70n_2 - .37n_3 + .51n_4)$,

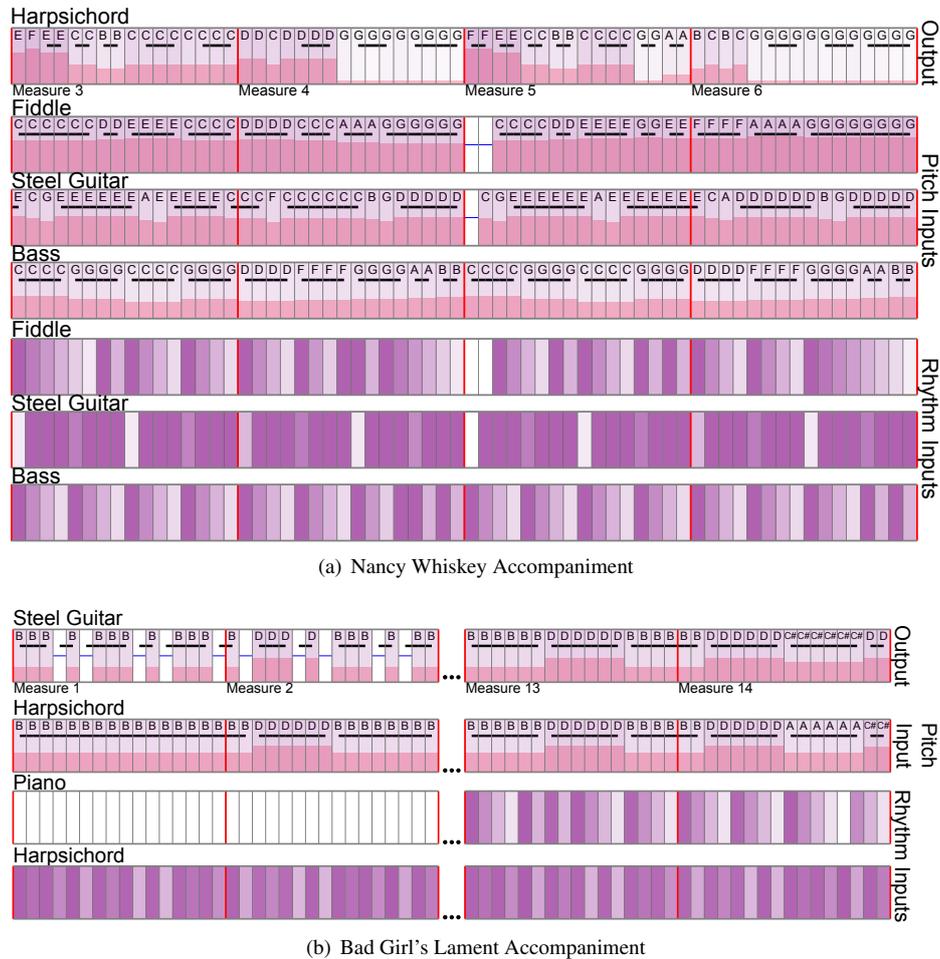


Figure 3. Accompaniments Generated by FSMC. Accompaniments evolved for Nancy Whiskey and Bad Girl's Lament are shown in (a) and (b). Each accompaniment can be heard at <http://eplex.cs.ucf.edu/fsmc/smc2011>. The accompaniments are produced by CPPNs that relate the scaffold inputs to the generated outputs. The Nancy Whiskey accompaniment shown in (a) is also shown in staff form for measures 5 and 6 in figure 4. The rhythm and pitch relationships between the scaffolds and accompaniments derive from the functional relationship between them.

where $\sigma(x) = \frac{1}{1+e^{-1.1x}}$, and n_i is node number i . In this way, musical relationships really are being encoded as functions. It is important to understand that the simplicity of these relationships resulted from a process of human selection through IEC that ended when the human was satisfied, which means it reflects the human user's implicit preferences. These results show that simple relationships can be appealing and convincing. In this way, this kind of application can tell us something about the nature of the implicit musical relationships that we appreciate.

5.2 Listener Study

The complete results of this study are shown in table 1. On average, the 66 participants judge the intentionally poor example as significantly more likely ($p < 0.001$ according to Student's t-test) to be computer-generated than any other song in the survey. This difference indicates that participants understand the survey.

Although the accompanied Nancy Whiskey is judged significantly more likely ($p < 0.05$) to be computerized than the human song Chief Douglas' Daughter, it is not judged significantly more likely than Kilgary Mountain to

| Survey Results | | |
|---------------------------------------|------|-----------|
| MIDI Name | Mean | Std. Dev. |
| Control | 7.82 | 2.15 |
| <i>Nancy Whiskey with Accomp.</i> | 5.45 | 2.65 |
| Chief Douglas' Daughter | 4.32 | 2.61 |
| Kilgary Mountain | 4.86 | 2.39 |
| <i>Bad Girl's Lament with Accomp.</i> | 4.82 | 2.44 |

Table 1. Survey Results (lower means more human-like). The average ratings and standard deviations for the samples show that FSMC accompaniments can sound human. The Bad Girl's Lament MIDI, which is partly computer-generated, ranks less likely to be computer-generated than the fully human-composed song, Kilgary Mountain, although this difference is not significant.

be computerized. This result indicates that the accompanied Nancy Whiskey can pass the musical Turing test, i.e. people cannot distinguish it from a song that is entirely human-generated.

The Bad Girl's Lament accompaniment is even more difficult for participants to differentiate. It is not judged sig-

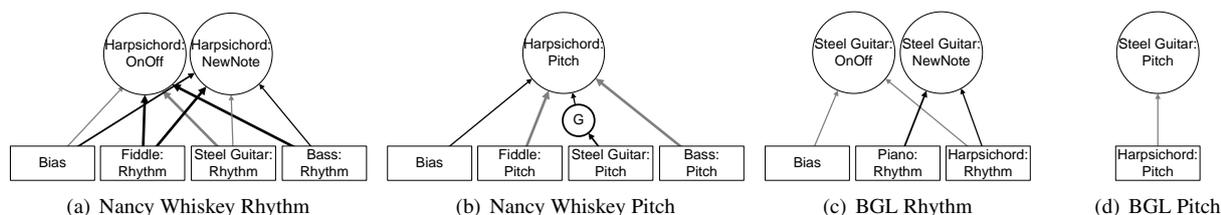


Figure 5. CPPNs Corresponding to Accompaniments. CPPNs for generating accompaniment for Nancy Whiskey (a and b) and Bad Girl’s Lament (BGL; c and d) are shown above. Each accompaniment can be heard at <http://eplex.cs.ucf.edu/fsmc/smc2011>. Line thickness is proportional to weight and gray lines are inhibitory. The simplicity of the CPPNs shows that the functional relationship between different musical parts is often plausible even if it is simple.



Figure 4. Nancy Whiskey Accompaniment Generated by FSMC in Staff Notation. Measures 5 and 6 from the score and the generated accompaniment illustrate FSMC accompaniment in staff notation. The top line shows the generated harpsichord output while the next three voices show the fiddle, steel guitar, and bass respectively.

nificantly more likely to be computer-assisted than either of the human pieces, i.e. Chief Douglas’ Daughter or Kilgary Mountain. In fact, on average, accompanied Bad Girl’s Lament scored slightly *less* likely to be computerized than the entirely human song Kilgary Mountain.

These results validate that evolved accompaniments are at least plausible enough to fool human listeners into confusing partly computer-generated compositions with fully human-composed ones, even though FSMC has almost no a priori musical knowledge programmed into it.

6. DISCUSSION AND FUTURE WORK

The primary contribution of this paper is to show how little prior information is necessary to generate plausible accompaniments. While most generation methods require an extensive corpus to analyze or prior music knowledge [6, 18, 20], this study shows that simple functional relationships are central to musical composition and therefore possible to exploit. The listener study confirms that such accompaniments can be indistinguishable from fully-human compositions by average listeners.

Not only is the insight simple that a functional relationship is foundational to the idea of accompaniment, but the evolved relationships between the accompaniments and the scaffold themselves also turned out to be simple. Nancy Whiskey contains a single hidden node in its rhythm and pitch CPPNs, indicating at most two function compositions. Bad Girl’s Lament accompaniment is even simpler.

Of course, these results are anecdotal and do not imply that complex relationships would not be appealing in some cases, yet they do raise the intriguing hypothesis that many such relationships could be simpler than they sound intuitively. In the future, it will be interesting to discover in which cases evolution leads to more complex relationships between scaffold and accompaniment.

Interestingly, because the evolved patterns are encoded as transformative functions (i.e. CPPNs), in principle such a function can be applied to a *different* scaffold, which is like transferring an accompaniment “personality” to a new composition. In this way, evolved CPPNs may ultimately apply effectively to multiple pieces of similar genre, which will be explored in the future.

While incorporating at least some musical knowledge into FSMC may ultimately be necessary for widespread application, the contribution so far is to provide a core insight around which further structure can later be built. For the moment, the result that listeners cannot detect the difference between songs with computer-generated accompaniment and those without it suggests that the simple idea of exploiting functional relationships provides a promising starting point for a future research direction.

7. CONCLUSION

This paper provides insight into the relationship between musical parts through a new theory called functional scaffolding for musical composition (FSMC). By representing the relationship between existing parts and a new accompaniment functionally, plausible accompaniments are generated. The resulting pieces with accompaniments are even sometimes indistinguishable from fully-human compositions. The main conclusion is that FSMC provides an alternative to data-intensive approaches to music generation and analysis that nevertheless promises a different kind of insight into the nature of musical accompaniment.

Acknowledgements

This work was supported in part by the National Science Foundation under grant no. IIS-1002507. Special thanks to Barry Taylor for granting special permission to utilize his own MIDI productions of folk music in this work. Barry Taylor originally sequenced Nancy Whiskey and Bad Girl’s Lament (without accompaniment).

8. REFERENCES

- [1] P. J. Silvia, "Discernment and creativity: How well can people identify their most creative ideas?" *Psychology of Aesthetics, Creativity, and the Arts*, vol. 2, no. 3, pp. 139–146, 2008.
- [2] R. J. Sternberg, "The nature of creativity," In *The Essential Sternberg Essays on Intelligence, Psychology, and Education*. Springer, LLC, 2009.
- [3] M. L. Maher, "Evaluating creativity in humans, computers, and collectively intelligent systems," In *Proc. of the 1st DESIRE Network Conference on Creativity and Innovation in Design*, 2010, pp. 22–28.
- [4] P. N. Johnson-Laird, "Freedom and constrain in creativity," In *The Nature of Creativity: Contemporary Psychological Perspectives*. Press Syndicate of the University of Cambridge, 1988.
- [5] R. C. Shank, "Creativity as a mechanical process," In *The Essential Sternberg Essays on Intelligence, Psychology, and Education*. Press Syndicate of the University of Cambridge, 1988.
- [6] D. Cope, *Computer Models of Creativity*. Cambridge, MA: MIT Press, 2005.
- [7] A. Dorin and K. B. Korb, "Improbable Creativity," In *Proc. of the Dagstuhl International Seminar on Computational Creativity*, 2009.
- [8] W. Hodges and R. J. Wilson, "Musical patterns," In *Mathematic and Music: A Diderot Mathematical Forum*. Springer-Verlag, 2002, ch. 5.
- [9] J.-C. Risset, "Computing musical sound," In *Mathematic and Music: A Diderot Mathematical Forum*. Springer-Verlag, 2002, ch. 13.
- [10] L. Harkleroad, *The Math Behind the Music*, ser. Outlooks. Cambridge University Press, 2006.
- [11] S. Cannon, "Maximally smooth diatonic trichord cycles," In *Proc. of the 2nd International Conference on Mathematics and Computation in Music*, 2009.
- [12] D. Tymoczko, "Three conceptions of musical distance," In *Proc. of the 2nd International Conference on Mathematics and Computation in Music*, 2009.
- [13] C. Rhodes, D. Lewis, and D. Müllensiefen, "Bayesian model selection for harmonic labelling," In *Proc. of the 1st International Conference on Mathematics and Computation in Music*, 2007.
- [14] K. M. Kitani and H. Koike, "Improvgenerator: Online grammatical induction for on-the-fly improvisation accompaniment," In *Proc. of the 2010 Conf. on New Interfaces for Musical Expression (NIME 2010)*, 2010.
- [15] M. Dahia, H. Santana, E. Trajano, G. L. Ramalho, C. Sandroni, and G. Cabral, "Using patterns to generate rhythmic accompaniment for guitar," In *Proc. of the International Conference on Sound and Music Computing*, 2004, pp. 111–115.
- [16] W. F. Walker, "A computer participant in musical improvisation," In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1997, pp. 123–130.
- [17] A. K. Hoover, P. A. Szerlip, and K. O. Stanley, "Interactively evolving harmonies through functional scaffolding," In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2011)*. New York, NY: ACM, 2011, to appear.
- [18] D. Ponsford, G. Wiggins, and C. Mellish, "Statistical learning of harmonic movement," *Journal of New Music Research*, vol. 28, no. 2, pp. 150–177, 1999.
- [19] I. Simon, D. Morris, and S. Basu, "MySong: automatic accompaniment generation for vocal melodies," In *Proc. of the Twenty-Sixth Annual SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2008, pp. 725–734.
- [20] C.-H. Chuan and E. Chew, "A hybrid system for automatic generation of style-specific accompaniment," In *4th International Joint Workshop on Computational Creativity*, 2007.
- [21] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," *Proc. of the IEEE*, vol. 89, no. 9, pp. 1275–1296, Sep 2001.
- [22] J. A. Biles, "Interactive genjam: Integrating real-time performance with a genetic algorithm," In *Proc. of the 1998 International Computer Music Conference*, 1998.
- [23] A. K. Hoover, M. P. Rosario, and K. O. Stanley, "Scaffolding for interactively evolving novel drum tracks for existing songs," In *Proc. of the Sixth European Workshop on Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMUSART 2008)*, 2008.
- [24] A. K. Hoover and K. O. Stanley, "Exploiting functional relationships in musical composition," *Connection Science Special Issue on Music, Brain, & Cognition*, vol. 21, no. 2, pp. 227–251, June 2009.
- [25] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic Programming and Evolvable Machines Special Issue on Developmental Systems*, vol. 8, no. 2, pp. 131–162, 2007.
- [26] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, pp. 99–127, 2002.
- [27] P. Johnson-Laird, "Jazz improvisation: A theory at the computational level," In *Representing Musical Structure*, ser. Cognitive Science Series. London, UK: Academic Press Limited, 1991, ch. 9.

A RULE-BASED GENERATIVE MUSIC SYSTEM CONTROLLED BY DESIRED VALENCE AND AROUSAL

Isaac Wallis, Todd Ingalls, Ellen Campana, and Janel Goodman

School of Arts, Media, and Engineering

Arizona State University

Tempe, AZ, USA 85281

{iwallis | testcase | ecampana | jgoodman}@asu.edu

ABSTRACT

This paper details an emotional music synthesis (EMS) system which is designed around music theory parameters and previous research on music and emotion. This system uses a rule-based algorithm to generate the music from scratch. Results of a user study on this system show that listener ratings of emotional valence and arousal correlate with intended production of musical valence and arousal.

1. EMOTIONAL MUSIC SYNTHESIS

Historically, music has been used in ceremonial, religious, and artistic settings for the purpose of affecting listeners emotionally and setting a mood. Emotional music synthesis is the computerized generation of music which has a recognizable emotion, for the purpose of setting a mood. Music can influence the emotions of listeners in multiple ways [1], but most EMS systems use one of two psychological mechanisms: (1) they manipulate musical expectations, or (2) they manipulate structural or performance features of the music in order to set up a perceptible mood.

In the approach manipulating musical expectations, the melody, harmony, and rhythm of the music are designed so that listeners will begin to expect specific musical events. Sometimes these expectations stem from standard musical theory: examples of these include harmonic cadences, in which specific chord patterns (e.g. the V-I resolution) are common. If these expectations are satisfied, listeners will experience a sense of resolution. If these expectations are not satisfied, listeners may experience emotions such as surprise or of being “left hanging.” [2]

In the structural mood-based approach, features of the music are changed in order to create the perception that the music itself has a mood. Over the last century, music psychologists have identified the emotional effects of many musical features such as tempo or harmonic mode; a large list of these features and their mappings can be found in Gabriellson and Lindström [3]. However, most of these studies were performed using western forms of music, such as classical, and enlisted participants who were familiar with those forms of music.

Musical features, and their emotional mappings, can be thought of using a framework of emotional mirroring: when we perceive emotion in an agent, whether that agent be a person, a pet, a movie, or a piece of music, what we are really doing is performing a structural analysis on that agent. Some component of the structural analysis answers the question: “What emotions would drive me, personally, to exhibit these structural features?” If we hear someone singing sweetly we assign a pleasant emotion to that music—the same emotion we would feel if singing that way ourselves. When we hear fast-paced music we assign it the same high-energy emotions we might feel if we were to move our bodies at a fast pace, or watch a movie with a fast-paced editing style, or even drive a car at high speed. The structural feature moving rapidly is applicable to many situations, and results in similar emotional connotations.

There are limits to the emotional resonance of EMS. For example, musical emotions will not always be internalized by listeners. This makes sense in the context of day-to-day life—if the musical mood were always internalized, no one would choose to listen to sad or angry music. It is possible that expectation-based EMS, since it deals with internal expectations of listeners, could be more successful in affecting felt emotions. Another limit is that complex emotions such as shame or jealousy are difficult to convey by music alone, though some evidence suggests that this is not impossible [4]. Narrative in the form of words or visuals must usually be included in order to supply the necessary emotional focus; without context, there is not much to feel jealous of [5]. Also, although evidence exists to suggest that listeners may internally feel multiple conflicting emotions in response to music [6], there is none to suggest that music is capable of having, in itself, simultaneous conflicting moods.

EMS is a fairly new field of interactive music, but examples of working EMS systems do exist. Friberg’s pDM system takes, as input, pre-composed musical note-lists and outputs emotionalized versions of them by manipulating features of their performance [7]. Winter created a system expanding on pDM which also manipulates harmonic features of the music in real-time [8]. Livingstone et al. [9] took an approach similar to Friberg and Winter. Oliveira and Cardoso [10] are in the process of developing an EMS system in which the musical features to be manipulated were derived through analysis of a corpus of emotion-tagged music files. They intend to use these fea-

tures to transform the emotion in other pieces of music.

Transformative EMS systems like these take pre-existing musical compositions as input, which are then transformed emotionally and output. The mood of the resulting music is different, but the music itself remains recognizable. This document presents an EMS system which generates music from scratch using rule-based algorithmic composition.

2. SYSTEM DESIGN

This paper builds on previous work by evaluating an EMS system which is described in depth in [11]. This section provides a brief description of the parameters and design of this EMS system.

An EMS algorithm was designed to emulate piano accompaniment techniques. Models of emotion and musical emotion were explored for insight into algorithm interfaces. The most rigorously developed and accurate such models tend to be multidimensional models such as the Geneva Emotion Music Scale (GEMS) [6]. GEMS has nine dimensions: wonder, transcendence, tenderness, nostalgia, peacefulness, power, joyful activation, tension, and sadness. These nine dimensions can be factored down to three: sublimity, vitality, and unease. This model was designed for classification of induced musical emotions only, and no attempt was made to form relationships between musical features and emotions.

Although models like GEMS point to an interesting area of interface research, at this stage of our research we decided to use a simpler model as our interface. One of the simplest scientifically validated emotional models is the Circumplex Model [12], a two-dimensional model where the Y-axis corresponds to emotional arousal or intensity and the X-axis corresponds to emotional pleasure or valence. Common emotion words, if rated along these axes by enough people, will fall in a ring around the origin with words like anger in the upper left, joy in the upper right, serenity in the lower right, and sadness in the lower left. Since it is a two-dimensional model, it allows the implementation of a simple canvas interface—depicted in Figure 1—allowing users to select valence and arousal with a single click. This style of interface has been used in other EMS systems, such as pDM [7].

The circumplex model is intuitive and easy to use in interfaces, but it is not entirely emotionally accurate. For example, anger and fear are close to one another on the circumplex, but most people perceive these as very different emotions; far more different than other closely-spaced emotions such as sadness and depression. The only way to overcome this drawback is adding more dimensions to the model, such as a dominance axis separating anger and fear [13]; the logical result of this is a complex model such as GEMS.

Ten musical parameters—pitch register, loudness, rhythmic roughness, tempo, articulation, harmonic mode, and upper extensions—were developed using Gabrielsson and Lindström’s work [3] as a guide, then fine-tuned by ear. As most of the musical features in Gabrielsson and Lindström’s review are standard concepts from music theory, Persichetti [14] was also a useful resource in parameter de-

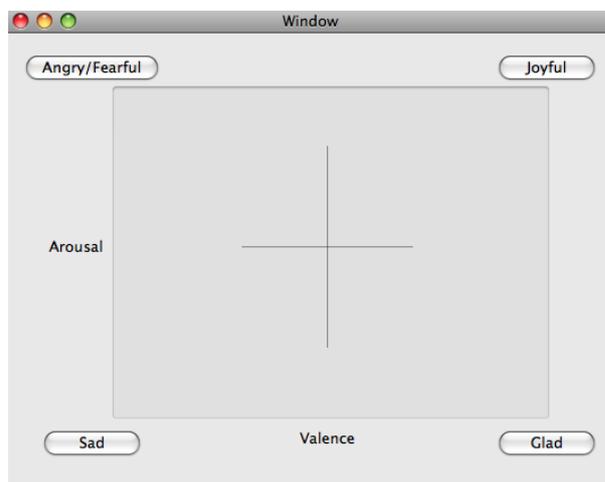


Figure 1. Clickable valence/arousal interface.

velopment. These musical parameters were attached to the valence and arousal of the interface. This algorithm needs direct control in real-time of information regarding rhythm and timing, harmony, and note selection (it does not have melodic parameters because it is designed for chordal piano accompaniment). The following subsections describe these ten musical parameters in greater detail.

2.1 Rhythmic/Timing Parameters

The rhythmic module has three lower-level parameters attached to it: rhythmic roughness, tempo, and articulation. Rhythmic roughness is a musical feature first studied by Gundlach [15]. This parameter determines the variation in note lengths over a measure of music: if all notes are of equal duration, roughness is low, and if notes are of varying length, roughness is high. In this system, at the lowest roughness each measure is populated with sixteen equal-length notes. As roughness increases, pairs of notes are selected at random and joined into single notes (Figure 2). This means that lower roughness correlates with higher note density over time, which has profound effects on the perceived rapidity of the music.

Tempo and articulation are well-known musical timing concepts. In this system, tempo is a global speed value (in beats-per-minute) affecting how quickly the notes of each measure is played—there are four beats per measure. A chord change also takes place upon each measure. Articulation refers to the actual duration each note is played in comparison with its allotted time. Notes with long articulation (i.e. legato) take up all the time available between



Figure 2. (a) Smoothest rhythm, consisting of sixteen equal-length rhythmic events. (b) Rougher rhythm, in which four of the sixteen events are joined at random. (c) Roughest rhythm, in which five additional events (nine total) are joined.

the previous note and the upcoming one, while notes with short articulation (i.e. staccato) take up just a fraction. In this system, when the articulations are set to the longest setting, notes will overlap by extending beyond the allotted times.

2.2 Harmonic Parameters

Two musical parameters were harmonic in nature: upper extensions and harmonic mode. These are each well-known musical terms, although upper extensions is a concept more commonly applied in jazz than other forms of music.

2.2.1 Harmonic Mode

The harmonic mode parameter is discrete, switching between six of the natural harmonic modes: Lydian, Ionian, Mixolydian, Dorian, Aeolian, and Phrygian. Locrian is left out; it is less commonly used in classical or jazz modal composition because it has a dissonant tonic chord. When the parameter changes and a new mode is selected, all chords in the progression are replaced by chords from the new mode which serve the same chord functions—tonic, subdominant, or dominant—as the original progression. In this way, the new progressions sound similar to the old progression, but the flavor of the new mode is introduced. This EMS algorithm is distinctive in its use of six harmonic modes; to our knowledge, other EMS algorithms incorporate only the major and minor modes. This parameter is mapped to valence so that the darkest mode¹ is at the lowest valence setting and the brightest mode is at the highest valence.

2.2.2 Upper Extensions

This system uses chord progressions made of triads. Triads are chords consisting of a scale degree plus the notes at the intervals of a third and a fifth above that root scale degree. On any triad within a mode, there are somewhere between two and four notes—forming seventh, ninth, eleventh or thirteenth intervals with the root—which can be played without forming discordant intervals with any of the three original triadic notes. These extra notes are called upper extensions, and when the upper extensions parameter increases, more of these notes are allowed into the chords played by the EMS system.

2.3 Note Generation

The system tempo controls the rate of an underlying chord progression which is partially determined by the harmonic parameters. When the system needs a new note, any note from the underlying chord could be generated, subject to a few note generation parameters. The first two parameters deal with the volume and thickness of each musical event. The loudness parameter specifies average note velocity, and the voicing size parameter determines how many notes are used simultaneously upon each event; or, put another way, how many fingers the virtual pianist uses on each chord. Voicing size is perceptually confounded

¹ According to Persichetti, the modes in order from darkest to lightest are: Locrian, Phrygian, Aeolian, Dorian, Mixolydian, Ionian, and Lydian [14].

with loudness, because playing more simultaneous notes will lead to a louder overall sound.

In addition, there are three parameters related to the rule-based note generation in the system: voice spacing, voice leading, and pitch register. Whenever a note is needed, the system selects a chord-tone from the underlying chord in a probabilistic way, with the probabilities being weighted by the three rules. Since the rules conflict with one another, the result is a music generator which is unpredictable, yet remains within certain musical constraints.

2.3.1 Rule 1: Pitch Register

The new note should be selected from a range which is determined by the pitch register parameter, and should tend to be more frequently selected from the center of this range. This keeps the music from going too high or too low.

2.3.2 Rule 2: Voice Spacing

As this is a piano emulation system, new notes should not play on top of notes which are already playing. In addition, new notes should tend to avoid playing near existing notes to a degree specified by the voice spacing parameter. As this parameter increases, the area of avoidance around existing pitches widens.

2.3.3 Rule 3: Voice Leading

Pianists often follow a principle called voice-leading where new chords are voiced to be as similar as possible, in terms of the intervals and placement on the keyboard, as previous chords; this minimizes the required arm and finger movement. We capture this phenomenon by specifying that new notes will tend to be generated where other notes have been recently released. Notice that this rule conflicts with the previous rule: existing notes repulse new notes, but once those existing notes are released, new notes are attracted to the same area. This conflict would lead to a directionality of note generation, where new notes are generated at an increasingly high or low pitch, but the first rule eliminates this possibility.

2.4 Mappings

Table 1 and Figure 3 show how parameters are manipulated in the EMS system. Each parameter is linearly mapped to valence or arousal in the most continuous possible way; for example, harmonic mode thresholds the valence selection space into six equal regions, while loudness increments between low and high arousal settings. The system is designed so that musical parameters can be easily re-scaled or re-mapped, so the values in Table 1 represent the system in only one configuration. In practice, often a scaled-down subset of parameters was used, in which the voice spacing and voice leading parameters were fixed at a central value across all valence settings. In preliminary study, these simplifications seemed to have little effect on the emotional connotations of the system. Tempo was also fixed at 80 bpm across all arousal settings; the rhythmic roughness parameter alone was sufficient to affect the perceived rapidity and arousal of the music.

| | Valence Min | Valence Max |
|----------------|-------------------------------|-------------------------------|
| Mode | Phrygian | Lydian |
| Extensions | 2-4 Added Pitches | No Added Pitches |
| Pitch Register | Centered on C4 | Centered on C6 |
| Voice Spacing | Avoid 2 nd Int. | Avoid 6 th Int. |
| Voice Leading | w/in 3 rd interval | w/in 5 th interval |

| | Arousal Min | Arousal Max |
|---------------|-------------------|-------------------|
| Roughness | 9 Joinings | 0 Joinings |
| Tempo | 60 BPM | 100 BPM |
| Articulations | Overlapping Notes | Half-Length Notes |
| Loudness | Velocity 50 | Velocity 70 |
| Voicing Size | 2 Notes | 8 Notes |

Table 1. Shows how musical features map to valence and arousal in this EMS system.

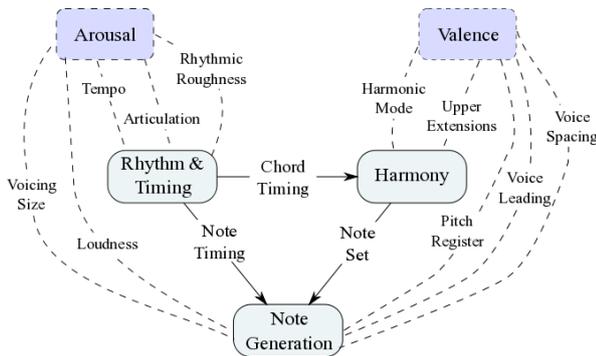


Figure 3. Flow of algorithm operation. Solid edges indicate control or information flow, dashed edges indicate an “implemented by” relationship.

3. EXPERIMENT

A user study was performed in order to determine how well the intended valence and arousal system parameters corresponded with actual listener evaluations of valence and arousal in the music. For this experiment, the scaled-down subset of musical parameters discussed in Section 2.4 was used.

3.1 Participants

Eleven participants—five female, six male—recruited from a student pool at ASU consisting of psychology students, whose ages ranged from eighteen to twenty-one years of age. As per the requirements of recruiting from this student pool, each participant was given class credit in return for participation.

3.2 Procedure

Each participant was stationed at a computer terminal consisting of monitor, keyboard and mouse, and speakers. Several practice trials were completed, and when the participant felt ready the experiment began. Each trial proceeded as follows: First, the EMS algorithm generated music at a specific valence and arousal setting—randomly selected from thirty-six valence/arousal configurations evenly dividing the parametric space into a six-by-six grid—for fif-

teen seconds. Once the music stopped, a clickable grid appeared on the graphical user interface and the participant clicked at the valence/arousal point which, in his or her opinion, most closely matched the music. After the click, the grid disappeared and another trial began. Four blocks of thirty-six trials were completed by each participant.

3.3 Results

Each trial’s X and Y mouse-click location was recorded along with the valence and arousal of the generated music. Correlations and t-tests were performed on this data as depicted in Table 2. Also, graphs were generated showing the mean clicked valence or arousal for every intended valence or arousal setting, and the standard deviations over these means (Figures 4 and 5). Data was not normalized before these operations.

| | Intended Valence | Intended Arousal |
|-------------------|--|--|
| Perceived Valence | $r = 0.475(p < .001)$ $t = 36.82(p < .001)$ | $r = 0.364(p < .001)$ $t = 18.09(p < .001)$ |
| Perceived Arousal | $r = 0.008(p < .001)$ $t = 0.44(p < .66)$ | $r = 0.679(p < .001)$ $t = 23.61(p < .001)$ |

Table 2. Shows correlations between perceived and intended arousal and valence.

4. DISCUSSION

This user study shows that our EMS algorithm is fairly well designed since changes to the settings of the valence or arousal parameters resulted in participants hearing corresponding changes in the emotion of the music. One caveat exists, which is the fact that there is some crossover between intended arousal and perceived valence (Figure 5b). Ideally, changes to the arousal parameter would result in no change at all in the perceived valence, but this was not the case. The fact that the crossover is not symmetrical—perceived valence correlates with intended arousal, but perceived arousal does not significantly correlate with intended valence—is interesting. This indicates that the crossover is not a result of mapping a musical parameter to the wrong emotional axis. Therefore, the crossover is probably a result of a non-orthogonal relationship between arousal and valence with regard to one or more of the features. It could also be the result of uncontrolled factors such as cultural effects. In any case, this crossover does not greatly affect the desired operation of the EMS system, because the within-dimensions results behave as expected.

The fact that this EMS system generates music algorithmically instead of transforming pre-composed music has benefits and drawbacks. One benefit is that the generated music is completely new to listeners, therefore previous exposure to the music could not bias user evaluations in this study. A potential drawback is that, although the music in this system is randomly-generated and non-repeating, it still has stylistic similarities with itself. Whereas other EMS algorithms are capable of more variety because they transform pre-existing pieces, this system sounds like a single composition. This leads to the possibility that this

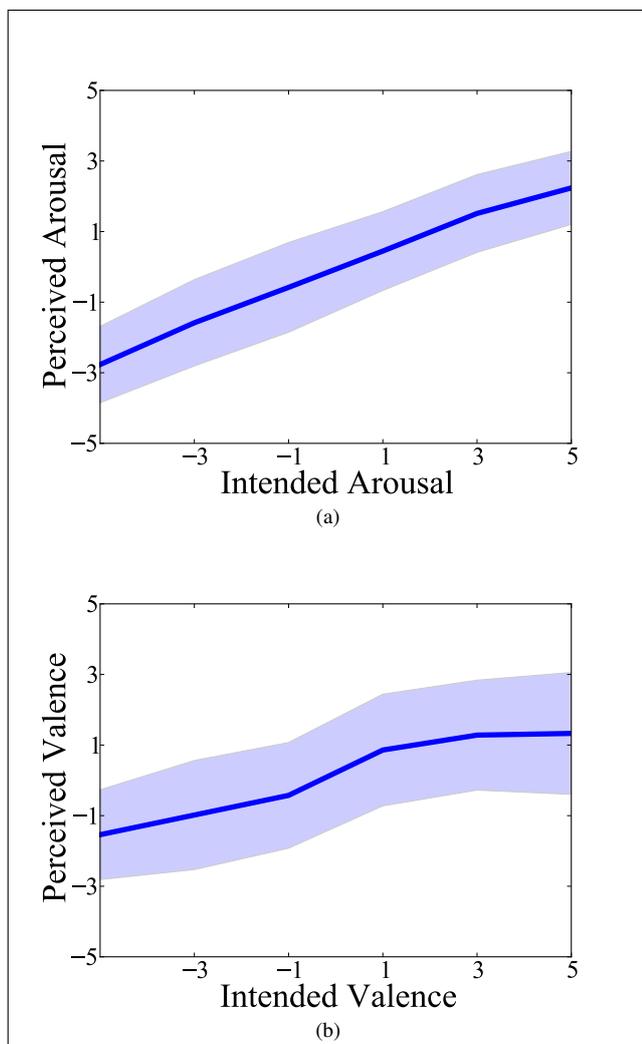


Figure 4. Within-dimensions results of study. These graphs show the mean clicked valence or arousal (and standard deviation above and below the mean), at each generated valence or arousal setting.

music could be emotionally biased; that this entire system, in comparison with other music in the world, may have an emotional value in the same way many people feel certain genres of music are inherently angry or sad (e.g. “hard rock sounds angry”).

4.1 Topics for Further Study

Although the musical parameters manipulated by this EMS system affected listener perceptions of musical emotion as expected, the complex relationships between parameters mean that separating the parameters or using them in different configurations will have unpredictable results. For this reason, this user study is presented only as an evaluation of this EMS algorithm; we make no claims about musical emotion in the overall sense. However, during the course of this project, certain insights emerged about EMS design. These need further study as they have not been empirically proven. However, they seem to be plausible hypotheses. These hypotheses are described in the following sub-sections.

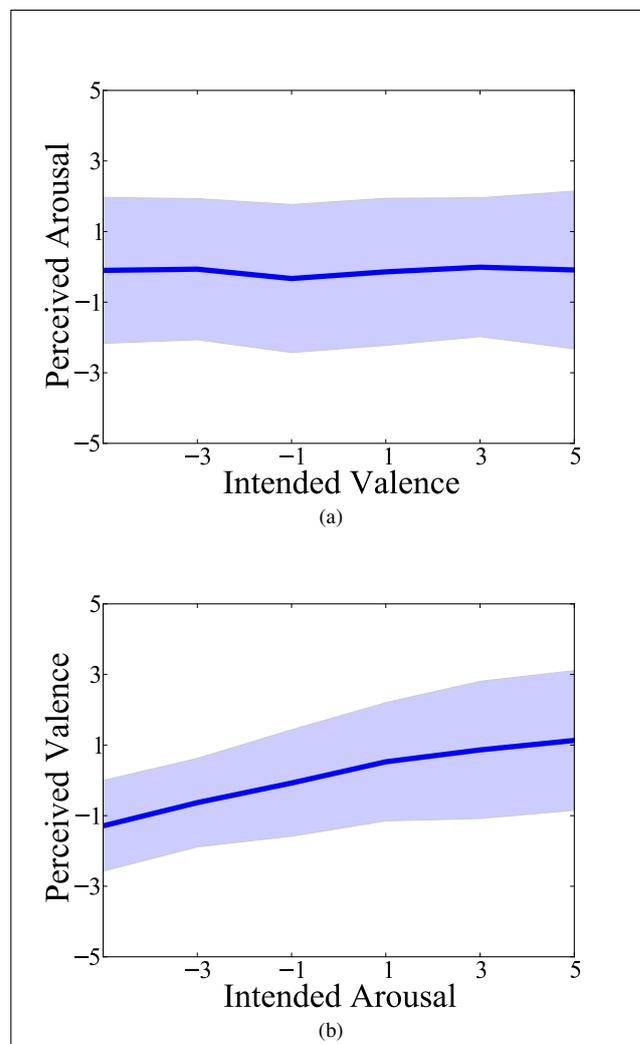


Figure 5. Between-dimensions results of study. These graphs show the mean clicked valence or arousal (and standard deviation above and below the mean) at each generated valence or arousal setting.

4.1.1 Timing and Volume Parameters Should be Mapped to Arousal, While Tonality and Timbre Parameters Should be Mapped to Valence

All system mappings between the valence/arousal space and the musical features were first based on mappings in Gabriellson and Lindström’s compilation of emotional music study results, and then were subject to much experimentation before a final design was finalized. In the end, all timing-based parameters (rhythmic roughness, tempo, articulation) and all loudness-based parameters (voicing size, loudness) were mapped to arousal. Similarly, all parameters related to tonality (mode, register, voice spacing) were mapped to valence. Although this division of parameters is simplistic, it seems to suffice for the purposes of generative EMS. The arousal link with timing is in keeping with existing theory on proprioceptive emotional contagion [16]. Also, some evidence suggests that valence and tonal parameters are cognitively linked with language and prosody [17], suggesting that many timbral features of music, which were not directly manipulated in this sys-

tem, would also map to valence. Studies of timbre with regard to emotion indicate that this is the case, although some aspects of timbre do correlate with arousal [18]. Exploring the salience of this mapping hypothesis—that timing and volume parameters should be mapped to arousal while tonal and timbral parameters should be mapped to valence—is an interesting and necessary direction for future study.

4.1.2 Density is Most Important Parameter For Arousal

Of all musical features, event density seems most important for determining arousal in EMS systems. In our system, event density was dictated by a combination of rhythmic roughness and tempo, but either would have been sufficient by itself to control arousal. Had we mapped only one of these features to arousal with no other features, it seems likely that listeners would have perceived the intended arousal. On the other hand, had we mapped all other arousal features (loudness, voicing size, articulation) to arousal without including either tempo or rhythmic roughness, listeners would not have strongly perceived the intended arousal.

4.1.3 Mode is Most Important Parameter For Valence

In EMS algorithms based on western music, mode will likely be the most important musical feature for determining valence. In Figure 4b, the steepest part of the graph is the center, which divides the Dorian mode from the Mixolydian mode. These modes are very similar, differing only in that the Dorian mode flattens its third scale degree. However, modes with a flattened third degree (such as Phrygian, Aeolian, and Dorian, which are on the left side of the graph) are considered to be minor modes, and modes without a flat third pitch such as Mixolydian, Ionian, and Lydian, which are on the right) are considered to be major modes. Had we mapped only the harmonic mode parameter to intended valence and no other musical parameters, most listeners would still have perceived the intended valence. However, had we mapped all the other parameters except harmonic mode (upper extensions, pitch register, etc.) the intended valence would not have been perceived as strongly.

4.1.4 Perceived Musical Emotion is Relative

It seems possible that emotion perception is relative, meaning that judgments on the emotion of music depend on comparing it to other music. This could be a comparison within a genre (e.g. “this music is angry compared to most jazz”), a comparison within a single composition (e.g. “this part is angry compared to the rest of the song”), or simply a comparison with the music one has been recently listening to. As stated earlier, the fact that this system has limited musical variation means that this study was vulnerable to emotional bias in the music. However, in Figures 4 and 5 it is obvious that the curves in all four graphs are centered on, and nearly symmetrical about, the mid-line. If the music were emotionally biased, and listener perception was not relative, some of these would be offset vertically instead of centered on the mid-line.



Figure 6. Interface to an emotional slideshow generator using EMS background music.

4.2 An Artistic Use of the EMS System

The EMS system in this paper was combined with Flickr [19] to create an emotional slideshow generator, the interface of which can be seen in Figure 6. The interface provides a drop-down list of emotions to select from, each of which has a valence and arousal value which has been determined by Russell [12]. When an emotion is selected, the EMS system begins to play at the correct valence and arousal while any Flickr images tagged with the emotion word are downloaded and displayed one after another. When the selected emotion has high arousal, the images will cycle through more rapidly.

Since image selection is dependent on tagging by Flickr users, there is nothing to ensure the emotional accuracy of the slides. With that said, the generated slideshow is, in the overall sense, remarkably apt. Sometimes unexpected connections are made which serve to highlight the use of social media; for example, setting the interface to serene might result in a slideshow consisting of mostly sleeping babies, sleeping pets, and landscapes—but once in a while, a bottle of pills may appear. In our view, these unexpected images are desirable, as this application was created for artistic/aesthetic purposes, not for any clinical use. To repurpose this application for scientific use, Flickr would need to be replaced with an image database whose images have been rigorously rated in terms of affect, such as the International Affective Picture System (IAPS) [20].

4.3 The Importance of Emotional Music Synthesis

The uses of compelling EMS are numerous and obvious; almost anywhere background music is currently used, EMS could be used. This makes it useful for the purposes of marketing and entertainment. Since EMS can be automated, it could be put to effective use in video games. Since EMS deals with emotions, it might be used in therapeutic systems for emotional imbalances and autism.

There exists a theory of emotional contagion [6, 16, 21] which is proprioceptive in nature, positing that certain structural features observed in outside agents such as music seem emotional because they correspond with movement

features. These movement features may be emotional for evolutionary reasons; at some point in evolutionary history, odds of survival increased if high-intensity emotions such as anger or terror correlated with rapid rates of movement. Therefore, if we observe some agent moving rapidly, even if it only moves rapidly in a conceptual sense as in the case of high-tempo music, it can seem as though that agent has a high-intensity emotion such as anger or terror. If this theory is accurate, then affecting a person's emotions will have repercussions in that person's movement. For this reason, EMS might be an especially useful form of interactive music for systems designed to rehabilitate Parkinson's Disease or stroke-induced impairments, such as the stroke rehabilitation system described in [22].

Most interactive media is designed to engage participants. In existing narrative multimedia, such as cinema, background music seems to greatly increase the chances that the audience will become immersed in the story. Therefore, one question of importance to interactive media is the following: How can we best leverage musical emotion in order to make interactive media appealing? Hopefully, in the future, scientists will be able to deploy EMS algorithms such as this one in order to better understand the answer to this question.

5. CONCLUSIONS

In this paper, we discussed the computerized generation of music which communicates with listeners on an emotional level. We presented a system which algorithmically generates emotional music; this system differs from many current EMS systems because it composes the music from scratch rather than transforming pre-existing music. Evaluation of this system shows that listeners perceive the emotions which the EMS system is designed to produce. This project resulted in the generation of some insights on EMS design and new hypotheses for future study. Lastly, this paper discussed the importance of EMS in the context of entertainment, therapy and rehabilitation, and digital media in the context of future uses.

Acknowledgments

This material is partially supported by the National Science Foundation CISE Infrastructure Grant No. 0403428 and IGERT Grant No. 0504647. We would like to thank Laura Gonzales for her helpful comments.

6. REFERENCES

- [1] P. N. Juslin and D. Västfjäll, "Emotional responses to music: The need to consider underlying mechanisms," *Behavioral and Brain Sciences*, vol. 31, no. 5, pp. 559–621, October 2008.
- [2] L. B. Meyer, *Emotion and meaning in music*. University of Chicago Press, 1956.
- [3] A. Gabrielsson and E. Lindström, "The influence of musical structure on emotional expression," in *Music and Emotion: Theory and Research, Series in Affective Science*, P. Juslin and J. Sloboda, Eds. Oxford University Press, 2001, ch. 10, pp. 223–248.
- [4] R. Bresin, "What is the color of that musical performance?" in *Proc. Int'l. Computer Music Conference (ICMC)*, Barcelona, Spain, 2005, pp. 367–370.
- [5] G. Collier, "Why does music only express some emotions? a test of philosophical theory," *Empirical Studies of the Arts*, vol. 20, no. 1, pp. 22–31, 2002.
- [6] M. Zentner, G. Didier, and K. Scherer, "Emotions evoked by the sound of music: Characterization, classification, and measurement," *Emotion*, vol. 8, no. 4, pp. 494–521, August 2008.
- [7] A. Friberg, "pdm: An expressive sequencer with real-time control of the kth music-performance rules," *Comput. Music J.*, vol. 30, pp. 37–48, March 2006.
- [8] R. Winter, "Interactive music: Compositional techniques for communicating different emotional qualities," Master's thesis, KTH Royal Institute of Technology, 2006.
- [9] S. R. Livingstone, R. Muhlberger, A. R. Brown, and W. F. Thompson, "Changing musical emotion: A computational rule system for modifying score and performance," *Comput. Music J.*, vol. 34, pp. 41–64, March 2010.
- [10] A. Oliveiro and A. Cardoso, "Modeling affective content of music: A knowledge-base approach," in *Proc. Sound and Music Computing Conf.*, Berlin, Germany, July 2008.
- [11] I. Wallis, T. Ingalls, and E. Campana, "Computer-generating emotional music: The design of an affective music algorithm," in *Proc. 11th Int'l. Conf. Digital Audio Effects (DAFx-08)*, Espoo, Finland, September 2008.
- [12] J. A. Russell, "A circumplex model of affect," *J. Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, December 1980.
- [13] J. A. Russell and A. Mehrabian, "Evidence for a three-factor theory of emotions," *J. Research in Personality*, vol. 11, no. 3, pp. 273 – 294, 1977.
- [14] V. Persichetti, *Twentieth-century harmony: creative aspects and practice*. New York: W. W. Norton, 1961.
- [15] R. Gundlach, "Factors determining the characterization of musical phrases," *American J. of Psychology*, vol. 47, no. 4, pp. 624–643, 1935.
- [16] J. Chen, V. Penhune, and R. Zatorre, "Listening to musical rhythms recruits motor regions of the brain," *Cerebral Cortex*, vol. 18, no. 12, pp. 2844– 2854, December 2008.

- [17] N. Cook, T. Fujisawa, and K. Takami, "Evaluation of the affective valence of speech using pitch substructure," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 142 – 151, January 2006.
- [18] T. Eerola, V. Alluri, and R. Ferrer, "Emotional connotations of isolated instruments sounds," in *Proc. 10th Int'l. Conf. Music Perception and Cognition (ICMP 10)*, Sapporo, Japan, 2008, pp. 483–489.
- [19] Flickr. [Online]. Available: <http://www.flickr.com/>
- [20] P. J. Lang, M. Bradley, and B. N. Cuthbert, "International affective picture system (iaps): Instruction manual and affective ratings," NIMH Center for the Study of Emotion and Attention, Tech. Rep., 2005.
- [21] P. Ekman, R. W. Levenson, and W. V. Friesen, "Autonomic nervous system activity distinguishes emotions," *Science*, vol. 221, pp. 1208–10, September 1983.
- [22] I. Wallis, T. Ingalls, T. Rikakis, L. Olsen, Y. Chen, W. Xu, and H. Sundaram, "Real-time sonification of movement for an immersive stroke rehabilitation environment," in *Proc. 13th Int'l. Conf. Auditory Display (ICAD)*, Montréal, Canada, June 2007.

AUTOMATIC MULTI-TRACK MIXING USING LINEAR DYNAMICAL SYSTEMS

Jeffrey Scott, Matthew Prockup, Erik M. Schmidt, Youngmoo E. Kim

Drexel University - Electrical and Computer Engineering

{jjscott mprockup, eschmidt, ykim}@drexel.edu

ABSTRACT

Over the past several decades music production has evolved from something that was only possible with multi-room, multi-million dollar studios into the province of the average person's living room. New tools for digital production have revolutionized the way we consume and interact with music on a daily basis. We propose a system based on a structured audio framework that can generate a basic mix-down of a set of multi-track audio files using parameters learned through supervised machine learning. Given the new surge of mobile content consumption, we extend this system to operate on a mobile device as an initial measure towards an integrated interactive mixing platform for multi-track music.

1. INTRODUCTION

The advent of digital audio and high-speed global communication has revolutionized the way people produce, distribute and consume music. It has become possible for individuals to make near professional recordings in their own home using a laptop and a microphone. While the technology has progressed enormously, a significant amount of skill and experience operating a digital audio workstation (DAW) is necessary to produce high quality results. Learning when and how to perform certain operations to transform a set of tracks into a polished product requires a large investment of time and training.

This paper targets the most basic studio production parameters, namely the mixing coefficients (fader levels) used to sum the tracks together to form a single audio output. Within the context of a standard rock/pop instrumentation (i.e. guitar, vocals, bass, and drums) the proportion of each track present in the mix is one of the most important factors determining the overall sound of a song. In this work we aim to predict time varying mixing coefficients for a set of multi-track *stems* that will produce a perceptually coherent and consistent final mix. Stem files are audio files that contain either a single instrument or a sub-mix of several instances of the same instrument or related instruments.

We describe a process for estimating the weighting coefficients when the source tracks and final mix are available

but the actual gain parameters used in the final mix-down are unknown. We use the estimated fader values to train a linear dynamical system (LDS) that estimates the weighting coefficients using a set of acoustic features extracted from the audio. The mix attained from applying the predicted weights to the source tracks accurately represents the true version of the song. Currently, the system requires prior knowledge of the type of instrument present on each track and is limited (by the training data) to bass, drums, guitar, vocals and backing accompaniment. Here, backing accompaniment can be vocal harmonies, additional guitar, percussion or keyboards.

1.1 Structured Audio Integration

We introduce an initial implementation of an integrated system for interactive music mixing on a mobile platform for *structured audio* using Apple's iOS for the Apple iPhone, iPod Touch and iPad. In the broadest sense, structured audio is a representation of sound content using symbolic or semantic information as a means of encoding the data. Using parameters estimated by the automatic mixing system outlined herein, we can generate a mix of individual source tracks in real-time on the mobile device.

The application also facilitates the use of multi-track sessions exported directly from a producer's DAW using the Advanced Authoring Format (AAF). Unlike proprietary DAW formats, AAF is an open standard, and is accessible by an available set of APIs that allow developers to easily access information about a DAW session. This includes the source audio for each track and session parameters such as time sampled mixing coefficients and panning [1]. The session is uploaded to a user's mobile device where the mix can be re-automated and modified in a custom structured audio player. The mobile platform can alter the gain values in real time using the actual fader values for unknown source material or the estimated values computed using the automatic mixing system. An overview of the AutoMix iOS implementation, complete from producer to playback, is shown in Figure 1.

The remainder of the paper is organized as follows, Section 2 elaborates on the previous research in this area and Section 3 details the dataset and how it affected decisions in the system implementation. In Section 4 we describe the model used to extract the weights from the dataset as well as the training and testing of an LDS to perform prediction. Section 5 discusses our experiments and results. Section 6 describes the mobile application and Section 7 summarizes our findings and provides insight into the next

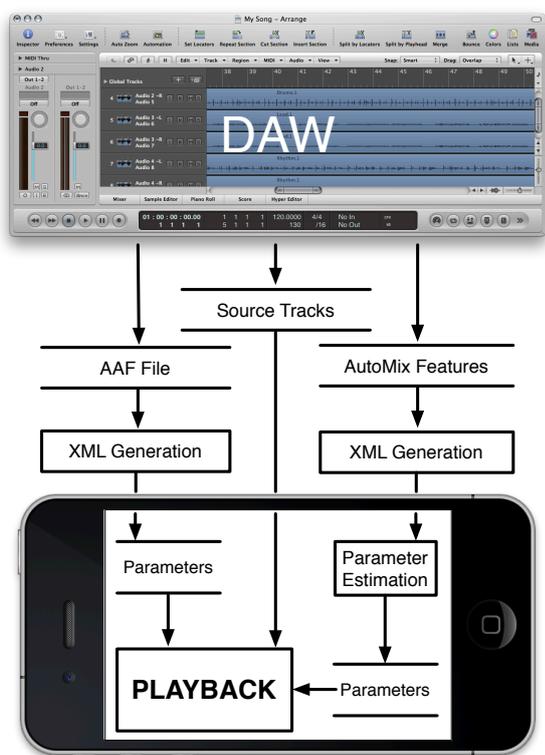


Figure 1. Diagram of iOS AutoMixing Implementation

avenue of research.

2. BACKGROUND

Much of the research in the area of automatic audio signal mixing is devoted to applications in the context of a live performance or event. Initial research on the subject was oriented toward broadcast, live panel discussion and similar environments dealing with the human voice as the primary audio source [2]. These systems analyze the amplitude of the audio signal and apply adaptive gating and thresholding to each input signal to create a coherent sound source mixture of the individual tracks in addition to feedback prevention.

More recent work incorporates perceptual features (loudness) into systems designed for live automatic gain control and cross-adaptive equalization [3, 4]. The implementation of the former focuses on adapting the fader level of each channel with the goal of achieving the same average loudness per channel. The latter is designed for use in live settings as a tool for inexperienced users or to reduce equipment setup time. The system attempts to dynamically filter various frequency bands in each channel so that all channels are heard equally well.

Structured audio is the representation of sound content with semantic information or algorithmic models [5]. This form of encoding allows for much higher data transmission rates as well as retrieval and manipulation of audio based on perceptual models. Currently, professional music post-production is performed by a highly skilled engineer with years of training. Using structured techniques, a parameterized, generative version of this process that is applicable

to a variety of source audio is possible.

Other related work seeks to equalize an audio input based on a set of descriptive perceptual terms such as *bright* or *warm* [6]. Rather than attempt to navigate the complex network of sliders and knobs in an audio interface, a user can specify a high level term that describes the desired sound quality and an appropriate equalization curve will be applied. The system was developed through collecting user ratings for audio examples and performing linear regression to find a weighting function for a particular instrument/timbre pair.

3. DATASET

The dataset used to learn the mixing parameters is a set of multi-track source files from the RockBand[®] video game. A total of 48 songs were selected randomly from various pop/rock artists with each song belonging to a unique artist. The ‘final mix’ experienced during gameplay was acquired by recording the optical audio output of the game console onto a computer and aligning it to the source tracks. The game console mix was used, as opposed to the radio/album release, due to synchronization issues between the source files and the radio version. It was evident that time stretching/compression was performed on many of the RockBand[®] releases since the song from the commercial release was often not the same length as the version from the game console.

There were several inconsistencies in the dataset which we had to account for in order to make comparisons between songs more accurate. The number and type of sources varied between each song, with a minimum track count of eight and maximum of 14. For example, many songs had individual stereo (L and R) waveforms for each instrument, whereas other songs only had mono tracks for some instruments and stereo tracks for others. Additionally, not all songs had individual tracks for the kick drum, snare drum or overhead drum microphones.

To deal with this discrepancy, we opted to form five mono tracks for each song: bass, drums, guitar, vocals and backup. The instruments in the backup track vary from song to song and may contain vocal harmonies, synthesizers, percussion, guitar or a variety of other instruments, however the content of the backup track within a song is fairly consistent. Given the content of the dataset, this method created more uniformity between the content of each song.

To create a single mono track for each instrument class, we mixed all audio that belonged to the given instrument class according to the track weights computed using the method described in Section 4.1. A diagram of the preprocessing step is shown in Figure 2.

4. AUTOMATIC MIXING

With the current dataset of RockBand stems and the mixed output file, we do not have access to the exact fader values used to create the final output mix, therefore we must estimate these parameters in order to train our model. The weight estimation process is subject to several unknowns

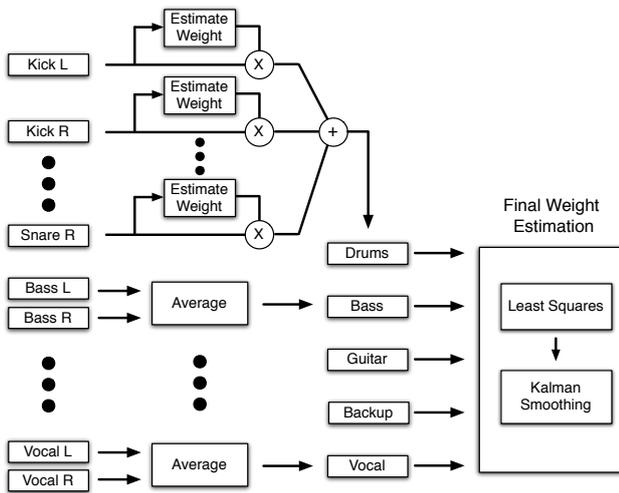


Figure 2. Diagram of dataset preprocessing for each track.

including additional compression and equalization of the stem tracks on the game console in producing the final mix. We use our estimated weights as ground truth for supervised machine learning and train an LDS to estimate a series of weighting coefficients for each track from a set of acoustic features extracted from the audio [7].

4.1 Weight Estimation

The process of mixing multi-track source files down to a single track is a linear combination of the audio sources in the time domain

$$\alpha_{1t}u_{1t} + \alpha_{2t}u_{2t} + \dots + \alpha_{kt}u_{kt} = v_t \quad (1)$$

where $\{\alpha_{1t}, \dots, \alpha_{kt}\}$ are the mixing coefficients of the k tracks at time t and $\{u_{1t}, \dots, u_{kt}\}$ are the time domain waveforms of each track.

Since the Fourier transform is a linear operator, we assume that the spectrum of the final mix at time t is a linear combination of the spectra of the source tracks at time t . Considering a single frame in time, we have

$$\begin{bmatrix} U_{11} & U_{12} & U_{13} & \dots & U_{1k} \\ U_{21} & U_{22} & U_{23} & \dots & U_{2k} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ U_{N1} & U_{N2} & U_{N3} & \dots & U_{Nk} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix} \approx \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix} \quad (2)$$

$\mathbf{U}\alpha \approx \mathbf{V}$

where each column in \mathbf{U} is the magnitude spectrum of the k th track and \mathbf{V} is the spectrum of the final mix. We are careful here to note that Equation 2 is not an exact equality in the context of our real data. Small offsets due to misalignment of the stems with the reference track will introduce error as a phase offset.

Given a set of multi-track stems and the resulting audio produced by mixing the individual tracks, we can estimate the mixing coefficients, α_k , using non-negative least

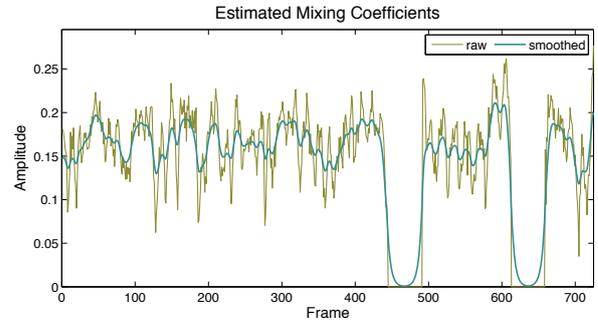


Figure 3. Extracted weights for bass guitar using NNLS, Kalman smoothing and normalization.

squares (NNLS) [8].

$$\hat{\alpha} = \min_{\alpha} \|\mathbf{U}\alpha - \mathbf{V}\|_2^2 \quad \alpha \geq 0 \quad (3)$$

We select NNLS to estimate the weights since the mixing process is additive by definition. Using unconstrained least squares, we experience both very large values for some weights since the algorithm can increase the weight of tracks that contain very little energy to reduce the overall error.

We perform this analysis on a frame-by-frame basis using a 1 second rectangular window and overlap the frames by 0.75 seconds. In each frame, we compute the spectrogram of each individual track using a 1024 sample window with a 512 sample overlap. We vectorize and concatenate the spectrograms to attain the form given in Equation 2 then compute the weights. A resolution of 0.25 seconds for changing fader values is sufficient to capture the dynamic changes in each track.

To improve the initial estimate of the weights, we only include tracks that contain audio in the given frame. Assuming we have k tracks, if $\text{RMS}(u_{kt}) < 0.01$, then we negate the track in the estimate of the weight vector for the current frame and use $k - p$ tracks, where p is the number of inactive tracks. Removing these tracks prevents very large weight coefficients from being calculated for tracks that have very little energy. The value of 0.01 was empirically determined to provide good peak suppression in the weight estimates.

We then process the weight vector using Kalman smoothing to reduce the noise that still remains in the signal [9]. The initial weight estimates as well as the smoothed weights are depicted in Figure 3. In the following section, we assume that the mixing coefficients are Gaussian when modeling the data. A histogram showing the distributions of mixing coefficients for multiple instruments is shown in Figure 4.

It is significant to note that while these coefficients produce a mix that is perceptually very similar to the original track, they are not the actual ground truth weights. We provide online audio examples of the original song and the mix using the estimated weights¹.

¹ <http://music.ece.drexel.edu/research/AutoMix>

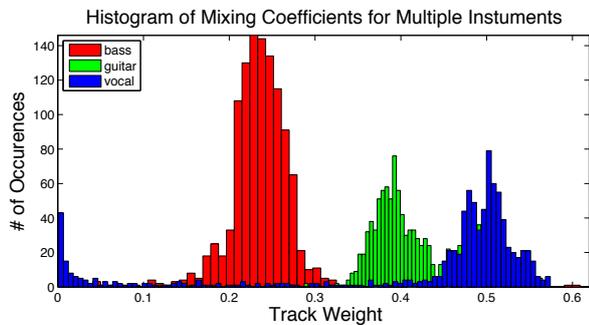


Figure 4. Histogram of mixing coefficients.

4.2 Feature Extraction

The features we extract to train the LDS are a combination of time domain and spectral domain values. The following features were used to train the model:

- Spectral Centroid
- Root Mean Square (RMS) Energy
- Slope/Intercept from fitting a line to the spectrum

4.3 Modeling

We train two different models using acoustic features to predict the time-varying mixing coefficients for an unknown input song. We first use multiple linear regression (MLR) to find the projection from features to weights that minimizes error in the least squares sense. To model time dependence between the mixing coefficients of a given track, we use a linear dynamical system (LDS) and compute the latent states using Kalman filtering.

4.4 Multiple Linear Regression

We assume that each weight vector α is a linear combination of our features $\{y_1, \dots, y_m\}$

$$\alpha = Y\beta \quad (4)$$

where Y is an $N \times M$ matrix, M is the number of features we have per frame, N is the number of frames and k indexes the track. We compute the projection matrix used to find the weighting coefficients of a new song,

$$\hat{\beta} = \min_{\beta} \|Y\beta - \alpha\|_2^2 \quad (5)$$

and estimate the mixing coefficients of an unknown song by applying the projection to the feature data Y .

$$\hat{\alpha} = Y\hat{\beta} \quad (6)$$

This model assumes that the mixing coefficients are independent with respect to time. In the next section we describe a model that considers the time dependence of the data.

4.5 Linear Dynamical System

We treat the time-varying mixing coefficients α as the latent states resulting from some noisy process and our features, y as noisy observations of the output of our model.

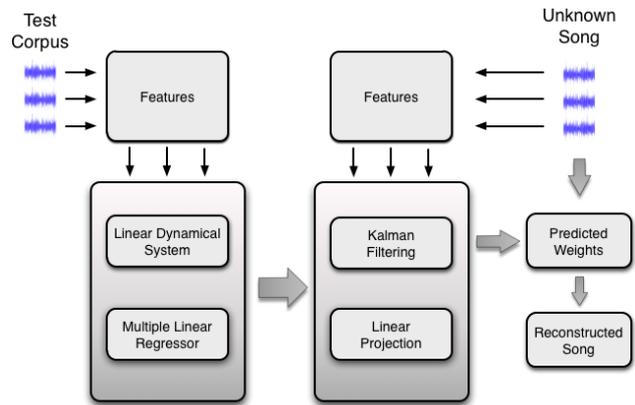


Figure 5. Supervised machine learning of gain coefficients using LDS and MLR.

We formulate the linear dynamical system as follows

$$\alpha_t = A\alpha_{t-1} + w_t, \quad (7)$$

$$y_t = C\alpha_t + v_t \quad (8)$$

Here, w_t and v_t are zero mean Gaussian noise sources

$$w \sim \mathcal{N}(0, Q) \quad (9)$$

$$v \sim \mathcal{N}(0, R) \quad (10)$$

The dynamics matrix A models the evolution of the weights as a linear transformation in each time step and C translates the α values into our observation space $y \in Y^R$.

To train the model we estimate A and C through constraint generation and least squares, respectively. We opt for a constraint generation approach to estimate C since a stable solution is guaranteed [10]. The covariances Q and R are computed from the residuals of A and C . Prior to training, we remove the means of the features and weights since our model assumes that the process is Gaussian and zero mean. The feature \bar{y} and weight $\bar{\alpha}$ means are retained for the testing phase.

For an unknown set of stems, we compute our acoustic features for each track and remove the training feature bias, \bar{y} . We then perform the forward Kalman recursions using the A , C , Q and R parameters learned during training to get an estimate of the weighting coefficients. Adding the weight bias $\bar{\alpha}$ to this result yields our final estimate of the mixing coefficients. A diagram of the feature extraction, training and estimation/prediction is shown in Figure 5.

5. RESULTS

Training and testing is performed in a typical manner for a supervised machine learning task. Given the relatively small size ($N = 48$) of the dataset we opt to use leave-one-out cross-validation, training on $N - 1$ songs and testing on the remaining song. This process is repeated for all N songs such that each is a test song only once.

We define Y_{train} as a matrix formed by concatenating the features of all songs, and α_{train} as the matrix formed by concatenating all weighting coefficients for all songs.

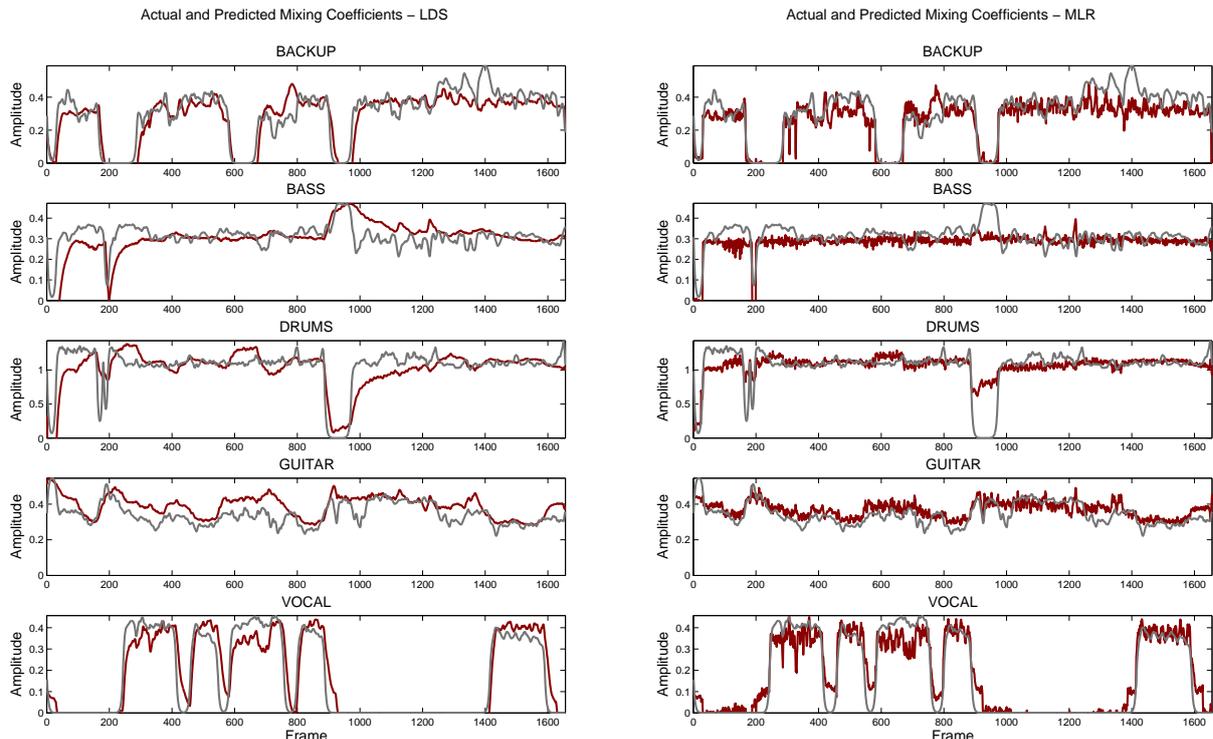


Figure 6. Results for weighting coefficient prediction using LDS (left) and MLR (right). The estimated ground truth weights are shown in gray and the predicted coefficients are depicted in red.

| Track | LDS | MLR |
|--------|---------------------|---------------------|
| backup | 0.0126 ± 0.0076 | 0.0091 ± 0.0075 |
| bass | 0.0191 ± 0.0183 | 0.0086 ± 0.0102 |
| drums | 0.1452 ± 0.1237 | 0.0590 ± 0.0444 |
| guitar | 0.0158 ± 0.0169 | 0.0075 ± 0.0077 |
| vocal | 0.0188 ± 0.0107 | 0.0149 ± 0.0124 |

Table 1. Average mean squared error across all songs between ground truth weights and predicted weights for MLR and LDS.

These quantities are then used to train the parameters of an LDS. We perform Kalman filtering on the remaining test song using the parameters learned in the training phase to estimate the time-varying weights for the song.

Figure 6 shows the predicted and actual weights plotted on the same axis for each instrument in the song “Constant Motion” by Dream Theatre. The resulting weights from MLR fit the data better and result in a lower error and the weights computed through Kalman filtering are much smoother yet sometimes exhibit bias or offset from the actual values. Table 1 shows the average mean squared error for all songs in the database for both algorithms.

Using a small dimensional feature set, we are able to generate a mix that is comparable to the desired result. Audio examples of the original mix, the drum sub-mixes and the reconstructed mix using the predicted weights can be found online at the previously specified link. A listening analysis performed by the authors finds that the LDS and MLR models yield very similar perceptual results. For

comparison, we generated audio mixes using a simple averaging of all tracks. The result of this oversimplified model is hardly comparable to the results from the automatic mixing system.

Although these results are good, we note that the weights estimated in Section 4.1 are not the true parameters. In order to have a cleaner data set we need a large collection of multi-track session files and access to the actual parameters. This idea is explored further in the next section as we introduce a mobile device application that can faithfully reproduce a mix using the actual parameters as well as the estimated model parameters from our system.

6. INTEGRATION INTO STRUCTURED AUDIO FRAMEWORK

The system detailed in Section 4 essentially describes a structured audio representation of music content. A song is represented by its component tracks and the output of a model, which, when applied to these component tracks produces a song. We incorporate the auto-mixing system into a larger framework for mixing and post-production of audio stem files. Within this framework we can potentially incorporate the actual parameters from DAW production sessions into a platform that allows the user to interact with the multi-track session in a variety of ways.

6.1 Automatic Mixing Implementation

The wide use of mobile technology for digital content consumption is the motivation to implement the automatic

mixing system for use on the iOS series of mobile devices. We created a set of tools that import audio features onto a mobile device then use these features to estimate the parameters using a hardware accelerated linear algebra library. The estimation is computed efficiently in real-time, mixing the output audio appropriately.

The source audio, feature data, and parameters of each model are computed offline, then uploaded to the mobile device in an XML wrapper to be parsed by the mixing platform. The LDS model includes the following data:

- features - \mathbf{Y}
- dynamics - \mathbf{A}
- translation - \mathbf{C}
- covariance - \mathbf{Q}
- covariance - \mathbf{R}

The MLR model is simpler, and only requires the features \mathbf{Y} and projection matrix β .

As of iOS 4, Apple added functionality of the Basic Linear Algebra Subprograms Library (BLAS) as part of their Accelerate Framework². This allows the device to perform hardware accelerated linear algebra calculations that are crucial to the MLR and LDS approaches to gain parameter estimation. The calculations are performed on a frame by frame basis and applied to the individual audio channels in real-time. The LDS approach is more computationally intensive in its operation, whereas MLR is a simple projection of the feature data. Both, however, run in real-time on the device.

6.2 Future Work: Collaboration with Producers

The current system works well for a specific subset (rock instrumentation) of source track content on the device due to the RockBand[®] dataset used for testing and training. While the current model may extrapolate to a variety of source content, the overall goal is to move beyond the RockBand[®] stems and collaborate directly with producers in the music industry in order to obtain data.

Most professional mixing is performed on a Digital Audio Workstation (DAW) such as AVID's Protools, Apple's Logic, MOTU's Digital Performer, and Steinberg's Cubase. All of these platforms, as well as many others allow the user to export an AAF file containing metadata of the parameters used to combine and process the individual source tracks. The metadata is the automation of parameters such as the gain and pan of a mixer channel. When the gain of a certain track is adjusted over time, the DAW records this alteration and reproduces it on playback. Using the source audio and the mixer automation information, a simple DAW session can be recreated outside of the platform it was created in. Collecting data in this format will facilitate integration and comparison between the metadata from divergent platforms.

We plan to use the data in AAF files to generate a more robust model of the parameter space in Section 4. We hope to increase the accuracy of the current predictions in regards to the standard rock instrumentation as well as enable reliable modeling for a larger class of instruments. Incorporating these ideas into the current framework will allow

novice users to interact in a novel way with available structured audio content with their own user generated content.

7. CONCLUSIONS

Leveraging a structured audio representation of music content, we developed a system that can estimate the original mixing coefficients of a multi-track recording session through least squares estimation. Using this result as ground truth, we trained a system to predict the time-varying parameters that produce a perceptually coherent mixture of unknown source content using minimal prior information. We deployed this system on a popular mobile device to show the creative potential for developing interactive music production applications that facilitate various modalities of personalization and customization for consuming music content.

8. REFERENCES

- [1] B. Gilmer, "AAF-the advanced authoring format," White paper, AAF Association, July 2002, <http://www.aafassociation.org>.
- [2] D. Dugan, "Automatic microphone mixing," *J. Audio Eng. Soc.*, vol. 23, no. 6, pp. 442–449, 1975.
- [3] E. Perez-Gonzalez and J. Reiss, "Automatic gain and fader control for live mixing," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009, pp. 1–4.
- [4] —, "Automatic equalization of multichannel audio using cross-adaptive methods," in *Audio Engineering Society Convention 127*, 10 2009.
- [5] B. Vercoe, W. Gardner, and E. Scheirer, "Structured audio: Creation, transmission, and rendering of parametric sound representations," in *Proceedings of the IEEE*, 1998, pp. 922–940.
- [6] A. T. Sabin and B. Pardo, "A method for rapid personalization of audio equalization parameters," *Proceedings of ACM Multimedia*, pp. 769–772, 2009.
- [7] E. M. Schmidt and Y. E. Kim, "Prediction of time-varying musical mood distributions using kalman filtering," in *Proceedings of the 2010 IEEE International Conference on Machine Learning and Applications*, 2010.
- [8] C. L. Lawson and R. J. Hanson, *Solving least squares problems*, 3rd ed., C. L. Lawson and R. J. Hanson, Eds. Prentice-Hall, 1995.
- [9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [10] S. Siddiqi, B. Boots, and G. Gordon, "A constraint generation approach to learning stable linear dynamical systems," in *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press, 2008, pp. 1329–1336.

² <http://developer.apple.com/performance/accelerateframework.html>

DANCERPRODUCER: AN AUTOMATIC MASHUP MUSIC VIDEO GENERATION SYSTEM BY REUSING DANCE VIDEO CLIPS ON THE WEB

Tomoyasu Nakano^{†1}

Sora Murofushi^{‡3}

Masataka Goto^{†2}

Shigeo Morishima^{‡3}

[†] National Institute of Advanced Industrial Science and Technology (AIST), Japan

[‡] Waseda University, Japan

¹ t.nakano[at]aist.go.jp

² m.goto[at]aist.go.jp

³ shigeo[at]waseda.jp

ABSTRACT

We propose a dance video authoring system, *DanceReProducer*, that can automatically generate a dance video clip appropriate to a given piece of music by segmenting and concatenating existing dance video clips. In this paper, we focus on the *reuse* of ever-increasing user-generated dance video clips on a video sharing web service. In a video clip consisting of music (audio signals) and image sequences (video frames), the image sequences are often synchronized with or related to the music. Such relationships are diverse in different video clips, but were not dealt with by previous methods for automatic music video generation. Our system employs machine learning and beat tracking techniques to model these relationships. To generate new music video clips, short image sequences that have been previously extracted from other music clips are stretched and concatenated so that the emerging image sequence matches the rhythmic structure of the target song. Besides automatically generating music videos, *DanceReProducer* offers a user interface in which a user can interactively change image sequences just by choosing different candidates. This way people with little knowledge or experience in MAD movie generation can interactively create personalized video clips.

1. INTRODUCTION

User-generated video clips called *MAD movies*¹ or *mashup videos*, each of which is a derivative (mixture or combination) of some original video clips, are gaining popularity on the web and a lot of them have been uploaded and are available on video sharing web services. In this paper, we focus on music video clips of dance scenes (dance video clips) in the form of MAD movies or mashup videos. Such a MAD music video clip consists of a musical piece (audio signals) and image sequences (video frames) taken from other original video clips. The original video clips are called *1st generation (primary or original) content*, and

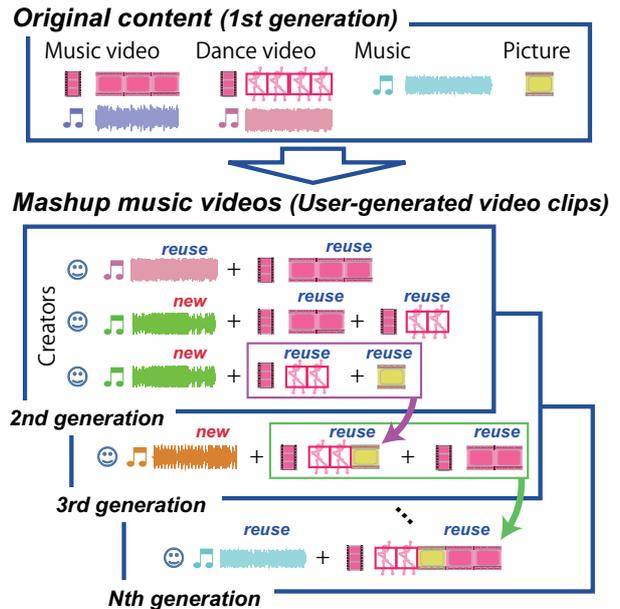


Figure 1. Generation of mashup music videos (user-generated music video clips) by reusing existing original content.

the MAD video clips generated by users can be considered *2nd generation (secondary or derivative) content* (Figure 1). In a MAD video clip, good music-to-image synchronization with respect to rhythm, impression, and context is important.

Although it is easy to enjoy watching MAD movies, it is not easy to generate them because a creator needs to (1) search, in existing video clips, for image sequences that give impressions appropriate to a given target musical piece, (2) segment and concatenate image sequences to fit the target piece, and (3) time-stretch the sequences to match the tempo of the target piece because existing video clips usually have tempi different from the tempo of the target piece. Moreover, for better music-to-image synchronization, the music structure and context of a musical piece and image sequences should be taken into account, but it requires considerable time and effort.

To give a chance of enjoying such difficult MAD movie generation to everybody, we have developed a new system called *DanceReProducer* that can automatically gen-

Copyright: ©2011 Tomoyasu Nakano et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

¹ http://en.wikipedia.org/wiki/MAD_Movie

erate a dance video clip for any given piece of music by segmenting, concatenating, and stretching existing dance video clips (Figure 2). This system provides an interface in which a user not only listens to music but also enjoys music visually by directing (supervising) the (semi)automatic generation of dance video image sequences. If the automatically generated video clip is satisfactory, the user can just watch it, but if the user does not like generated image sequences for some musical sections (e.g., A, B, and C in Figure 2), the user can easily choose another favorite image sequence from ranked candidates for each musical section. These candidates are also automatically proposed by the system and would also match a given musical section of the input piece according to our mapping model. This mapping model was trained through an analysis of a large amount of user-generated dance video clips available on a video sharing web service. In particular, we focus on the reuse of video clips of the 2nd, 3rd, and N th generation content (Figure 1) as well as the 1st generation content. In other words, our system enables a user to generate a new mashup video clip by reusing existing mashup video clips on the web.

2. RELATED WORK

Previous works generated visual patterns based on some musical aspects, such as visualizing music chords by color [1], visualizing musical mood [2], and controlling a computer-graphics dancer under musical beats [3, 4]. There were also previous works automatically generating music-synchronized video by reusing media content: for example, some reused images and photographs from the web [5, 6], and others reused home videos [7, 8] under audio changes [7] or repetitive visual and aural patterns [8]. Previous works, however, did not reuse dance video clips on the web to generate a new mashup video clip.

3. SYSTEM DESIGN

To develop DanceReProducer, we first considered the criteria that people use in judging “what is an appropriate image sequence for a particular piece of music”, as described below. We then describe functions of the system interface.

3.1 Criteria of natural/skillful relationships between an image sequence and music

To design the system, we considered the criteria from two aspects – local relationships and context (global) relationships explained below – taking into account previous work [7, 8] and the comments offered by human creators of MAD movies².

Local relationships : criteria for impression synchronization between the music and image sequences.

- *Rhythm*: Visual rhythms such as dance motion, camera work, and cut (e.g., dissolve) are synchronized with beat and musical accent.

² Some creators disclosed their creative processes on the web.

Automatic mashup music video generation system

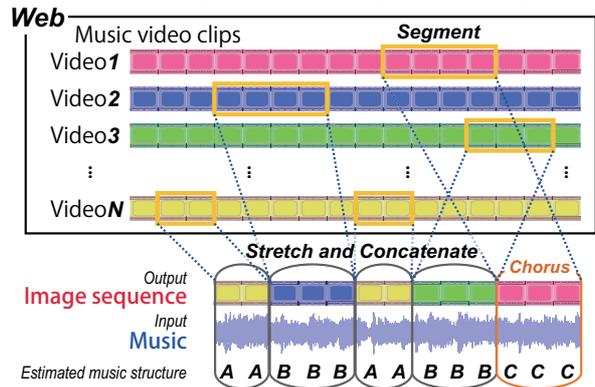


Figure 2. An automatic music video generation system DanceReProducer by reusing existing music video clips.

- *Impression*: Visual impressions such as dance motion, color, brightness, and lighting are synchronized with the musical impression.

Context relationships : criteria for context synchronization between music and image sequences.

- *Music structure*: Visual impression (temporal) changes are synchronized with the music structure. (e.g., verse A, chorus).
- *Temporal continuity*: Image sequence has temporal continuity, but visual impression can be changed easily on a music structure boundary.

The above criteria are not all satisfied at any given time, and are not mutually independent. However, they provide a useful foundation for generating an image sequence appropriate to a particular piece of music.

3.2 Image sequence generation

The mashup video generation done manually is difficult and time-consuming. To enable more efficient generation, our system first automatically generates an image sequence appropriate to the music. However, the generated sequence may not be to the user’s taste. In such cases, other sequence candidates are shown on a screen so that the user can simply choose a preferred one. Even though it would be difficult for a user to manually find another candidate from among a huge number of candidates, it is easy to interactively choose a preferred candidate.

We provide an overview of the interface’s image sequence generation and functions below.

3.2.1 Automatic image sequence generation

To reuse existing content, we first gather dance video clips on a video sharing web service and the system estimates the tempo and bar line of the music (audio signals) in those video clips. We assume the music and its dance motions within each video clip are synchronized while dealing with the local relationships, and use each bar (measure) of the



Figure 3. Example of the DanceReProducer screen.

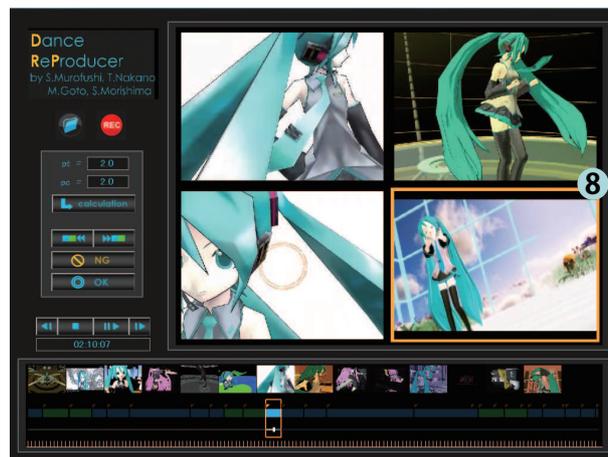


Figure 4. Example of interactive sequence selection. Four different image sequence candidates are previewed and the lower-right candidate is chosen by a user.

music as the minimum unit for segmenting and concatenating image sequences. Hereafter, we denote an image sequence (series of video frames) for the bar-level minimum unit as a *visual unit*.

Second, the system searches for a visual unit appropriate to each bar for the input target musical piece. The units are time-stretched under the tempo of the input music, and then are concatenated to generate an image sequence. In this regard, to deal with the context relationships, the system selects visual units which take into account music structure and temporal continuity.

To satisfy each criterion described in 3.1, we implement the following processes.

Rhythmic synchronization: A musical bar is used as the minimum unit for segmenting and concatenating. A visual unit is stretched under input music tempo.

Impression synchronization: By modeling the mapping between the extracted audio and visual features for impression, the system automatically selects an appropriate visual unit to input music impression in each bar.

Music structure and Temporal continuity: By introducing costs representing the temporal continuity and music structure of the generated sequence, the system automatically selects an image sequence considering the context relationships.

3.2.2 Interface

Screenshots of the implemented DanceReProducer interface are shown in Figure 3 and 4. There are basic functions for viewing, such as a window showing the generated image sequence (Figure 3, ①), functions to load input music and save the generated video (②), to play and stop/pause the generated video (③), and a playback-position “slider” and the music structure estimated automatically [9] (④). The green rectangular markers in the music structure represent chorus sections, and the blue markers represent other sections. In addition, the total duration of the input music is equally divided into 15 sections (⑤).

This interface also provides the following functions to reflect the user’s preferences.

Interactive re-selection of a generated image sequence:

By clicking the NG button (Figure 3, ⑥), the user can see other sequence candidates on a screen and simply choose the preferred one (Figure 4, ⑧). The user can see and compare different candidates during playback and can choose his/her favorite sequence. Since this interactive re-selection function works on each section of the music structure (e.g., A, B, and C in Figure 2), the user can use this function to easily consider the music structure and context.

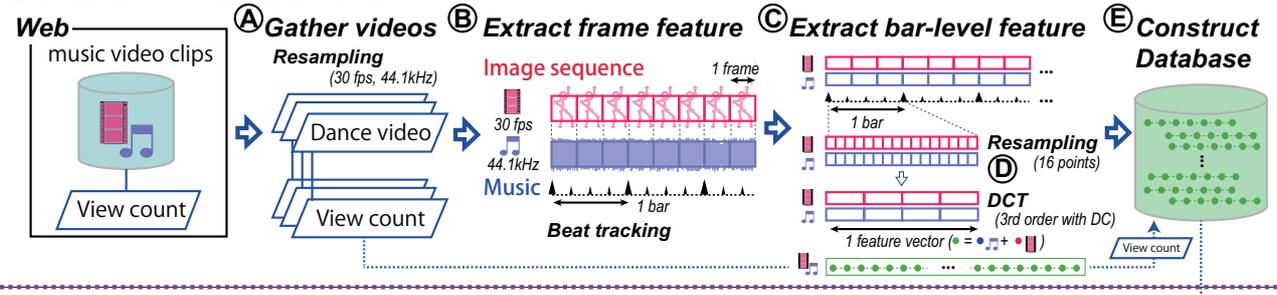
Jumping to the beginning of sections: By clicking the jump button (Figure 3, ⑦) or visualized sections (④), a user can directly jump to and view the previous or the next section of a song.

4. INTERNAL MECHANISM OF DANCEREPRODUCER

To develop DanceReProducer, we modeled the relationships between music and video, and then generated image sequences appropriate to input music by considering the local and context relationships. In general, it is difficult to model such relationships, but we solved this problem through training using a huge quantity mashup video clips posted to the web. Since the content videos were made by humans, there were various types of mutual relationship between the music and the image sequences. This suggests that such videos can be used to learn the relationships through a machine-learning technique.

Modeling using the mashup clips suffers from two problems. One is that complex relationships exist, such as where “the same image sequences are used for different music” or “different image sequences are used for the same music” (Figure 1). Another problem is that the video quality varies strongly, and it is difficult to judge the possibility

Database construction



Video generation

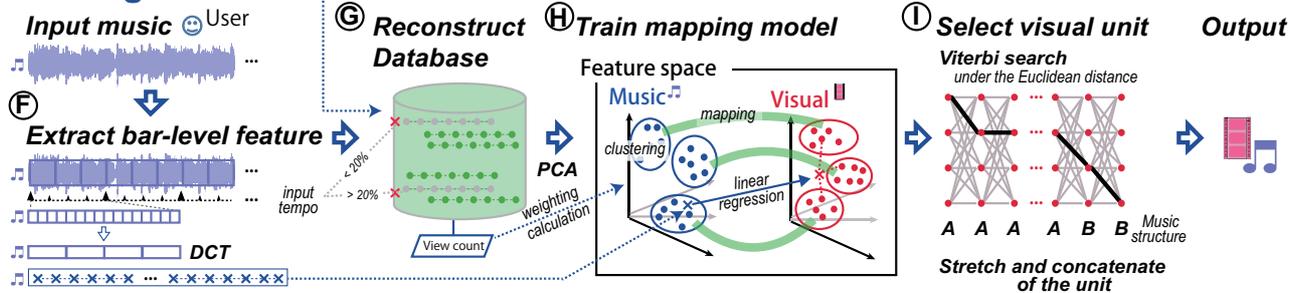


Figure 5. Overview of DanceReProducer, a dance video authoring system that can automatically generate a dance video clip appropriate to a given piece of music by segmenting, concatenating, and stretching existing dance video clips.

of its reuse. These obstacles make it difficult to model the relationships, and were not dealt with previous works.

Figure 5 gives an overview of the DanceReProducer system. The system consists of two procedures: database construction and video generation. In this section, we describe the details of the system and explain how we solve the above two problems in modeling using the mashup clips.

4.1 Database construction

In the database construction, database videos are gathered via the web and then audio and visual features are extracted from the videos through the following steps.

- Step 1) Gather dance music videos via web, and resample the sampling frequency of the music to 44.1 kHz, and the frame-rate of the image sequence to 30 fps (Figure 5, (A)).
- Step 2) Estimate bar line of the videos by using beat tracking techniques (B).
- Step 3) Extract feature vectors to learn their relationship (B–C). Since the analysis frame matches the frame rate, the discrete time step (1 *frame-time*) is about 33 ms (about 1470 points). The extracted features in each frame-time are called *frame features*. The frame features are then integrated in each bar to obtain what are called *bar-level features*.

4.1.1 Beat tracking

Much work has been done on beat tracking [3], and we plan to focus on using such techniques in the future, but our current implementation is a simple one which was effective in our preliminary experiment.

The system first calculates the power of the input audio signal, and then calculates its autocorrelation values and estimates their peak time. Since it represents the periodicity of the power, we use the time as tempo (one beat time). In this regard, to avoid octave error (e.g., half-/double-tempo error), the estimation is limited to tempo within a range of 60 – 120 bpm (beat per minute).

Second, the system calculates cross-correlation between the power and the pulse signal generated under the estimated tempo. Since the peak time of the cross-correlation represents the first beat time, the system regards the time as the beginning time of the first bar. In addition, we assume that the dataset videos have a length of 4 beats (one measure in 4/4 time), and then the system decides all bar lines mechanically.

4.1.2 Frame feature extraction (Music)

The frame features of music are defined with the help of previous work on relationships between audio and visual [10, 11] and musical genre classification [12]. These features represent musical accents and impressions.

As the frame features for accents, to represent temporal change in the power of the audio signal, we extract the filter bank output (4 dims.) and spectral flux (1 dim.). As the frame features for impressions, to represent timbre, we extract the zero-crossing rate (1 dim.) and 12th order MFCCs (mel-frequency cepstral coefficients) with a DC component (13 dims.).

4.1.3 Frame feature extraction (Image sequence)

The frame features of an image sequence are defined with the help of previous work on relationships between audio and visual [10, 11]. These features represent visual accents

and impressions. To extract the features, the image resolution is resampled to 128×96 .

As the frame features for accents, to represent camera work and dance motion and related temporal changes, we extract the mean values of the temporal derivative of the well-known optical flow and brightness (2 dims.). We use a block-matching algorithm to detect the optical flow from image sequences; we use a 64×48 block which is shifted by 1 (maximum range is 4). The frame features for impressions are the mean values and standard deviations of the hue, saturation, and brightness values (6 dims.). In addition, 2-dimensional DCT (discrete cosine transform) coefficients are extracted (4 dims. for vertical and 3 dims. for horizontal).

4.1.4 Bar-level feature extraction

We propose a *bar-level feature* which is an integration of the frame features in each bar. To extract features from one piece of music or one video clip, in most previous work (e.g., musical genre classification) integration was done using the time average and its standard deviation [12]. However, such integration drops temporal information of the audio/visual features.

In this paper, we integrate these frame features to bar-level features via using DCT (Figure 5, ①). In each bar, frame features are resampled to 16 points for the time axis, the system computes DCT for each dimension, and then the 3rd order DCT coefficients with a DC component used as the bar-level features. Therefore, the number of dimensions of the bar-level features is four times the number from the frame features.

4.2 Video generation

In the video generation, to select visual units for each frame from the database, the system process consists of the following steps.

- Step 1) Extract the bar-level features of a given musical piece (Figure 5, ②).
- Step 2) Reconstruct the database (③). To avoid generating a video with unnaturally fast/slow tempo, visual units with tempi 20% above or below the input tempo are not used for the following steps.
- Step 3) Apply PCA (principal component analysis) for all bar-level features of all bars, and store low N -dimensional features. The N -dimension is decided based on the cumulative contribution ratio ($\leq 95\%$). For our investigations, the dimensions of audio and visual features described above were reduced from 76 to 62 and from 80 to 68, respectively³.
- Step 4) Model relationship between music and image sequence from the database (④). This step is explained in more detail below (section 4.2.1).
- Step 5) Select visual units under the criteria of the relationships described in 3.1 (⑤).

³ Since the database is reconstructed depending on the tempo of the input, the reduced dimension is not constant.

4.2.1 Linear regression models for multiple clusters

In this paper, a local cost is calculated by a linear regression model, which is used to learn the relationships between the audio and visual bar-level features. However, to model complex relationships, such as “the same visual units are used for different music” or “different visual units are used for the same music” (Figure 1), one regression model is insufficient.

Therefore, we propose a linear regression, where the system uses linear regression models for multiple clusters. The multiple clusters are obtained by applying k -means clustering to feature vectors, where a feature vector is defined as a concatenation of a bar-level audio feature (of music) and a bar-level visual feature (of image sequences) in the database. Note that this feature vector is used just for the clustering. For each cluster, a linear regression model is trained so that bar-level visual features can be predicted by bar-level audio (music) features (Figure 5, ⑥).

4.2.2 Image sequence selection under the criteria for natural/skillful relationships

By introducing costs representing the local and context relationships, we can solve this video generation problem by minimizing the costs through a Viterbi search (Figure 5, ⑦). The model of the cluster having the centroid nearest to the input features is selected, and visual features appropriate to the input audio features are estimated by using the model. To calculate the costs of the local relationships, the system calculates the distance between the estimated features and the visual features of all units.

To represent the costs of the context relationships, a musical structure and chorus section are estimated using RefraiD [9]. The estimated beginning and ending times of all sections are used as the boundaries of a musical section. However, sections less than 4 bars in length are not used as a section for this purpose.

Let $d(n, k_m)$ be the Euclidean distance representing the local cost between the n ($1 \leq n \leq N$)th bar level feature of the input and the m th video’s k th unit’s features of the database. The calculated local costs and accumulated costs are defined as follows.

$$c_l(n, k_m) = \begin{cases} d(n, k_m) & \text{if } ch(n) = 1 \\ & \wedge ch(k_m) = 1, \\ p_c \times d(n, k_m) & \text{otherwise} \end{cases} \quad (1)$$

$$c_a(n, k_m) = \min_{\tau, \mu} \begin{cases} c_l(n, k_m) & \text{if } (\mu = m \wedge \kappa = k - 1) \\ +c_a(n - 1, \kappa_\mu) & \vee st(n) \neq st(n - 1) \\ p_t \times c_l(n, k_m) & \\ +c_a(n - 1, \kappa_\mu) & \text{otherwise} \end{cases} \quad (2)$$

where $ch(n)$ returns 1 if n is included in a chorus section, and $st(n)$ returns the number of musical sections. A higher p_c value means that the unit of chorus sections are more easily selected at a chorus section. A lower p_t value means that the selected unit has less time continuity. To minimize the accumulated cost, at the N measure, the system

selects a unit which has minimum accumulated cost d_{min} , and then a image sequence is generated by back-tracing.

$$d_{min} = \operatorname{argmin}_{k,m} c_a(N, k_m). \quad (3)$$

The interactive re-selection function is implemented so that the system chooses four different candidates for each section (Figure 4). These candidates are made from four different accumulated costs, and then four image sequences are generated by back-tracing. To expand the variety of generated image sequences, the chosen candidates are made from minimum, 1/3 minimum, 2/3 minimum, and maximum accumulated costs. This enables generation of a variational image sequence.

4.3 Model training weighted according to view counts

This paper focuses on the reuse of the MAD movies available on the web. Since there are many creators, the authoring quality of generated videos varies widely. In other words, each video will have a different level of reliability regarding the relationships between music and image. We assume that a video generated by a user having good MAD movie skills will have higher reliability and higher possibility of its reuse. Therefore, to model an appropriate image sequence to particular music, the system should introduce a weighting factor in the model training process where higher quality video will be given a greater weight.

To enable automatic judgment of the quality, we introduce the idea of using the view count of each video clip on the web as a weight since the view count reflects the video quality. Let ω be an integer weighting factor defined as follows, where V_c indicates the view count:

$$\omega = \max(\alpha \times \lfloor \log_{10}(V_c) + 0.5 \rfloor + \beta, 0). \quad (4)$$

In our current implementation, α and β are set to 2 and -7 , respectively. This means, a view count of 10,000 corresponds to $\omega = 1$, while a view count of 100,000 corresponds to $\omega = 3$. To implement the weighted training, the number of bar-level audio/visual features (training samples) of a video clip is virtually increased by its ω (doubled by $\omega = 2$, for example) in training linear regression models.

5. IMPLEMENTATION OF DANCEREPRODUCER

In this section, we describe the dataset used and trial user comments regarding the system effectiveness.

5.1 Dataset

To generate a dance video by segmenting and concatenating from existing dance video, and to model the various relationships between music and an image sequence, the database should fulfill the following four conditions.

- Condition 1) The main content of video clips is dance.
- Condition 2) Video clips are similar types of MAD movies so that their mixture generated by our system can look like a consistent content.

Condition 3) Each video clip has the view count by users on the web.

Condition 4) The number of available video clips is large enough.

As content fulfilling all of the above conditions, we used mashup videos which are generated from Japanese dance simulation games full of dance scenes, “THE IDOLM@STER” and “THE IDOLM@STER LIVE FOR YOU!”⁴. In addition, we also used dance videos which are generated using *MikuMikuDance (MMD)*⁵ that is a 3-dimensional human motion synthesizer for dance performance. Both videos can be found on a video sharing service *NicoNicoDouga*⁶. To construct a database, we gathered 100 of these mashup video clips and 100 of these MMD video clips, all of which had the view count of over 10,000 on the NicoNicoDouga.

5.2 Trial usage and introspective comments

Many videos generated by DanceReProducer were synchronized regarding rhythm and impression between the music and image sequence. This suggests that the system can be effective and the modeling is appropriate.

Trial users of the system offered comments, especially regarding the effectiveness of the interactive re-selection function. A typical comment was that “the function was useful and effective”; however, in contrast, another user commented that “occasionally there was no appropriate candidate”.

Some comments were on ways to improve the system performance. One user, who had no experience in MAD movie generation, said it would be useful to have “more candidates for the image sequence”. Another comment, from a user who had MAD movie experience, was that the system needed an “adjustment function for the bar and boundary of the musical section”.

6. CONCLUSION

DanceReProducer is a dance video authoring system that can automatically generate dance video appropriate to music by reusing existing dance video sequences. Trial usage of the system has shown that it is a useful tool for users with little knowledge or experience in MAD movie generation⁷. Although dance video content is currently supported in our implementation, our approach has capability to utilize for any other music video clips.

One benefit of DanceReProducer is that a user does not need to engage in time-consuming manual generation. Moreover, the “reuse” approach described in this paper is novel in that it allows the use of ever-increasing user-generated content on the web. We expect the expansion of mashup content (n th generation content), and its supporting systems, to create an opportunity for a new form of entertainment. Remaining issues, such as a quantitative

⁴ <http://www.bandainamcogames.co.jp/cs/list/idolmaster/>

⁵ http://www.geocities.jp/higuchuu4/index_e.htm

⁶ <http://www.nicovideo.jp/>

⁷ Demonstration video clips generated by our system are available at <http://staff.aist.go.jp/t.nakano/DanceReProducer/>

evaluation of this system, feature extraction for dance motion in detail (like the body motion detection⁸), and an interface that can adjust measure or section boundaries, will be topics covered in our future work.

Acknowledgments

We thank Yuki Hasegawa and Tatsunori Hirai for their help.

7. REFERENCES

- [1] T. X. Fujisawa, M. Tani, N. Nagata, and H. Katayose, "Music mood visualization based on quantitative model of chord perception," in *Journal of Information Processing Society of Japan*, vol. 50, no. 3, 2009, pp. 1133–1138. (in Japanese)
- [2] C. Laurier and P. Herrera, "Mood Cloud : A realtime music mood visualization tool," in *Proc. of the 2008 Computers in Music Modeling and Retrieval Conference*, 2008, pp. 163–167.
- [3] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," in *Journal of New Music Research*, vol. 30, no. 2, 2001, pp. 159–171.
- [4] T. Shiratori and K. Ikeuchi, "Synthesis of dance performance based on analyses of human motion and music," in *IPSJ Transactions on Computer Vision and Image Media*, vol. 1, no. 1, 2008, pp. 34–47.
- [5] X.-S. Hua, L. Lu, and H.-J. Zhang, "Automatically Converting Photographic Series into Video," in *Proc. of the 12th annual ACM international conference on Multimedia*, 2004, pp. 708–715.
- [6] R. Cai, L. Zhang, F. Jing, W. Lai, and W.-Y. Ma, "Automated Music Video Generation using WEB Image Resource," in *Proc. of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2007)*, 2007, pp. II-737–II740.
- [7] J. Foote, M. Cooperand, and A. Girgensohn, "Creating music videos using automatic media analysis," in *Proc. of the tenth ACM international conference on Multimedia*, 2002, pp. 553–560.
- [8] X.-S. Hua, L. Lu, and H.-J. Zhang, "Automatic music video generation based on temporal pattern analysis," in *Proc. of the 12th annual ACM international conference on Multimedia*, 2004, pp. 472–475.
- [9] M. Goto, "A chorus-section detection method for musical audio signals and its application to a music," in *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 5, 2006, pp. 1784–1794.
- [10] O. Gillet, S. Essid, and G. Richard, "On the correlation of audio and visual segmentations of music videos," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 2, 2007, pp. 347–355.
- [11] M. Nishiyama, T. Kitahara, K. Komatani, T. Ogata, and H. G. Okuno, "A Computational Model of Congruency between Music and Video in Multimedia Content," in *IPSJ SIG Technical Reports 2007-MUS-069*, vol. 2007, no. 15, 2007, pp. 111–118. (in Japanese)
- [12] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," in *IEEE Trans. on Speech and Audio Processing*, vol. 17, no. 2, 2002, pp. 293–302.

⁸ EyesWeb: <http://www.infomus.org/EywMain.html>

ON THE CREATIVE USE OF SCORE FOLLOWING AND ITS IMPACT ON RESEARCH

Arshia Cont

IRCAM CNRS STMS

Musical Representations Team

arshia.cont@ircam.fr

ABSTRACT

Score following research is one of the active disciplines of sound and music computing since almost 30 years that have haunted both algorithmic and computational development in realtime music information retrieval, as well as artistic applications in interactive computer music. This paper explores the creative use of such technologies and brings attention to new scientific paradigms that emerge out of their artistic use. We show how scientific and artistic goals of score following systems might differ and how the second, continuously helps re-think the first. We focus mostly on the musical goals of score following technologies which brings us to an underestimated field of research, despite its obviousness in creative applications, which is that of *synchronous reactive programming* and its realization in *Antescofo*.

1. INTRODUCTION

Score following is traditionally the automatic and realtime alignment of audio streams from musician(s) on the stage into a symbolic music score. In its artistic use, it allows realtime coordination and synchronization of live electronic programs with human performers for mixed interactive computer music pieces. In the scientific literature, it is also employed in off-line mode for alignment of audio to symbolic music scores as a front-end for music information retrieval applications.

The score following literature is one of the research disciplines in sound and music computing with clear impacts on both research literature and artistic applications in computer music. The number of published articles on score following algorithms are constantly increasing every year, and since a few years, more composers of interactive computer music are employing such technologies into their compositions. Since the inception of score following paradigms in the 1980s, the two fronts have been evolving together and giving birth to a handful of interactive softwares and concepts for computer music. With the advent of robust score following techniques with explicit musical considerations both for composer and performers and the recent flow of composers employing such systems (such as [1,

2]), the interaction between artistic use and scientific paradigms of score following is more than apparent.

This paper explores the creative use of score following and its impact on the research. The artistic cases discussed are limited to mixed electronics and instrumental pieces in the computer music repertoire. Specifically, we draw lines between the *scientific* and *artistic* goals of score following in general, and attempt to show how the second, often underestimated in the research literature, gives rise to new scientific paradigms to explore. The scientific paradigm exposed here brings the act of composition, as the authorship of time and interaction, close to *synchronous programming paradigms* in computer science.

The pieces and concepts explored in this paper are taken from the *Antescofo*¹ [1] repertoire, an anticipatory score follower equipped with a synchronous language for realtime computer music composition and performance. *Antescofo* is probably the first score following system featuring a coupled recognition system and synchronous language. This feature of *Antescofo* came rather as a necessity from its artistic use than pure scientific endeavor.

We begin the paper by some background on the creative use of score following. We then clarify the scientific and musical goals of score following in section 3. Particularly we draw on specific architectures used in most known score following paradigms within an artistic context, and draw conclusions on specific research paradigms that should be considered within this context in section 4. We proceed in section 5 by defining the architecture in *Antescofo* that addresses these issues, and define the semantics of our synchronous action language in section 6. We finish by exposing some recent examples employing *Antescofo* in section 7 and demonstrating the discussed points.

2. HISTORICAL BACKGROUND

Score following research was introduced in [3, 4] and initially geared towards *automatic accompaniment* applications in which the computer would synchronously perform and render the accompaniment section with a live performer undertaking the solo part of a given music score. The technical paradigm of score following has passed various stages ever since, evolving from symbolic string-matching techniques, to pitch detection, and probabilistic models. For a historical overview of score following algorithms we

Copyright: ©2011 Arshia Cont et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

¹ <http://repmus.ircam.fr/Antescofo>

refer the curious reader to [5] and instead, focus on its artistic employment hereon.

Artistic uses of score following technologies have been made within two trends: *Automatic Accompaniment* and mixed instrumental and electronics pieces.

2.1 Automatic Accompaniment

The goal of automatic accompaniment systems is first to *listen* to the live performer and extract position and tempo parameters with regards to its music score, and second to perform the accompanying parts synchronous to this live performance. The accompaniment part can be either symbolic data rendered into audio via some synthesis techniques (such as in early versions in [3, 4]) or employ realtime phase vocoding techniques on an audio recording in the style of *Music-Minus-One*. Among systems employing the latter is that of Christopher Raphael [6], performing automatic accompaniment on a number of classical music repertoire.

2.2 Mixed Instrumental and Electronics Repertoire

The consensus for interaction between a live music performance and composed electronics dates back to early experiments of Bruno Maderna², Karlheinz Stockhausen and Mario Davidovsky among other composers in 1950s, through tape and instrumental pieces. Synchronization between the instrumental music and electronics was assured either by using click-tracks or active listening. Despite the new possibilities that electronics had brought into the musical language, production means of electronics had introduced enough burden in the process of composition *and* performance that this realm remained highly experimental up to the 1980s.

Shortly after the advent of score following technologies, several composers recognized the opportunities that such tools could offer both at the compositional and performative levels of computer music. The most evident application would be naturally in synchronizing a pre-written electronic score to a live performance, extending the applications of score following from automatic accompaniment to a mixed repertoire.

Besides the performative comfort in employing score following technologies, some composers immediately recognized and incorporated the new opportunities that score following would bring in authoring *interactive electronic scores* coupled with realtime capabilities sound generation and transformations. The possibility of creating interactive music systems attracted new artists and researchers, and created one of the most fruitful periods in computer music. Robert Rowe's two volumes [7, 8] demonstrates how such novel paradigms have affected different practices in computer music. Among composers exposed to these new possibilities, Philippe Manoury was one of the earliest composers who integrated interactive music systems into his compositions and as a compositional process both for authoring and live performance. In particular, Manoury's early formalizations of the paradigm in collaboration with

² The first mixed music for tape and instrument appears to be "Musica su due dimensioni" by Bruno Maderna for Flute and Tape (1952).

Miller Puckette, led to the birth of the *Max* programming environment³, further developed and integrated by other composers such as Boulez, Lippe, and Settle, and since then widely referred to as the *realtime school* of composition. The most interesting concept brought by Manoury is that of *Virtual Scores*[9] developed hereafter.

2.2.1 Virtual Scores

A virtual score is a musical organization in which we know the nature of the parameters that will be processed but not their exact outcome at runtime since they're expressed as a function of the live performance. A virtual score hence consists of electronic programs with fixed or relative values/outcomes to an outside environment. A realtime electronic process is therefore one that exists in a music score, next to the instrumental transcription, and whose outcome is evaluated during live performance and as a function of the instrumental part's interpretation with all its diversity and richness.

The idea of virtual score is thus to bring in both the performative and compositional aspects of computer music within one compositional framework. A score following technology is then responsible for enabling the communication channels between the computer and the musicians according to a score and by allowing complex musical interactions similar to that of human musicians.

The framework of *virtual scores* is present and at the core of most interactive programming environments in computer music today. Despite its similarity to a traditional framework of composition, it does not limit its practice to traditional norms of music composition and on the contrary it has integrated non-traditional practices of computer music such as interactive composition [10], hyperinstrument composition [11], composed improvisations [12] and more, as employed in Manoury's early realtime pieces among others [13].

It is worthy to note that the realtime school was subject to constructive and interesting criticisms and debates at its very inception in a 1999 issue of *Contemporary Music Review* journal. Of particular interest to our work are that of Risset [14] and Stroppa [15] who underlined the lack of compositional and temporal considerations in existing frameworks at the time.

We would like to emphasize that interactive pieces in this sense, are not dissociable from *automatic accompaniment* paradigms in their architecture. Where automatic accompaniment deals with notes or chords in the accompaniment rendering, virtual scores replace them with processes and realtime electronic programs and transformations.

2.2.2 Score Following in Practice

To motivate further discussions, we attempt to provide the architectural design of a typical realtime mixed piece employing score following technologies. We use the "Introduction" part of the piece *Anthèmes II* composed by Pierre Boulez for violin and live electronics (1997) as demonstrated in figure 1. This music score shows the instrumen-

³ In fact, Manoury's pieces *Jupiter* and *Pluton* can be considered as the first historical *Max* pieces and patches.

Figure 1. First two bars of *Anthemes II* by Pierre Boulez, for violin and live electronics (1997).

tal (violin) section in parallel to an approximative notation for the realtime electronics accompanying the system. Each system corresponds to a specific electronic process, whether realtime or samplers, by themselves accompanied by spatialization parameters. The sequencing of electronics in this score are either notated as relative to the performance tempo or fixed absolute values if necessary. The circled numbers in the score correspond to *synchronization points* between the instrumental and the electronics scores.

Figure 2 shows a generalized design diagram for the computer music realization of a mixed piece similar to the one in figure 1. This diagram generalizes most mixed electronic pieces employing score followers in the Ircam repertoire or employing *MaxMSP* or *PureData*⁴. It demonstrates the common trend which consists of having separate *instrumental* and *electronics* scores. The instrumental score plus synchronization tags (circled numbers in figure 1) are fed into the score follower which takes care of online alignment. The electronic score in turns is stored as tagged sequential data-structures (commonly referred to as *qlists* in *Max* and *PureData*). The electronic queues store variable/message pairs attached to symbolic time indexes and usually scheduled on a milli-seconds basis. The symbolic time indexes (tags) would then correspond to synchronization pivots in the instrumental score, destined for live synchronization. The modularity of environments such as *Max* or *PureData* allow co-existence of multiple sound processes in a single patch that can be controlled through the sequential electronic score.

The general diagram of figure 2 can be seen as two complementary systems: an *interactive system* consisting of the score follower and the musician, and a *reactive system*

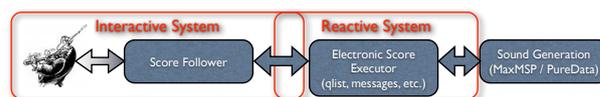


Figure 2. General diagram of a typical interactive mixed music.

consisting of the electronic score and its sequential execution as a reaction to the received tags from the interactive system. The two components are *interactive* and *reactive* due to their implicit nature of time and following [16]. The score follower is an *interactive systems* since it should be considered as part of the physical world (with the musician), and the second *reactive* since it runs on its own implicit clock and in reaction to an external environment.

3. SCIENTIFIC AND MUSICAL GOALS OF SCORE FOLLOWING

The score following literature is constantly increasing every year with new algorithmic contributions. It is important to distinguish between the *scientific goals* and *artistic goals* of such systems. Our argument here is that while the two goals are non-dissociable, they are however distinct and achieving one does not necessarily entail the other, and hence new research paradigms should be explored.

3.1 Scientific Context

Score following, in its scientific context today, deals with correct alignment of audio streams onto a symbolic score and correct extraction of musical parameters of the interpretation in hand. Besides automatic accompaniment systems, score followers and automatic alignment systems are

⁴ See Pd Repertory Project: <http://crca.ucsd.edu/~msp/pdrp/latest/files/doc/>

employed as front-ends for many MIR applications such as score-informed editing softwares and source separation. It is clear that in such applications the alignment and extraction precision is of utmost importance and much algorithmic effort has been dedicated to achieve this. Figure 3 shows a simplified diagram of this aspect of score following where the computer is aware of the symbolic score expected to be performed by the live performer.

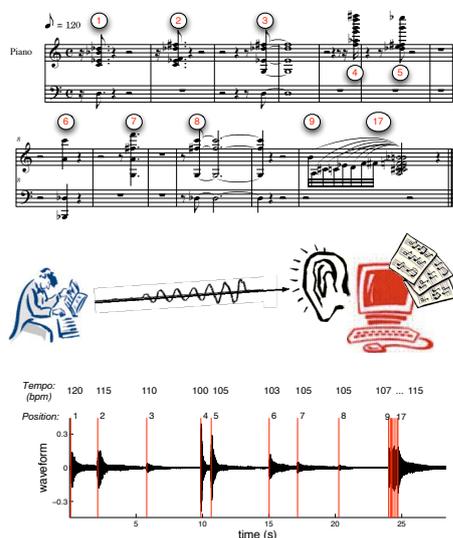


Figure 3. General diagram for an online alignment system.

Most score followers in the literature focus on this aspect of its applications. This is particularly clear by the effort of the community for evaluations of such systems [17, 18].

3.2 Artistic Context

Despite advances in alignment systems, very few systems have explicitly considered the artistic goals of score following systems and their direct design consequence. The artistic goals of score following as an interactive music system, and their discrepancy with its scientific goals can be discussed within two folds:

1. *Realtime Music Performance:* Score following is naturally a tool for realtime performance of mixed instrumental and live electronics. The discrepancy between the scientific and artistic goals in this context has much to do with the idea of *robustness* of the realtime alignment algorithms in score followers, especially in the context of live performance and realtime processing within uncertain environments. An ideal musical performance for an architecture depicted in figure 2 can be naturally achieved if the interactive system in hand has 100% precision in the recognition phase and given any performance. While recent systems demonstrate high precision and performance in realtime, the architecture in figure 2 is probably a bad choice for the musical finality of an interactive music system. To summarize, the musical goal of such interactive systems require that the *musical output* is acted upon expected despite any

error from the live performer or the recognition system.

2. *Authoring and Composition of Realtime Music Processes:* The ultimate goal of score following as an interactive music system is naturally to express reactive programs that create the electronics part during the composition phase. This task requires a minimum of *musical expressivity* within the language that describes such interactions and gets naturally close to the idea of *virtual scores* as discussed earlier. Treating the *interactive* and *reactive* phases of an interactive piece separately as depicted in figure 2 most often evades expressing such interactions and most existing systems have found comfort in leaving this very important issue apart.

To demonstrate the above issues, consider again the score in figure 1 as an example: The first issue is clear by differentiating electronic events tagged by ① and ③. Event ① does not necessarily depend on the recognition of the first note in bar 1 (an *F*) while event ③ can be considered as having a local scope. If the musician or the recognition system misses the first high *F* in the violin part, it is evident that ① should not be dismissed. This is however not true for ③. If the chord corresponding to ③ is missed, that event can be consequently dismissed in order to maintain a musical coherence in the output. The second issue has much to do with the authoring of the electronic events presented graphically in figure 1. Parallel lines in this score correspond to *concurrent* electronic programs which are expected to output synchronously during live performance with their timing notated relative to the live performer's tempo. Most realtime programming environments however do not neither allow such timing expressivity for programming as time is usually expressed in absolute values, nor explicit concurrency in expressing electronic processes.

4. CONSIDERATIONS FOR CREATIVE USE OF SCORE FOLLOWING

With the above introduction, we draw important requirements for the use of score following as an interactive music system, destined both for *composition* and *performance*:

4.1 Time is Resource

Explicit modeling of *time* is of utmost importance for a score following system both at the recognition phase (for realtime performance) and authoring (composition). However, most score following techniques have focused on the event level (pitch, spectrum observation, etc.) and left temporal models approximate or implicit. At the same time, any music score contains important timing information such as tempo, relative durations and timing hierarchies within elements, that can help both phases of score following use. This issue is of extreme importance when such systems are to be employed in realtime (and thus in absence of future information for decoding), and can significantly enhance

recognition and also access to temporal elements for computer music composition. The only practical score following systems which consider explicit time models for both phases are Music Plus One[2] and Antescofo[5] where tempo and event durations are first-class citizens in the systems and are employed both during recognition and accompaniment. In [2], temporal considerations are taken into account as a secondary pass and cascaded to an event recognition HMM based system, requiring offline learning of time parameters for best performance. In *Antescofo* an explicit time model is coupled with an audio recognition system through *Anticipatory Learning*, attempting to reduce complexity of computation and with no requirement for off-line learning. *Antescofo* in particular makes time and tempo variables explicitly available for programming reactive electronics.

4.2 Heterogeneous Models of Time

Any classical piece of music has multiple and heterogeneous models of time, which should be explicitly considered in the conception of any interactive music system. For example, grace notes in classical music are typical of *atemporal* events which exist spatially in a score but do not contribute to the tempo variations as opposed to *temporal* events (regular notes and chords). Glissandos whenever the instrumentation allows are also typical of *continuous time* events as opposed to *discrete time* events in regular notes and chords. Finally, trills and tremolos in the classical repertoire undergo *hierarchical time* structures where the global event itself can occupy a discrete or continuous duration while its internal elements can constitute (for example) atemporal elements.

Presence of heterogeneous times is more than evident in the contemporary music repertoire and more than essential in expressing electronic processes. An electronic process can contain discrete (relative or absolute time) events as well as continuous controls, or in some cases recursive processes. The point here is that such considerations are neither unique to electronic music, nor to any specific style of music. Western music notation has internalized such temporal structures that are in use by all composers and performers while computer music languages are still behind in terms of expressivity of time and their models. We will come back to this issue later in section 5.

4.3 Critical Safeness

The musical output of an automatic accompaniment or score following system should not solely depend on the recognition system, or even to the live performer at some instances. This is in analogy to human coordination for ensemble performance: A live music performance should be smooth in time, and does not halt in presence of any error in realtime. As discussed in section 3, one of the discrepancies between scientific and artistic goals of a score following system is the issue of live performance and robustness of the recognition in realtime. We showed on a simple example in section 3.2 how a simple specification of electronic processes can save the musical output despite

any error from the environment or the interactive recognition system. Interestingly the issue of critical safeness is the subject of study in most realtime systems [19] and already employed in the industry. These paradigms should also be adopted for score following systems.

4.4 Authoring of Time and Interaction

The most important issue for creative use of score following, is in *how* such systems would bring live interaction as a first-class citizen in the compositional phase and enable an authoring of time and interaction for artists. While common computer music programming environments enable live interaction with musicians, they are particularly poor for authoring of time and interaction for composed music. The lack of explicit authoring tools of this kind has led to a common division between the performative and compositional aspects of computer music [20], criticized thoroughly by several pioneers of computer music [14, 15], and has been the subject of debate in a recent colloquium on the subject between various artistic disciplines [21].

This issue is directly related to domain-specific computer language design, and in our case for realtime interactive computer music. We believe that this topic should not be treated separately from common technical considerations of score following systems and is directly related to the musical goals of such systems.

5. ANTESCOFO'S ARCHITECTURE

Antescofo is the latest incarnation of score following technologies at Ircam since 2008[1]. It is a realtime score following technologies that aims to integrate the points discussed in section 4 within one single environment. To this end, it consists of a state-of-the-art realtime alignment system, capable of aligning complex polyphonic instruments as well as decoding the realtime tempo of live performance. The novelty of the recognition system in *Antescofo* lies in its coupling of a realtime audio and tempo agent, and capability of handling heterogeneous times during the recognition phase. It is an *Anticipatory System* with an attempt to predict event positions in the future in order to aid recognition and undertaking of electronic actions. We leave detailed discussions of the recognition algorithm to [5] and instead focus on its musical aspects with regards to points discussed in the previous section.

Antescofo is destined for interactive mixed instrumental and electronics pieces and aims at bringing both *interactive* and *reactive* components of a typical piece within this repertoire as discussed in section 2.2.2 within one single framework. Figure 4 shows the general diagram of this architecture. Within this architecture, electronic programs are handled within one framework that allows employment of various time scales and coupling of electronic actions to the live tempo if needed. *Antescofo* can be employed as a traditional score follower, in which case realtime positions and tempo of the performance are obtained as the module's direct outputs during performance. The composer can optionally integrate electronic messages inside the instrumental score, in which case, *Antescofo* handles

their message-passing to host programs that produce the electronics part. In this sense, *Antescofo* is used both during the compositional phase as an authoring tool for programming synchronous electronic events with regards to the instrumental score, and also in the performance phase by attempting to produce the desired musical output as de-

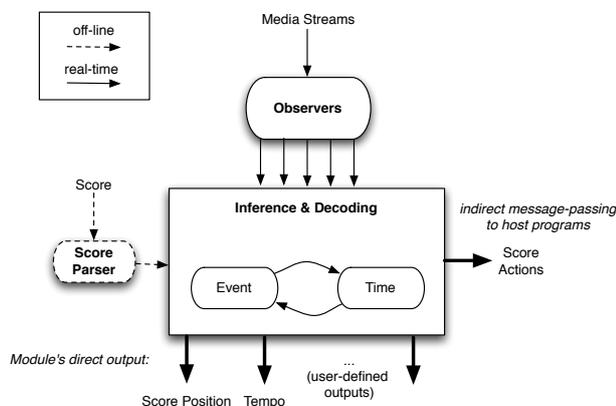


Figure 4. *Antescofo*'s general architecture, comprising of both *interactive* and *reactive* systems of figure 2 .

Antescofo can thus be used as compositional resource during the authorship of both instrumental and electronic score, on top of traditional use of score followers in live performance paradigm. The use of *Antescofo* as an authoring tool is possible because of the coupling of the recognition paradigm (natural to any score follower) with a synchronous realtime language for computer music composition. The synchronous language aspects of *Antescofo* has brought into focus the musical goals of such interactive music systems, which has been underestimated so far as a research paradigm itself. In the following section, we define briefly important aspects of this language with regards to points mentioned in section 4.

6. SEMANTICS OF PERFORMANCE SYNCHRONOUS LANGUAGE IN ANTESCOFO

The musical goal of score following has to deal with both the compositional and performative aspects of the piece of music in question. In the compositional phase, it has to be able to describe electronic processes in parallel to and ordered with regards to instrumental scores, and by employing the rich temporal semantics of musical intellect. The performative phase of such systems is responsible for evaluating electronic processes at a given position and tempo and as a reaction to the live performance. In this respect, an electronic score in a mixed interactive piece, is in close analogy to an orchestral or accompaniment score except that simple notes are replaced by programs with heterogeneous notions of time, and whose outcomes are not known in advance but deterministic in a musical context. Technically speaking, it is a reactive program with the ultimate goal of *determinacy* (in the computer science term), correct ordering at runtime, and (musical) critical safetiness. Such paradigms have been widely studied in the computer

science literature for *realtime synchronous languages* [16] and widely applied in the industry for realtime critical systems. Our goal in this project is to adopt a musical semantics for such languages, whose application paradigms seem to be closely related to the acts of composition and performance. We will not expose the syntax of the language and leave it to curious users, and instead focus on the constructive semantics that allow an authoring of time and interaction in computer music.

An important consequence of the architecture discussed earlier is the coexistence of the *instrumental score* and *electronic score* within one single score. An *Antescofo* score contains two semantics: one for describing the music score of the human performer, and another for describing electronic events in an *action semantics*. Both semantics are capable of describing multiple scales of time (absolute, pulsed, continuous) heterogeneously within one score. The electronic score is a simple *message-passing coordination language*, where messages are bound to symbols, ordered and grouped as desired to imitate a musical score. These synchronous messages are then scheduled in realtime to be delivered timely to electronic modules. The choice of such semantics is in accordance with the wide practice of interactive music within *MaxMSP* and *PureData* programming environments. The goal of the *action semantics* in *Antescofo* is to provide expressivity for authoring of time and interaction and the synchronous scheduling of events in realtime. The primitives of this semantic consists of:

Discrete Events Message(s) bound to symbols with an optional delay. The delay can be in absolute time or relative to tempo and thus evaluated in runtime and reactive to tempo changes for synchronous output.

Continuous Events Similar to Break-Point Functions (BPF) where output is interpolated between discrete elements and scheduled in relative or absolute time.

Periodic Events A constructive semantic that allows (absolute or relative) periodic discrete or continuous messages, running forever when launched unless *killed* somewhere in the score.

On top of these primitives, the user can employ the following constructive semantics:

Parallel Groups Primitives above, can be optionally *grouped* to construct polyphonic phrases in the electronic score using an optional process name. This semantic is there to bring polyphonic authorship as well as independent but relative timing between groups of electronic phrases. This feature also brings a *temporal scope* for each group within the score during composition which is respected in runtime using the synchronous scheduling relative to tempo and position.

Nested Hierarchies Groups can be nested hierarchically and recursively to allow independent but ordered timings; respected during realtime scheduling.

Macros Evaluated at score load (and non-runtime) in order to provide motivic patterns both in message content and timing for composition of the electronic score.

Dataflow Functionals Mathematical expressions evaluated at runtime, useful to make message content relative to external variables.

Each block of programs in *Antescofo* accept specific attributes. Among such attributes, composers can specify the *scope* of each program dealing with their critical safety in case of performance errors, and also the ability to define independent or varying tempo (accelerando or rubato) relative to the performance tempi.

The development of the above primitives and compositionals have been undertaken incrementally and by observing various uses of interactive systems in composition and performance. *Antescofo* language is currently text-based with graphical support through *NoteAbility Pro* notation software editor⁵. We believe that a thorough and well-defined semantics can give rise to graphical semantics of programming which is already the case for synchronous languages within the avionics industries [22].

7. EXAMPLES

In this section, we aim at representing the compositional aspects of the semantics described above. It goes without saying that a thorough presentation of this system is within a performance situation and live coordination of electronic programs with the live performer within the new score following paradigm. Curious readers can refer to *Antescofo* website⁶ for online videos and upcoming performances with the system.

Figure 5 shows an excerpt of the *Antescofo* score of “Otemo” for Vibraphone and live electronics by the composer Vassos Nicolaou [23] in the *NoteAbilityPro* score editor. The top staff is the Vibraphone score, where as the 11 bottom staves show the grouped primitives, similar to polyphonic lines, for the electronic scores. In realtime performance, each group is launched according to the performer’s position and during the entire life of each group (shown here as their length) rescheduled according to detected position and tempo to assure synchronicity.

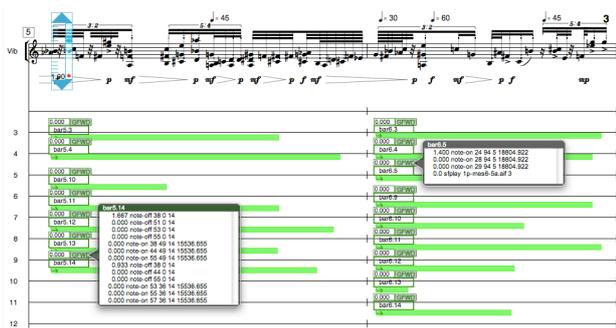


Figure 5. *Antescofo* grouped primitives visualization in NoteAbilityPro, excerpt from Vassos Nicolaou’s *Otemo* for Vibraphone and Live electronics (2008)

Figure 6 shows an excerpt graphical representation of the electronic part for “Hist Whist” by composer Marco Stroppa

for violin and chamber electronics as written in *Antescofo*. The timeline of the score is on the x-axis, where as the vertical bar demonstrate individual, parallel and nested processes within each block. This excerpt makes extensive use of nested macros written by the composer that control rhythmic progression of harmonization values and their amplitudes running on the realtime audio from the violin. It consists of 8 main blocks corresponding to 8 generated groups, each having hierarchical periodic primitives with periodic *kills* to imitate a rhythmic progression. Each pair actuates on one harmonizer module (one for transposition values and other on amplitudes), making it a total of four polyphonic realtime processors.

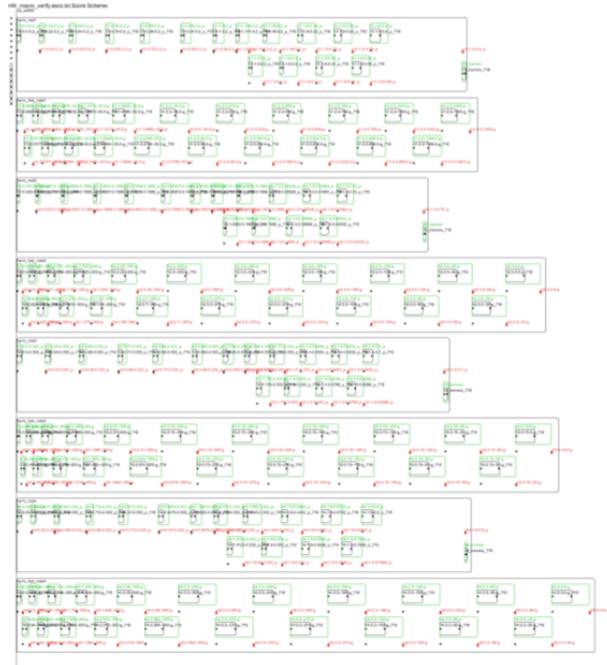


Figure 6. Visualization of electronic processes (excerpt from movement 3) of *Hist Whist* by Marco Stroppa, for violin and chamber electronics (2009).

The *Antescofo* score in figure 6 actually consists of 8 lines calling two macros with different arguments. Each macro recursively calls others defined by the composer to create the above patterns. The final outcome consists of more than 100 ordered and concurrent actions that will be synchronized to the performance in realtime.

The visualization in figure 6 is an experimental feature in *Antescofo* allowing composers to verify visually the contents of the generated scores during composition. It can be seen as hierarchical automata constructing a complex temporal system with resemblance to visual approaches in synchronous languages [24].

8. CONCLUSIONS

Whereas traditional score following paradigms have put emphasis on high precision in alignment and extraction of symbolic parameters from realtime audio, the creative use of such systems infer other goals which have been underestimated as a research paradigm in the literature. In this

⁵ <http://debussy.music.ubc.ca/NoteAbility/>

⁶ <http://repmus.ircam.fr/antescofo>

paper we attempted to show those missing paradigms that deal with authorship of time and interaction, critical safety of realtime electronic scores during performance, heterogeneous representations of time, and explicit models of time, both for composition and performance of interactive computer music.

We showed the close relation between the creative use of score following systems with that of realtime synchronous programming, bridging the gap between compositional and performative aspects of computer music, and bringing the rich expressivity of musical vocabularies into a simple computer language. The emergence of this research paradigm is mostly due to creative uses of score following systems which do not hesitate to rethink our practices and interfaces with computers for making music. We believe that this synergy will create an important momentum between artists and researchers in the years to come and hope that this paper has shed some lights on the importance of this new paradigm in both communities.

9. REFERENCES

- [1] A. Cont, "Antescofo: Anticipatory synchronization and control of interactive parameters in computer music," in *Proceedings of International Computer Music Conference (ICMC)*. Belfast, August 2008.
- [2] C. Raphael, "Music Plus One: A System for Expressive and Flexible Musical Accompaniment," in *Proceedings of the ICMC*, Havana, Cuba, 2001.
- [3] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," in *Proceedings of the International Computer Music Conference (ICMC)*, 1984, pp. 193–198.
- [4] B. Vercoe, "The synthetic performer in the context of live performance," in *Proceedings of the ICMC*, 1984, pp. 199–200.
- [5] A. Cont, "A coupled duration-focused architecture for realtime music to score alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 974–987, June 2010.
- [6] C. Raphael, "The informatics philharmonic," *Commun. ACM*, vol. 54, pp. 87–93, March 2011. [Online]. Available: <http://doi.acm.org/10.1145/1897852.1897875>
- [7] R. Rowe, *Machine Musicianship*. Cambridge, MA, USA: MIT Press, 2004.
- [8] —, *Interactive music systems: machine listening and composing*. Cambridge, MA, USA: MIT Press, 1992.
- [9] P. Manoury, *La note et le son*. L'Hamartan, 1990.
- [10] J. Chadabe, "Interactive composing: An overview," *Computer Music Journal*, vol. 8, no. 1, pp. 22–27, 1984.
- [11] T. Machover and J. Chung, "Hyperinstruments: Musically intelligent and interactive performance and creativity systems," in *International Computer Music Conference (ICMC)*, 1989, pp. 186–190.
- [12] X. Chabot, R. Dannenberg, and G. Bloch, "A workstation in live performance: Composed improvisation," in *International Computer Music Conference (ICMC)*, Octobre 1986, pp. 537–540.
- [13] M. Puckette and C. Lippe, "Score Following in Practice," in *Proceedings of the ICMC*, 1992, pp. 182–185.
- [14] J.-C. Risset, "Composing in real-time?" *Contemporary Music Review*, vol. 18, no. 3, pp. 31–39, 1999.
- [15] M. Stroppa, "Live electronics or live music? towards a critique of interaction," *Contemporary Music Review*, vol. 18, no. 3, pp. 41–77, 1999.
- [16] N. Halbwachs, *Synchronous Programming of Reactive Systems*. Kluwer Academics, 1993.
- [17] A. Cont, D. Schwarz, N. Schnell, and C. Raphael, "Evaluation of real-time audio-to-score alignment," in *International Symposium on Music Information Retrieval (ISMIR)*. Vienna, Austria, October 2007.
- [18] ScofoMIREX, "Score following evaluation proposal," webpage, August 2006. [Online]. Available: http://www.music-ir.org/mirex/2006/index.php/Score_Following_Proposal
- [19] N. Storey, *Safety critical computer systems*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1996.
- [20] M. Puckette, "Using pd as a score language," in *Proc. Int. Computer Music Conf.*, September 2002, pp. 184–187. [Online]. Available: <http://www.crca.ucsd.edu/~msp>
- [21] Ircam, "Colloque international écritures du temps et de l'interaction," in *Agora Festival*. Paris, France.: Ircam-Centre Pompidou, June 2006.
- [22] F. Dormoy, "Scade 6: a model based solution for safety critical software development," in *Proceedings of the 4th European Congress on Embedded Real Time Software (ERTS'08)*, 2008, pp. 1–9.
- [23] S. Lemouton and V. Nicolaou, "Polyphonic audio score following: The otemo case," Ircam - Centre Pompidou (Composer in Research Project 2009), Tech. Rep., 2009. [Online]. Available: <http://articles.ircam.fr/textes/Lemouton09c/>
- [24] D. Harel, "StateCharts: a Visual Approach to Complex Systems," *Science of Computer Programming*, vol. 8-3, pp. 231–275, 1987.

ENSEMBLE: IMPLEMENTING A MUSICAL MULTIAGENT SYSTEM FRAMEWORK

Leandro Ferrari Thomaz and Marcelo Queiroz

Computer Science Department – University of São Paulo – Brazil
{lfthomaz | mqz}@ime.usp.br

ABSTRACT

Multiagent systems can be used in a myriad of musical applications, including electro-acoustic composition, automatic musical accompaniment and the study of emergent musical societies. Previous works in this field were usually concerned with solving very specific musical problems and focused on symbolic processing, which limited their widespread use, specially when audio exchange and spatial information were needed. To address this shortcoming, *Ensemble*, a generic framework for building musical multiagent systems was implemented, based on a previously defined taxonomy and architecture. The present paper discusses some implementation details and framework features, including event exchange between agents, agent motion in a virtual world, realistic 3D sound propagation simulation, and interfacing with other systems, such as Pd and audio processing libraries. A musical application based on Steve Reich's Clapping Music was conceived and implemented using the framework as a case study to validate the aforementioned features. Finally, we discuss some performance results and corresponding implementation challenges, and the solutions we adopted to address these issues.

1. INTRODUCTION

In this paper we discuss implementation strategies and report recent experience with *Ensemble*, a musical multiagent framework first presented in [1]. The multiagent approach is well-suited for musical applications involving a number of autonomous musical entities that interact musically with one another, such as electronically-mediated collective performance [2, 3], automatic accompaniment and improvisation [4, 5], and biologically-inspired musical societies (used for studying emergent behaviors) [6, 7, 8, 9].

Although the literature on the use of agents in music is rather extensive, most of it deals with very particular problems [1]. Two previous works have much more general goals and are deeply connected to the present work, deserving special attention. The MAMA architecture [4] offers a framework for designing musical agents with interactive

behavior based on the speech act theory, which communicate using MIDI messages to perform a musical piece. The SWARM Orchestra [2] is an user-extendable library that deals with large and complex populations (swarms), which may be used to control several musical and motion parameters simultaneously.

The *Ensemble* musical multiagent framework, which was first proposed in [1], builds up on the ideas of these two systems to define a generic, extendable, and configurable framework. Two general types of agents are considered in this framework: musical agents, which inhabit a virtual environment and interact with one another via sensors and actuators, and an environment agent, which controls the virtual environment and all interactions therein.

Musical agents are autonomous pieces of software that may embody interactive musical algorithms, or may also serve as virtual proxies to external agents, such as instrumentalists or even other musical software systems. They can also serve as sound outlets, capturing sound at specific positions in the virtual environment and sending them out for playback on a real listening space, such as a concert hall or an installation space, using either loudspeakers or headphones.

Interactions are modelled as events, which can be of several types, such as sound events, motion events, visual events and textual/symbolic messages, and each event type is controlled by an event server (which is part of the environment agent).

Musical agents can be specified using initialization files or can be created and modified at runtime. Agent design allows agent components, including sensors and actuators, agent reasonings and sound-processing engines, also to be added and removed at runtime, making this a pluggable framework.

Interfacing the architecture with popular sound processing languages and environments, such as Pd or Csound, is also a major concern. Currently, agent creation and modification, as well as on-the-fly control of agent motion, sensing and acting, can all be done using Open Sound Control (OSC) messages [10].

Ensemble extends the functionalities of [4, 2] by aggregating many novel features, such as multimodal communication (audio, MIDI and text-based) between musical agents, pluggable components for defining agents and physical characteristics of the virtual environment, and 3D sound propagation simulation within the virtual world. In particular, audio exchange between agents and a realistic treatment of space and acoustics, both poorly explored in pre-

vious works, are defining characteristics of this work.

This paper is structured as follows. Section 2 discusses the specific details of the implementation of the framework, and also the specification of agents and components by the user. Section 3 presents a concrete musical application of the system, based on Steve Reich's *Clapping Music*, as a case-study to illustrate the framework from the user point-of-view. Finally, some concluding remarks and pointers to further work are given in section 4.

2. FRAMEWORK ARCHITECTURE AND IMPLEMENTATION

This implementation was coded in the Java SE 6 language. Although Java performance limitations and poor sound processing support are well known to the community, this choice was made so that any musical applications programmed by the user could be run on distinct platforms. The JADE 4.0 multiagent middleware¹ was chosen for being a well-documented and well-supported multiagent platform.

This framework can be classified as a *white-box* framework [11], since the user is required to have some knowledge of its internal implementation. User specific implementations and extensions to the system need to follow a few internal conventions, since the framework acts as a main program, calling user-defined methods. Nevertheless, we provide a reasonable amount of reusable components (such as analysis and synthesis engines) which may ease considerably the specification of a musical agent by the user.

Simplified UML class diagrams for the *MusicalAgent* and the *EnvironmentAgent* can be seen in figures 1 and 2, respectively. These two kinds of agents are based on the *EnsembleAgent* class, itself a subclass of JADE's *Agent* class, which provides basic functionalities such as mechanisms for message passing and scheduling/executing concurrent activities, as well as the definition and control of agent life cycles.

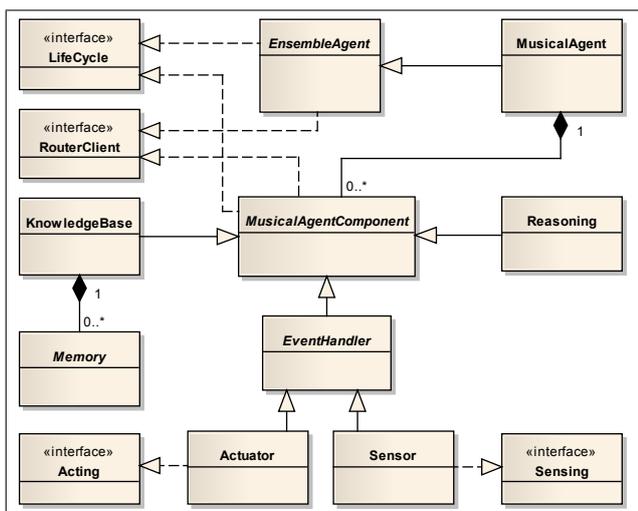


Figure 1. Class Diagram of the Musical Agent.

¹ Available at <http://jade.tilab.com/>. This and every other link mentioned in this text has been verified march 25th 2011.

A *MusicalAgent* is composed of one *KnowledgeBase* object (for holding multiple data, such as I/O sound information) and possibly several *MusicalAgentComponent* objects, as shown in figure 1. These components can be of two types: *Reasoning* components, which are responsible for an agent's decision processes, and *EventHandlers*, i.e. *Actuators* and *Sensors*, capable of interacting with the environment through corresponding *Acting* and *Sensing* interfaces.

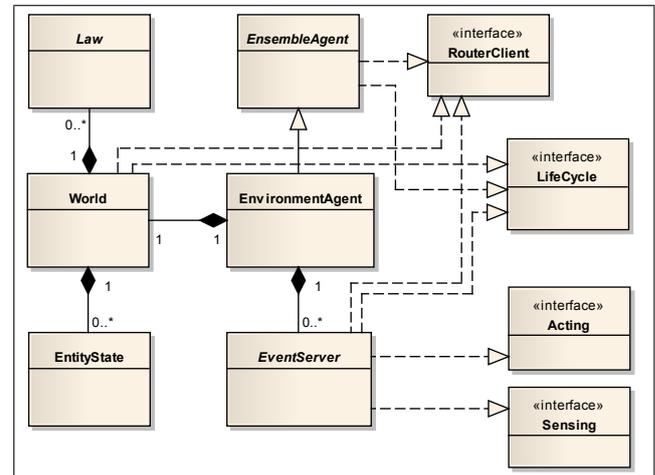


Figure 2. Class Diagram of the Environment Agent.

A special singleton agent, the *EnvironmentAgent*, represents the virtual environment and manages all interactions between *MusicalAgents*. As shown in figure 2, it is composed of a *World* object, for describing the virtual environment, and *EventServers*, for mediating event exchanges between *EventHandlers*. The *World* object contains the physical description of the space (number of dimensions, connectedness, boundaries) and also stores the current state of all entities (agent positions, motion intentions, sound produced and received, etc.) in *EntityState* objects. *Law* objects define the way the world state changes, i.e. they describe how to update the description in the *World* object given the last state, the current time instant and all actions currently performed by the agents. For example, realistic 3D sound propagation is defined by a particular *Law* object that receives all sound produced anywhere in the environment and delivers a specific mixture to each sound *Sensor* according to its position relative to each sound *Actuator*.

Major components of the framework (agents and their aggregated components) implement a *LifeCycle* interface. *LifeCycle* methods are: *configure()*, to set up configuration parameters before startup; *start()*, used by the framework to start each component; *init()*, user-specific initialization, implemented by the user and called automatically by *start()*; *stop()*, used by the framework to stop each component; and *fini()*, user-specific finalization, implemented by the user and called automatically by *stop()*. This approach provides greater flexibility when extending components, while ensuring the necessary control to the framework.

2.1 Event Exchange

Ensemble supports two kinds of event exchange methods: *sporadic*, where events can be sent at any instant and rate (e.g. changing position of an agent or sending a MIDI or text message); and *periodic*, controlled by a synchronous communication process with a fixed exchange frequency, where in each cycle *Actuators* are requested to produce *Events* and *Sensors* receive corresponding *Events* (e.g. audio communication). Both types of event exchange methods are controlled by corresponding *EventServers*, and *Actuators* and *Sensors* interested in that event type (audio, for instance) are required to register prior to participating in the communication process.

The periodic exchange mode is controlled by state machines built into the *EventServer* and registered *Actuators*. State changes are regulated by the Virtual Clock service, responsible for timing the current simulation and scheduling tasks. Figure 3 shows an UML sequence diagram of a complete cycle of a periodic event exchange between an *EventServer* and an *Agent* equipped with a *Reasoning*, an *Actuator* and a *Sensor*.

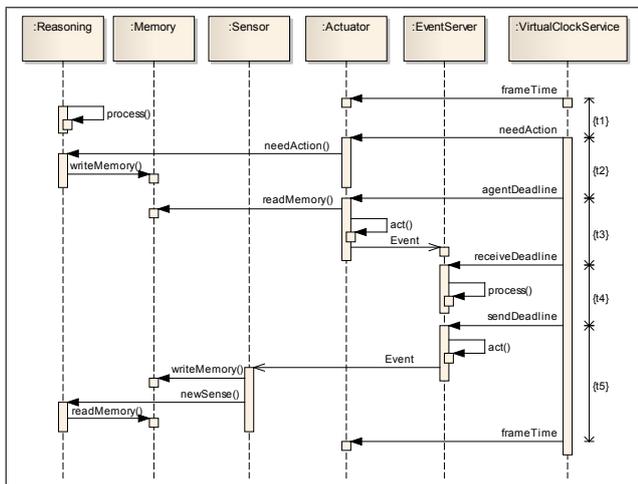


Figure 3. Sequence diagram of a periodic event exchange cycle. Time runs downward according to the timeline at the right of the figure.

At the beginning of each frame, there is a time interval for the agent to process any information that it wants to send in the next frame. When the deadline *needAction* is reached, the *Actuator* calls a *needAction()* method of the *Reasoning* responsible for writing the required information in the *Actuator*'s memory. The *agentDeadline* indicates that an *Event* must be sent through an *act()* method; failure to meet this deadline will result in no *Event* being sent (an empty audio frame, for instance). The *EventServer* waits for all arriving *Events* until the *receiveDeadline*, when it starts its own *process()* method, updating the *World* state and sending back response *Events* for registered *Sensors*, no later than the *sendDeadline*. When a *Sensor* receives an *Event*, it writes the corresponding information in its *Memory* and informs the *Reasoning* about it through the *newSense()* method. Finally, at *frameTime*, a new exchange cycle begins.

A *Reasoning*'s *process()* method can be defined as an endless loop, or it can be triggered by the *needAction()* method. All deadlines are user-defined, as they depend on the amount of time needed for each *process()* method and for the communication between *Agents* and *EventServers*.

2.2 Agent Memory

An agent's *Memory* is the part of its *KnowledgeBase* used to store incoming and outgoing *Events*, analogously to a computer's memory. Every *Sensor* and *Actuator* has an associated memory, which is automatically filled in when a corresponding *Event* is received by a *Sensor*, and read from when an *Actuator* is asked to act. When a *Reasoning* component wants to send out some *Event* (sound, for instance), it stores the information in the corresponding *Actuator*'s *Memory* and triggers the corresponding action. When an *Event* is received by a *Sensor*, it stores the information in its memory and lets all registered *Reasonings* know about it.

Memories are time-based, since all events have timestamps and durations, which are used to fill in the *Memory* or read from it. Although *Sensors* and *Actuators* follow a linear-forward time policy when accessing memories, there are many other components which may be interested in nonlinear or even random access, granted by generic *read(instant, duration, unit)* and *write(instant, duration, unit)* *Memory* access methods.

Two kinds of memories based on the same *Memory* interface were implemented. The simplest one is the *EventMemory*, which stores events in a double linked list, ordered by timestamp. Since search time tends to grow linearly with list size for regular linked lists, a simple heuristic was used to improve memory access performance for common situations. When processing audio or movement, sequential memory accesses are most likely to occur, and so the last seeked timestamp is used as a starting point for successive memory accesses.

A specialized *AudioMemory* was designed for dealing with audio data, which is implemented as a circular buffer of samples with a parameterized sample rate. This memory can be accessed using timestamps and also sample indices, using interpolation for continuous (floating-point) memory access. Linear interpolation is used by default, but non-interpolated or *N*-point interpolated accesses are easily configurable.

2.3 Agent Motion

A *MovementEventServer* is responsible for managing agent movement requests and updating agent positions. Agents equipped with movement actuators are able to change their position in the virtual environment using simple instructions such as WALK, ROTATE and STOP. Depending on the *World* definition (and its *Laws*) these may be used to instantly update an agent's position, to instantly start moving with a given velocity towards a certain direction, or more realistically, to define an acceleration (as if induced by a physical force) and let the *EnvironmentAgent* compute the corresponding trajectory using basic (i.e. Newtonian) mechanics. This is defined by a specific *MovementLaw*.

The rate at which the *MovementEventServer* updates its data about agent positions (within the corresponding *EntityState*) is defined by the user. At each update cycle, the server checks if any agent has pending movement instructions to be carried out. Friction can also be considered in the simulation, with user-defined friction coefficients. The *MovementEventServer* can check for obstacles, such as other agents or walls, thus restricting an agent's movement. Agents can be informed of the result of their movement requests (and their updated position in the environment) via specific *MovementSensors*.

A *MovementReasoning* was conceived to help design agent trajectories defined by waypoints and time-of-arrival constraints. This *Reasoning* sends out acceleration instructions to the *EventServer* through a movement actuator, and monitors the agent's actual position using a movement sensor.

2.4 Sound Propagation

Realistic and reliable 3D sound propagation simulation within the virtual environment was one of the central issues in the design of *Ensemble*. This corresponds to having each agent hear/sense exactly what it would in a real scenario, according to the positions of its sound *Sensors* and the positions of every sound *Actuator*, their corresponding velocities (with an implied Doppler effect), attenuation effect due to distance, sound shadows cast by obstacles, etc. This is of paramount importance when a sound design is going to be reenacted in a real listening environment, such as a concert hall or an installation, and the impression of a realistic spatial soundscape is intended.

There are a few technical issues involved in a realistic sound propagation simulation that will be discussed in the sequel. First of all, it should be noted that sound is only perceived at sound *Sensors* (and not anywhere in the space), and so a simulation of wave equations on a discretized grid representing the space would be computationally prohibitive and also useless for the most part. Instead, sound propagation is considered independently for each pair (sound *Actuator*, sound *Sensor*). Global simulation parameters such as speed of sound, attenuation due to distance and frequency filtering due to the environment can all be configured by the user.

The *SoundEventServer* is a periodic process that is required to deliver to each sound *Sensor* one audio frame per cycle, representing all incoming sound at the current position of the *Sensor*, which is allowed to vary continuously. This means that, for each sound sample of this frame corresponding to timestamp t , the *Sensor* S has a different position $Sp(t)$, and the same is true for every other sound-producing *Actuator* in the environment. So, for each timestamp t of this audio frame (to be delivered to a particular sound *Sensor*), the event server has to go through all sound *Actuators* in the environment and find out, for each sound *Actuator* A with an independent trajectory $Ap(\cdot)$, when did it produce sound that arrived at position $Sp(t)$ at time t .

Considering that sound travels in a straight path with constant speed c , the problem is to find an instant $d(t)$ in the past such that the path from position $Ap(d(t))$ to $Sp(t)$

takes exactly $t - d(t)$ time units, or in other words, to solve the following equation in the variable d for each given t :

$$Sp(t) - Ap(d) = c(t - d).$$

Since the functions $Sp(t)$ and $Ap(t)$ have simple analytic derivatives (according to the Newtonian equations), the Newton-Raphson method provides a quick way to find the solution $d(t)$ for each S , t and A . This solution for a timestamp t can be used as a starting point when finding the solution for the next timestamp $t + \Delta$, whose solution $d(t + \Delta)$ is likely to be close to $d(t)$. Experimental tests showed that, with this initialization, it takes about four iterations for the Newton-Raphson method to find $d(t)$ within a precision of 10^{-9} seconds.

Despite Newton-Raphson's efficiency, it should be noted that this problem has to be solved once for each sample n of each sound *Sensor* S and for each sound *Actuator* A , with a total of $(\#Sensors) * (\#Actuators) * (FrameSize)$ calls to this function. For instance, in a very simple setting of two *Sensors* and two *Actuators* using a *FrameSize* of 100 ms with a 44.1 kHz sample rate, it would take 17640 function calls or about 70560 Newton-Raphson iterations to complete each processing cycle, which gives less than 1.5 μ s of CPU time per Newton-Raphson iteration, only for sound propagation. As the number of *Sensors* and *Actuators* increase, the chance of the sound event server losing its periodic deadline becomes a threat.

To minimize this problem, a polynomial interpolation method combined with the Newton-Raphson method was used. This approach finds the precise values of $d(t)$ for the first and last sample of each sound frame, and for as many points in between as necessary according to the polynomial degree chosen. Then interpolation using Neville's algorithm is used to obtain $d(t)$ for the remaining samples. Experimental tests with a frame size of 100 ms showed that quadratic interpolation (3 points per frame) provide values of $d(t)$ within less than 10^{-5} seconds of their correct values, corresponding to subsample accuracy, even when *Sensors* and *Actuators* change their accelerations within the considered audio frame. Cubic interpolation (4 points per frame) drives errors down to 10^{-8} seconds or 0.000441 in terms of sample index.

Figure 4 shows performance measurements² made with the framework; these values correspond to the time dedicated to computing the sound propagation between one *Actuator* and N *Sensors*, expressed as a fraction of the frame size. As expected, values grows roughly linearly as a function of $(\#Sensors) * (\#Actuators)$ for each fixed frame-size, until a limit of operability is reached and the computation breaks down, meaning that not every sound produced gets propagated to every *Sensor*. This limit of operability (indicated in the figure by small boxes) increases with framesize, and for the particular equipment used in this experiment, frame sizes between 100 ms and 250 ms seem to offer a reasonable tradeoff between latency and stability for $(\#Sensors) * (\#Actuators) \leq 40$.

It should be noted that this approach do corresponds to a realistic sound propagation simulation that includes the

² Test were conducted using a MacBook Pro with a 2.7 GHz Inter Core 2 Duo processor and 4 GB of memory, running Mac OS X 10.6.

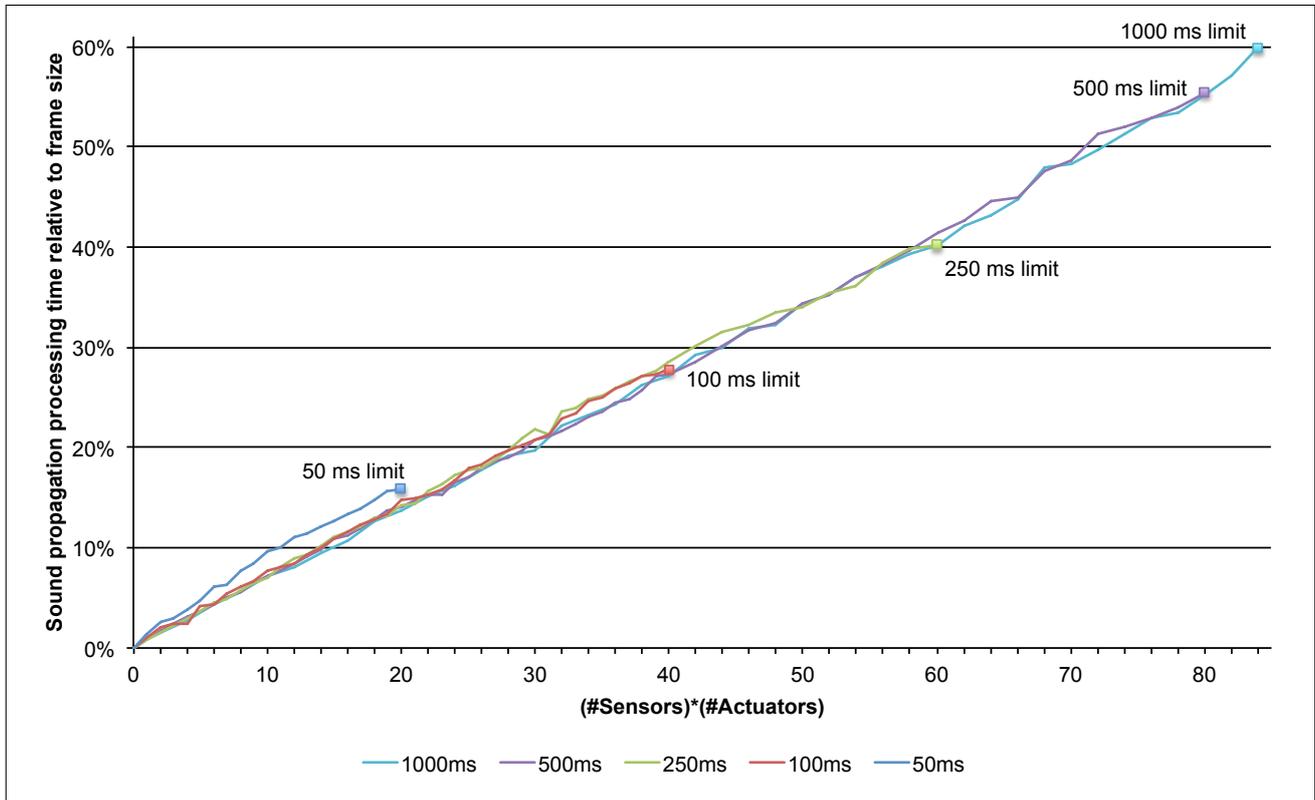


Figure 4. Sound propagation processing time for different frame sizes.

Doppler effect, since values of $d(t)$ depend on continuously changing values of position and speed of the *Sensors* and *Actuators*. Therefore, approaching targets imply compressing wavefronts, and distancing targets imply expanding wavefronts. Frequency shifts are a direct consequence of the definition of $d(t)$ above.

Since a single agent may have multiple sound *Sensors* positioned at different points of its virtual body, a sound wave has different arrival times for each *Sensor*. This allows the use of multi-sensor listener agents that capture 3D soundscapes and send them for external multi-channel playback. It also allows *Reasonings* to use these multiple inputs for source position identification, source separation, etc.

Last but not least, it should be noted that realistic sound propagation is but one of many alternatives in the design of acoustical laws for the virtual world. Some musical multiagent designs [6, 7, 8] rely on different formulations for the virtual space, such as discrete 2D spaces with planar or square wavefronts. Unearthly and bizarre sound propagation schemes could easily be implemented and used for sound experimentation, for instance sending out different frequency bands in different directions with different speeds spreading out from a single sound source.

2.5 Interfacing

Ensemble was designed to allow flexible implementation of musical multiagent applications, with the intention of generalizing from examples found in the literature. Nevertheless, interfacing the framework with other sound-processing and music-processing programs is beneficial for several reasons: it extends the available functionalities of

the framework, it affords code reusability, and it improves user comfort, by allowing part of the application to be developed using a language or environment of the designer's choice.

We will discuss in the sequel several aspects of interfacing with *Ensemble* that we consider fundamental: interfacing with specialized libraries, interfacing with general sound-processing programs, user interfaces and audience interfaces.

Interfacing with external libraries

Two external libraries for audio processing were incorporated into the framework: *aubio*³ and *LibXtract*⁴. Essential functionalities for audio processing such as FFT, digital filters and feature extracting functions can be used transparently when designing agents, reasonings, analysis and synthesis engines through these libraries.

Since these libraries are implemented in C and are platform-dependent, pre-compiled modules were created for the three most common operating systems used by musicians (MS-Windows, MacOS and Linux), which are accessed using Java Native Interface (JNI). An Abstract Factory Design Pattern approach ensures that other libraries can be incorporated in the framework when needed.

Interfacing with other programs

Every agent and component of the system is able to receive and send messages by means of a Router Service. This service, accessible through the *RouterClient* interface, is

³ Available at <http://aubio.org/>.

⁴ Available at <http://libxtract.sourceforge.net/>.

responsible for delivering every message to its correct recipient, using the internal JADE mechanisms for message exchange. The address scheme is based on a string containing the names of the system, agent and component. For example, one can send a message to a sound sensor belonging to a musical agent by directing the message to `/ensemble/pianist/right_ear`, or to an external program, for instance Pd, using the address `/pd`.

Open Sound Control (OSC) [10] is a message-exchange protocol specialized in musical applications, used and understood by major musical softwares. The routing mechanism of the multiagent framework can also be used to send and receive OSC messages through a special *RouterAgent* that implements an OSC server. For example, a Pd external was implemented that works as a graphical representation and user interface for an example 2D virtual world, using only OSC messages. Through it the user is able to see and interact with agents, send messages to start and stop sound processes, and place or move himself/herself within the environment to listen through headphones to a binaural version of the simulation.

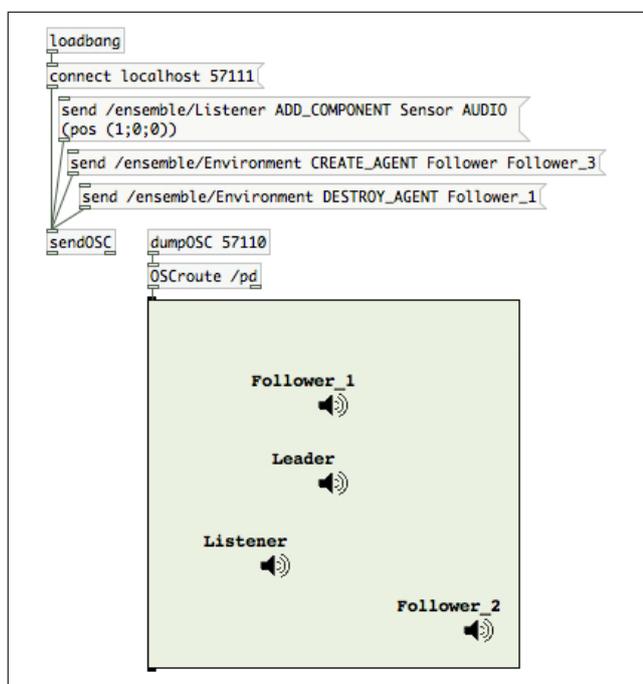


Figure 5. Pd interaction with *Ensemble* using OSC.

Figure 5 shows an example of a Pd patch that interfaces with *Ensemble* by means of the OSC protocol. Two Pd objects⁵ are needed for the communication to take place: *dumpOSC*, used for receiving OSC messages, and *sendOSC*, used for sending messages. These string messages must be parsed, using the *OSCroute* object, and interpreted by the Pd patch according to the user application. We developed a Pd external, *ensemble_gui*, which is a two-dimensional graphical representation of an agent's position on a rectangular virtual world. This external object processes commands sent by the *MovementEventServer* to update the graphical interface. The figure also illustrates

⁵ Bundled with Pd-extended, available at <http://puredata.info>.

some example messages that can be sent to *Ensemble* to create/destroy agents or to add audio sensors, all of which are possible at runtime.

Interfacing with a user or sound designer

Applications using *Ensemble* can be built using a single XML file, which contains parameters defining world components and agent components, as well as starting points for processes and rules for updating every aspect of the system. These XML constructs are meant to be simple indications of the methods, components and parameter values of the framework that will be used in actual simulation, and so they are much simpler than actual Java programming. Whenever an application relies on pre-built agent/world components, the user is able to use the XML file to assemble agents, plug-in their components (sensors, actuators, reasonings, etc), define world parameters and laws, trigger the start of the simulation and also to control the system at runtime using nothing but XML commands.

Extending the system is possible by either programming new components in Java, or alternatively by interfacing with other programs as already discussed. This second alternative is particularly interesting when designing graphical user interfaces for real-time user interaction, which is probably easier to do in Pd than for instance in Java.

Interfacing with real listening spaces

Nothing in this discussion would make sense if there were no channels to peep into or eavesdrop on the virtual environment while simulation is going on. Graphical user interfaces can be used to see the motion of agents as discussed above, while more advanced image processing techniques might also be considered for rendering spatial representations of the virtual world. But getting sound out of the virtual environment is the first and foremost goal when designing a musical multiagent application.

Two *Reasonings* were implemented to provide audio input/output using a regular audio interface hardware, so the user can hear what is happening inside the virtual environment and interact with it by inputting sound. The Jack audio system⁶ was chosen as the audio library for this task for its low-latency, portability and flexibility, allowing a finer control of timing and also the use of multiple channels of the audio interface (both impossible with Sun's current Java Sound implementation). Jack requires the use of JNI, meaning that a library must be compiled for each operating system.

With Jack, it is possible to route audio channels between supported applications (*Ensemble* included) and also to/from an external audio interface. Thus, an external application such as Pd or Ardour can export audio signals that are fed into *Ensemble* through a Musical Agent (using the *JackInputReasoning*), which may then be used as input to a musical Reasoning, or it may be propagated in the virtual environment through a sound *Actuator*. Any Musical Agent within *Ensemble* may likewise export audio signals using the *JackOutputReasoning*.

⁶ Available at <http://www.jackaudio.org/>.

One consequence of the periodic event exchange approach is that a delay of two frames plus the delay of the audio interface itself is introduced when sound is captured and played back in the virtual environment; sound exported from the virtual environment is subject only to the delay of the audio interface.

3. CASE STUDY: CLAPPING MUSIC

In order to test some recent advanced functionalities of the framework, a relatively complex musical application was conceived. The starting point was the musical piece called *Clapping Music*, written in 1972 by Steve Reich. In this minimalist piece, a small rhythmic pattern is repeatedly clapped by two performers. While the first one goes on repeating the exact same pattern, the second one circularly shifts the beginning of the pattern one beat to the left, every 8 (or 12) repetitions, until they are once again synchronized, after 96 (or 144) repetitions. Figure 6 shows the pattern and its first shifted repetition.

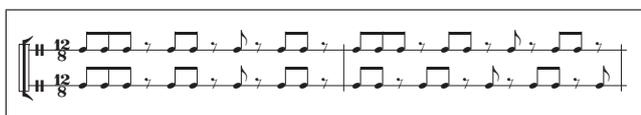


Figure 6. Clapping Music original pattern (first measure), and counterpoint produced with the first shifted pattern (second measure).

In the transposition of the piece to a musical multiagent application, some complicating hypotheses were introduced. First, the rhythmic pattern could be the one proposed by Reich, but it could also be randomly generated if the user so wished. Second, multiple agents (virtual performers) would be involved, with separate entries for each one, and also independent shifting patterns (described by user-controlled parameters). Third, and perhaps most importantly, agents would not be allowed to communicate by any means other than audio, i.e., we excluded the possibility of visual cues or trigger messages that would facilitate synchronization between agents. Everything would have to be done using audio analysis (i.e. finding out what the pattern is, when did it begin, etc.) and audio synthesis.

Three types of agents were implemented: a *Leader* agent who is responsible for proposing the pattern; a *Follower* agent who will try to discover the pattern and then start to play it, shifting each repetition by a certain amount; and a *Listener* agent who copies the output to an external user. At runtime, the *Leader* agent defines a rhythmic pattern (set by the user or randomly created based on the number of desired beats, bpm and wavetable), and start repeating it, accentuating each first beat of the pattern as a cue for other agents to pick up where it starts. Much more complicated pattern-discovering strategies could also be implemented without using such a cue, but they would inevitably prohibit patterns which contain repetitions of smaller sub-patterns or motives, such as AA or AAB patterns, for instance.

The *Follower* agent has two states: analysis and playing; in the analysis state, it must detect onsets as each audio

frame arrives in its sensors, and create a list of timestamps and intensities. Real-time onset detection and other signal processing calculations (like FFT and RMS) used the *aubio* library. As soon as the second repetition starts, the agent has the pattern and is ready to play, but since it can only produce output for the next audio frame, it will wait for the third repetition of the pattern to enter with its shifted version.

Agent movements were defined to graphically illustrate the amount of shifted beats of each *Follower* with respect to the *Leader*. While the *Leader* stays at the center, the *Followers* perform an orbital-like motion around it based on their current shift value. Whenever a *Follower* starts a new repetition with a new shift value, it will walk towards a new orbital position, and all agents align when a cycle is completed (this may take much longer than the 144 repetitions of the original piece). Agent positions also influence how the user (through the *Listener* agent) will hear the piece, since sound propagation takes all agent positions into account, with corresponding delays, attenuation and filtering.

The flexibility in the application setup invites experimentation with different parameters values, including the amount and periodicity of shifting and the pattern itself, which brings about a myriad of intricate interweaving patterns. A curious effect due to spatialization is that followers far away from the leader will never play in perfect synchronism, since sound waves take some time to propagate, implying that the pattern is recognized with a certain delay.

An important fact about this application is that it uses mainly built-in general purpose *Ensemble* components. An XML configuration file is used to assemble all components and set up the simulation, and only one customized Reasoning had to be implemented in Java. Both files account for less than 500 lines of code, including comments.

4. CONCLUSION

This paper presented an updated account on the development of *Ensemble*, a framework for musical multiagent systems. Much has been improved since the first implementation presented in [1]: the current framework is more flexible, allowing integration with external libraries and programs, more user-friendly, allowing the specification of applications without Java programming and also the use of external graphical user interfaces, and also shows an increased level of realism and better performance in the sound propagation simulation, due to physical simulation and major redesign in the internal data structures, such as agent *Memories* and virtual world *Laws*.

The choice of Java, motivated by the fact that it is a general-purpose, platform-independent language with wide availability and support, has also had its drawbacks, some of them predicted and others not so much.

Predictably, Sun's Java Sound API implementation is not suitable for a more demanding audio application. For instance, depending on the operating system and audio interface driver implementation, one cannot address a specific audio channel of an external audio interface. This diffi-

culty has been overcome through the use of PortAudio, which, although it requires pre-compiled modules to work on every platform, guarantees full access to most sound peripherals.

Java's garbage collection mechanism can sometimes interrupt important time-constrained operations of the framework, such as the periodic event exchange, or important audio processing operations, such as the sound propagation simulation. Since one cannot control when the garbage collector will be called, the framework is bound to lose some audio frames whenever the system becomes overloaded. Using Java Real Time might be a way to solve this problem, since processing start times and deadlines could be enforced by a real time operating system.

Also, the first few runs of each method were observed to be slower than subsequent runs, since the Java Interpreter always tries to execute code without compiling it, and only decides to natively compile some code excerpt after it detects intensive repetition. This problem was circumvented by creating a warm-up repetition routine for computer intensive methods.

The object-oriented approach used in the modeling and implementation of the architecture implied a great deal of object creation and destruction, on several levels of abstraction. These operations are somewhat expensive for the Java Virtual Machine, since memory need to be allocated and constructors/destructors called. Some time-constrained methods which deal with a lot of data, like the sound propagation simulation, are heavily hit by this fact. In order to increase performance, some coding techniques not much in line with the object-oriented paradigm were applied, like reusing the same object and minimizing the number of calls to a method.

Ensemble, in its current version, may be used to reproduce many kinds of musical multiagent applications, such as those discussed in [1]. Some performance improvements, relative to memory usage and synchronism between state machines, are already scheduled for implementation. These improvements are expected to allow the framework to work with lower latencies and an even larger number of agents.

The framework code, as well as example applications, is open-source and freely available on the web⁷. There is also a step-by-step tutorial on how to build a simple application with existing components. Documentation is expected to significantly improve in the near future.

Acknowledgments

This work has been funded by CAPES, CNPq and FAPESP (grant 2008/08632-8).

5. REFERENCES

- [1] L. Thomaz and M. Queiroz, "A framework for musical multiagent systems," in *Proc. Int. Conf. Sound and Music Computing*, Porto, 2009, pp. 213–218.
- [2] D. Bisig, M. Neukom, and J. Flury, "Interactive swarm orchestra—a generic programming environment for swarm based computer music," in *Proceedings of the International Computer Music Conference*. Belfast, Ireland, 2008.
- [3] M. Spicer, "AALIVENET: an agent based distributed interactive composition environment," in *International Computer Music Conference*, 2004, pp. 1–6.
- [4] D. Murray-Rust, A. Smaill, and M. Edwards, "MAMA: An architecture for interactive musical agents," in *Proceeding of the 2006 conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29–September 1, 2006, Riva del Garda, Italy*. IOS Press, 2006, pp. 36–40.
- [5] G. L. Ramalho, P. Y. Rolland, and J. G. Ganascia, "An artificially intelligent jazz performer," *Journal of New Music Research*, vol. 28, no. 2, pp. 105–129, 1999.
- [6] P. Dahlstedt and M. Nordahl, "Living melodies: Coevolution of sonic communication," *Leonardo*, vol. 34, no. 3, pp. 243–248, 2001.
- [7] K. McAlpine, E. Miranda, and S. Hoggar, "Making music with algorithms: A case-study system," *Computer Music Journal*, vol. 23, no. 2, pp. 19–30, 1999.
- [8] J. McCormack, "Eden: An evolutionary sonic ecosystem," *Advances in Artificial Life*, pp. 133–142, 2001.
- [9] M. Gimenes, E. Miranda, and C. Johnson, "The development of musical styles in a society of software agents," in *Proceedings of the International Conference on Music Perception and Cognition*, 2006.
- [10] M. Wright and A. Freed, "Open sound control: A new protocol for communicating with sound synthesizers," in *Proceedings of the 1997 International Computer Music Conference*, 1997, pp. 101–104.
- [11] R. Johnson and B. Foote, "Designing reusable classes," *Journal of object-oriented programming*, vol. 1, no. 2, pp. 22–35, 1988.

⁷ Available at <http://code.google.com/p/musicalagents/>.

AUDIO PHYSICAL COMPUTING

Andrea Valle

CIRMA - Università di Torino

andrea.valle@unito.it

ABSTRACT

The paper describes an approach to the control of electromechanical devices for musical purposes (mainly, DC motors and solenoids) using audio signals. The proposed approach can be named “audio physical computing”, i.e. physical computing oriented towards sound generation by means of audio signals. The approach has its origin in a previous physical computing project dedicated to music generation, the Rumentarium Project, that used microcontrollers as the main computing hardware interface. First, some general aspect of physical computing are discussed and the Rumentarium project is introduced. Then, a reconsideration of the technical setup of the Rumentarium is developed, and the audio physical computing approach is considered as a possible replacement for microcontrollers. Finally, a music work is described, in order to provide a real life example of audio physical computing.

1. INTRODUCTION: PHYSICAL COMPUTING AND SOUND

The use of computers in music is typically associated respectively to algorithmic composition, that is, computational approaches to the organization of musical form, and to sound generation, that is, generation of digital audio signals as the sonic output of a computer-based musical system. Rather, a less evident possibility concerns the use of a computer for the generation of acoustic sounds, in order to regain a specific acoustic physicality in output. With respect to such a goal, a still new but now firmly established perspective is opened by “physical computing”, the terms meaning “computation with physical objects” [1], [2]. Physical computing fosters the idea that computation can be taken outside standard computers and embedded into physical objects. The key element for developing physical computing are microcontrollers, that is, computation units packed in small-sized circuit boards, with I/O facilities allowing to connect sensors and actuators. Musicians are considered to be the first to practice physical computing [1]. In particular, a long even if often underground experimental tradition of analog electronic music has worked intensively on hardware hacking with the aim of coupling control information from electric signals, performer gestures and physical output [3]. In the digi-

tal reprise of such a perspective, microcontrollers can play a pivotal role. In particular, as microcontrollers can drive electromechanical devices, physical objects can be involved as final, acoustic sources of sound generation processes. In this way, it is possible to create “acoustic computer music”, that is, a music entirely controlled by computational means, but in which sounds are generated from acoustic bodies¹.

2. THE RUMENTARIUM PROJECT

The design and realization of the Rumentarium project [5] move from these assumptions. The Rumentarium is a set of handmade percussive instruments (“sound bodies”), made of heterogeneous resonators that are acoustically excited by DC motors. The motors are controlled via computer through microcontrollers. In the Rumentarium, the design and production of sound bodies is inspired by sustainable design [6]. The name “Rumentarium” originates from *rumenta*, in Northern Italian meaning “rubbish, junk”. The design embraces an ecological perspective based on the reuse of common objects. Many practices around the world have traditionally developed specific attitudes towards the “refabrication” of objects as a normal way of shaping and reshaping the semiotic status of material culture [7]. According to refabrication, in the Rumentarium the DC motors are scavenged from discarded electronics (CD and DVD players, mobile phones, toys) and they can be extended by adding parts of various materials (plastic, wood, metal), thus implementing different modes of excitation (percussion/friction). Resonators are assembled from a huge variety of recycled/reused materials: e.g. pipe tobacco boxes, glass bowls, broken cymbals, kitchen pans. The resulting sound bodies are generally assembled via metal wires, glue, soldering and they can include Lego parts. Figure 1 shows a version of Rumentarium including 24 sound bodies, installed at Share Festival, Torino, 2009. Rumentarium couples the ecologic perspective with an investigation into digital control of physical objects, as the sound bodies are entirely computationally controlled. Very naturally, it turned to microcontrollers as the hardware interface between the computer and the motors. In particular, Arduino Diecimila and Duemilanove boards were used, as Arduino [8], being inexpensive, open source, easy to program, is a *de facto* standard in physical computing. In Figure 1, Arduinos are contained in the plastic boxes with loudspeaker connectors. One of the main ideas in Rumentarium is to use a high-level software as a control interface. The whole

¹ An analogous perspective on the digital control of acoustic sound production is in [4].

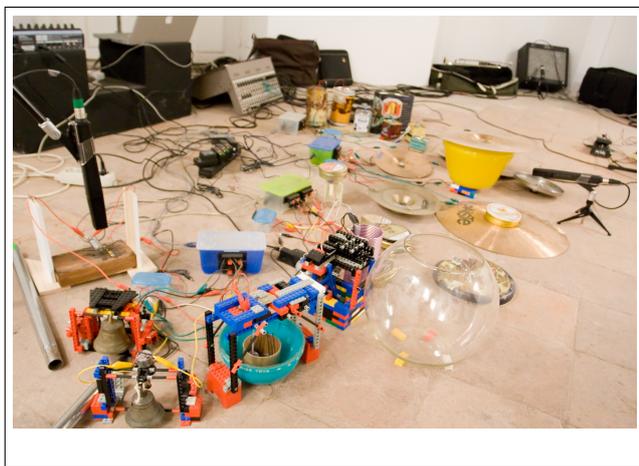


Figure 1. Rumentarium installed at Share Festival, Torino, 2009.

software interface is written in the SuperCollider language [9], that summarizes features common to other general and audio-specific programming languages (e.g. respectively Smalltalk and Csound), and at the same time allows to generate programmatically complex GUIs. While a general programming language (e.g. Java) could have been chosen for controlling the boards, SuperCollider offers native audio DSP capabilities, useful for audio/musical application: as an example, in this way, through the analysis of audio signals performed by SuperCollider, Rumentarium can be seamlessly programmed to react to external sound sources (e.g. to speech). In addition, MIDI protocol is natively supported, as it is of common usage for musical devices (e.g. control surfaces). On the top left of Figure 1, a MIDI controller and the main computer to which is connected are visible. Rumentarium has been used extensively live, both as an autonomous sound installation and as a performing instrument (Palermo, Spazio Orioles, September 2009; Torino, Festival Share, December 2009; Milano, Festival Audiovisiva, May 2010; Roma, Riunione di Condominio, June 2010). It can be heard on AMP2's *Hopeful Monster* album, in the *Musica Improvisata* box set by Die Schachtel [10], and two other albums are in press.

3. CONTROL PIPELINE: FEATURES AND ISSUES

While the computational control is a major (and aesthetically satisfying) feature in the Rumentarium, yet the overall technological pipeline has proven to be in some cases unstable and hard to debug, because of the many layers involved. Figure 2 shows the general structure of the Rumentarium, including both hardware and software elements (the latter specifically developed in SuperCollider for the project).

On each Arduino (2), the ports for PWM (pulse width modulation, where motors are attached) are indexed 3, 5, 6, 9, 10, 11, and the range of available values, to be converted in voltage, is [0, 255]. Each port is connected to the DC motor of a sound body through a Darlington transistor (following the basic design by [1]) that delivers current to a motor (through a power hub). The main software component is the RuMaster application (1), that acts as the soft-

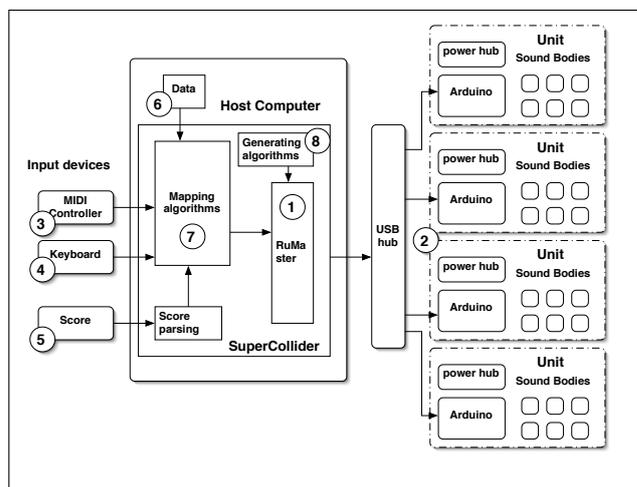


Figure 2. Rumentarium hardware/software setup.

ware layer toward the physical output (microcontrollers and sound bodies). A RuMaster instance allows to treat a set of e.g. four different Arduino microcontrollers (2) as a unique abstract device with 24 abstract ports indexed from 0 to 23, to be sent values in the range [0, 1]. On this abstract software layer it is possible to easily build mapping strategies [5]. Such strategies can be related to data coming in real time from external inputs, such as MIDI gestural controllers (3) or from the keyboard (4). Data can also be gathered in non real-time from external sources such as a handmade graphical score (5), or from digital information (6). In all these cases, mapping algorithms (7) are needed to specify a semantics in terms of Rumentarium's behavior. It is indeed also possible to directly generate control data for the RuMaster (8). While Figure 2 offers a high-level, hence blurred, view of the communication protocol, Figure 3 shows the information flow and the traversed hardware/software layers, both on computer (I) and microcontroller (II). The presented software configuration dates back to 2008-09, and the situation has partially changed since then. Still, this multilayered software configuration can be at the origin of different issues, related to different layers, as each software layer depends on different third-party developers (or communities of developers), and the integration of the layers is exclusively a feature of the final project. First of all, in order to be accessed from the operating system, Arduino boards needed platform-specific drivers (3). Low level software components, as the ones required to interface with serial port, are tightly dependent on operating system: an OS update can easily break their functionality. The driver issue is a complex one, and not by chance the new Arduino Uno² tries to solve it by presenting the operating system a generic HID interface. Then, in order to drive the Arduino boards, SimpleMessageSystem (SMS) was used, a third-party library that forces Arduino to listen to the USB port, where it can receive instructions from the host computer (4). The library is now obsolete and the actual way to communicate in real-time with Arduino controllers through the USB port is no more

² <http://arduino.cc/en/Main/ArduinoBoardUno>

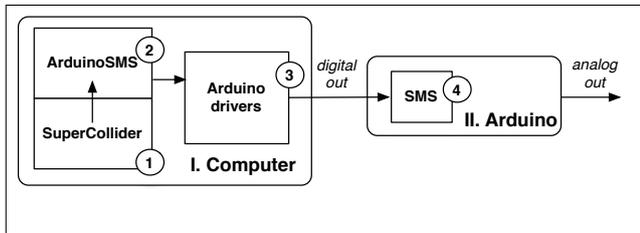


Figure 3. Hardware and software layers in the Rumentarium project.

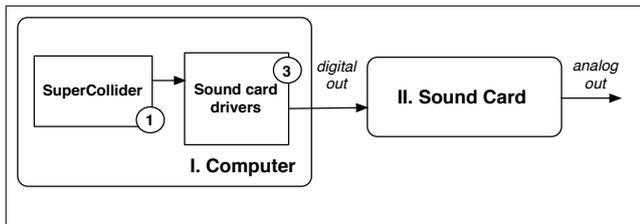


Figure 4. A typical hardware/software audio setup.

the SimpleMessageSystem library but the CmdMessenger one (that in the meantime has superseded an intermediate step, the Messenger one³). Finally, SuperCollider offered a third-party class, ArduinoSMS (2), as the interface –on the the SuperCollider side (1)– to the SMS library –to be loaded on the Arduino side. As evident from the previous discussion, even in a short timespan, such a situation has brought up relevant software compatibility issues. Software updates are not synchronized among layers and they can happen at different development rates. As an example, being SMS now almost obsolete, then the ArduinoSMS component on the SuperCollider side should be replaced with a new one. While this situation is indeed unavoidable in complex hardware/software systems, where maintenance routine is a major task, nonetheless it is quite complex to be handled by the composer/performer, even if s/he is acquainted to computer programming. Other issues are not related to software but depend on hardware. Instability in powering can lead to malfunction in the board. Data transfer rate and management on the board can be problematic: in the Rumentarium version involving up to 24 different sound bodies through 4 Arduinos, the number of control messages per second sent through the USB ports easily grows to a value that microcontrollers seemed not to be able to cope with. This became more apparent while using a MIDI controller, e.g. while turning a knob: in that case, a fast series of MIDI messages fires, each of them triggering a message to the controllers, easily getting beyond the maximum allowed.

4. AUDIO PHYSICAL COMPUTING: AN APPROACH

In order to solve, at least partially, all these issues, a possible new approach has been developed. The main mission for microcontrollers is embedding computation into autonomous physical objects. Besides, they provide in-

³ See in general <http://arduino.cc/playground/Code/>

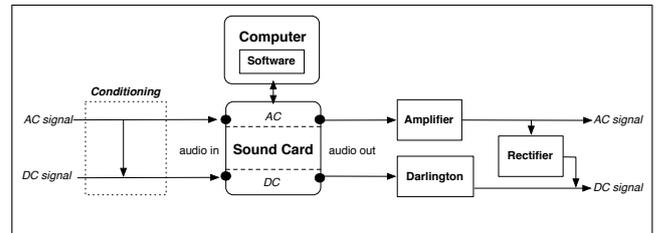


Figure 5. Audio physical computing chain.

put/output conversion from/to analog domain. As an example, in Rumentarium the microcontrollers simply act as digital-to-analog converters (DACs). Taking exclusively into account the conversion task, both in input and output, standard sound cards perform intuitively much better than a microcontroller. From the hardware point of view, sound cards are designed to receive in input and generate in output electric audio signals with minimized distortion. At least starting from the prosumer level, they are interfaced to operating system through reliable drivers and they can easily manage many parallel channels of audio⁴. Apart from sound card drivers, the only other software layer in such a configuration is audio software: available ones are many, and can fit various user’s needs and approaches. As an example, SuperCollider is indeed a highly specialized software for audio synthesis and processing. Thus, comparing Figure 3 with Figure 4, layer 3 is substantially no more an issue, and layers 2 and 4 can be eliminated. While sound cards are faster and more reliable if compared to microcontrollers, they are indeed much more expensive, up to a factor of 10 for analog in/out. As noted by O’Sullivan and Igoe: “Musicians are the pioneers of physical computing. They have solved lots of problems of getting physical gestures into electronic form”, but, “typically, their solutions are high-level and expensive” ([1], p. 354). While this holds true in a general physical computing context, sound cards are basic hardware tools for a musician, so they do not require an extra investment. On the other hand, while microcontrollers (or, better, the serial interfaces on which they are mounted) generally have a very limited bandwidth, the use of multichannel audio can really improve the input/output performance, as signals are efficiently computed at audio sample rate. An overall setup for input/output of audio signals to be used as control signals in a physical computing scenario is depicted in Figure 5. Such a configuration allows to implement “audio physical computing”, the term referring to physical computing based on audio signals for sensing and controlling physical objects. Figure 5 must be considered as a first draft of manifold possible solutions, to be fine-tuned empirically. A first distinction must be made between AC-coupled and DC-coupled sound cards. The first ones accept only AC signals in input and output, that is, standard audio signals. Most low-level sound cards are AC-coupled. DC-coupled sound cards are able to handle

⁴ Indeed, this does not guarantee that interface problems are automatically solved by using an audio card: audio drivers can require some kind of optimization (e.g. setting the optimum buffer size, sampling rate, etc.) in order to maximize their performance on a particular system and hardware installation can be a complex issue to manage.

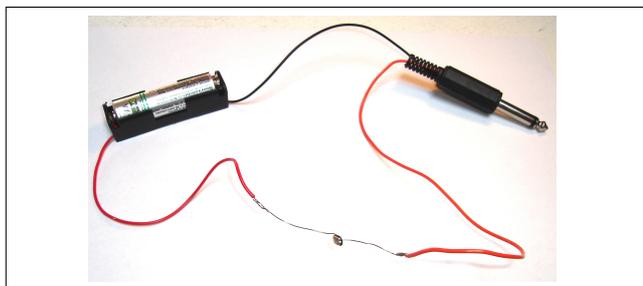


Figure 6. A photocell sensor as audio input.

not only AC signals, but also DC signals both in input and output, and thus represent a higher level subset (from the prosumer level) of AC-coupled ones. Here we just sketch some ideas on the input side, while focusing later on the output side, in order to provide a possible replacement for Rumentarium setup. A plethora of sensors can be connected to the sound card⁵, provided a different kind and degree of “signal conditioning”, that is, an analog manipulation stage on the signal so that it meets the requirements of analog-to-digital conversion [12] [13]. As conditioning strategies depend on each sensor’s features, here we just sketch some possible approaches. By the way, most solutions in [1] can be directly used or easily adapted to work with sound cards. For what concerns input signals, indeed microphones (including piezo contact ones) do not require conditioning, as they can be directly connected to AC ins. Some sensors (e.g. typically accelerometers) are able to generate DC signals that have enough voltage to be connected directly to a DC input: in negative case they must be previously amplified. Passive sensors can be fed into a DC input after connecting them to a voltage source. Figure 6 shows a photo cell acting as a passive resistance between a power source of 1.5 volt battery and the sound card’s DC in (here to be connected by means of a standard male 1/4 inch jack plug). The sound card’s phantom powered input can be useful in order to increase signal gain. It is also possible, even if not straightforward, to connect a DC signal to the AC input of the sound card by using an inverter that converts DC to AC. An inverter can be implemented by means of a Voltage Control Oscillator (VCO) based on a function generator integrated circuit (IC), that is, a special-purpose oscillator used to produce sine, square, or triangle waveforms (such as the NE555). The signal frequency/amplitude of an oscillator can be varied by the external sensor. The resulting (frequency modulated) AC signal can be connected to the AC input of the sound card. A pitch/amplitude tracking software algorithm running on the host computer then allows to reconstruct the modulating DC signal quantity from the carrier AC signal. Indeed, once digitized, the signal can be processed (e.g. smoothed, sampled, filtered, analyzed etc) by means of usual, well-known DSP techniques.

On the output side, analog signals can be used as control signals for various applications. DC-coupled sound cards are able to directly output analog DC signals. These sig-

⁵ The process can be termed “analog audio sensing”, as suggested by [11].

nals can be compared to voltage provided by Arduino outs and used similarly. As an example, they can be used to feed the Darlington transistor that has been previously discussed in relation to motor driving in the Rumentarium setup. An analogous technique is used in the Volta software by MOTU, a virtual instrument plug-in that turns the sound card into a voltage control interface⁶. Volta generates and sends DC signals via the sound card to voltage-controlled analog synths, thus allowing the user to reach a digital control over analog hardware. In turn, the audio signal output by the analog device can be connected in a feedback loop to the sound card input and received by Volta. Through this feedback loop, Volta is capable of advanced features such as autocalibration on analog synths. Volta is indeed a “hybrid control system” in which the computer generates digital control functions to be converted into voltages feeding the control input of synthesizer modules [14]. Thus, audio physical computing can be thought as a generalized hybrid system strategy.

A second option is based on AC signals (i.e. usual audio signals). In this case, an audio amplifier is needed after the sound card in order to ensure enough power, usually in order to drive loudspeakers. While this means a further increase in terms of invested money, a multichannel power amplifier is again a very general component in an audio/music studio setup, that can be literally used for decades. Furthermore, inexpensive stereo amplifiers are available in assembly kit from electronic resellers. The resulting output configuration, made of computer, sound card, amplifier, is indeed very stable and can be implemented without depending on specific hardware models or technologies. Moreover, in the context of physical computing, loudspeakers can be modified in order to exploit them as sources of mechanical force [3], e.g. to make vibrate different surfaces. Loudspeakers are a straightforward means to convert audio information into mechanical force. In the perspective of a generalized audio physical computing framework, an easy and effective solution in order to convert AC into DC signals is to add a rectifier as a further element to the chain, after the amplifier. The rectifier converts alternating to direct current by mirroring the negative part of the electric signal in the positive domain (as in the absolute value in the numerical domain). It can be made up of a single diode component with two inlet (from AC) and two outlets (to DC), thus resulting very easy to build (much more than the already minimal Darlington design or than the VCO circuit mentioned above). Figure 7 shows an 8-input (that is, audio channels) DIY rectifier. An amplified and rectified signal can be used to directly feed DC motors. In general, an amplifier is mandatory when using motors, as it not only provides the current required to drive them, but, also, protects the sound card over inductive back current loads that can damage it.

To summarize, the simplest solution among the possible audio physical computing configurations represented in Figure 5 involves a DC-coupled sound card: in this way, DC signals (as typical output by sensors) can be directly used in input. AC output is simpler to handle than DC, as it does not

⁶ <http://www.motu.com/products/software/volta/>

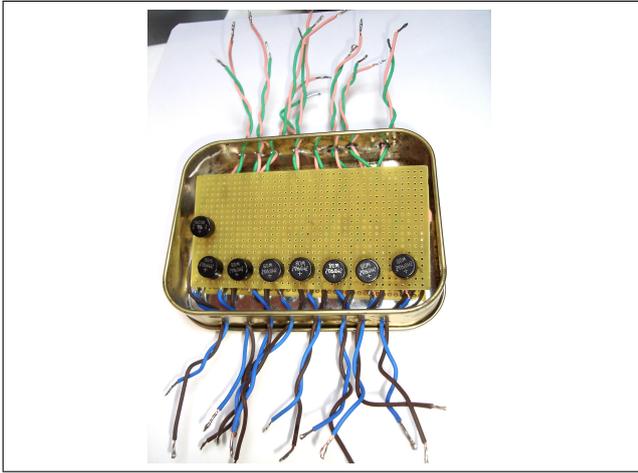


Figure 7. A DIY 8-input rectifier, AC (bottom) to DC (top).

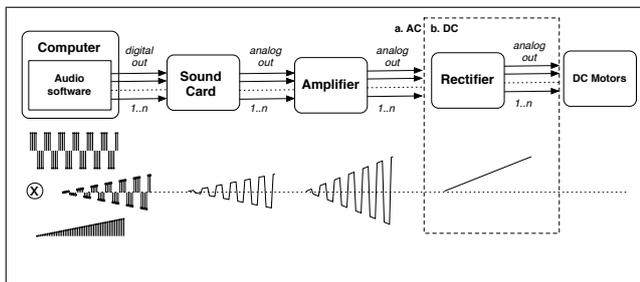


Figure 8. Audio physical computing scenario for DC motor driving: a. alternate current, b. direct current.

require further complex circuitry, it allows to include loudspeakers in the setup, to be used to deliver audio or to drive objects, and can be easily rectified, thus providing also DC signals if needed. Moreover, by inserting or removing the rectifier for some signals it is possible to define a mixed AC/DC scenario. The output part of this subchain will be described in the next section.

5. RE-ENGINEERING RUMENTARIUM IN THE LIGHT OF AUDIO PHYSICAL COMPUTING

By means of the previously discussed approach, it becomes possible to re-engineer the Rumentarium setup in order to replace microcontrollers with an audio chain. A different case, involving solenoids, will be described in the next section. As we have seen, the sound card-amplifier-(rectifier) subchain, even if starting from AC audio signals, allows to use both AC and DC signals. This possibility opens the way to two scenarios (Figure 8). The first one makes use of audio signals directly to feed the electromechanical devices, that is, it does not make use of the rectifier (Figure 8a). Collins has already proposed to use audio signals in order to drive electromechanical devices, in particular little DC motors and vibrators for cell phones and pagers [3]. He suggested to use the motors as audio drivers, by clamping their body directly to the object, thus transmitting the vibration. That is, the motors act as final amplification stages for audio signals. In the Rumentarium,

DC motors are instead used at a control rate, to excite a resonator. In Figure 8 a waveform is first digitally generated via software, then converted by the sound card and finally amplified. In case of DC motors, the alternating voltage sign in the AC signal does not impede working, but simply determines a corresponding inversion in rotation's direction (that is, a negative rotation speed). For each zero crossing in the signal there is an inversion in the motor's rotation direction, where the inversion rate is determined by the signal frequency. Rotation inversion was not possible with the previously discussed microcontroller setup, while it is largely useful in the Rumentarium context, as it allows to implement a backward mechanism for motor-driven beaters. The waveform is relevant too, as it determines the way the voltage is delivered to the motor.

A square wave results in a abrupt change of rotation direction while maintaining the same rotation speed. The duty cycle determines the duration ratio between opposite sign rotation phases. In Figure 8a (bottom), if a DC motor is connected to the output, the ramped square determines a constant oscillation between increasing opposite rotation speeds. Instead, a saw waveform results in a ramp signal that linearly decreases from the maximum rotation speed in one direction, to no motion (while crossing zero), to the maximum rotation speed in the opposite direction; then, it jumps abruptly to the initial maximum speed and rotation. A triangle waveform continuously increases/decreases the motor's speed while changing rotation direction two times per cycle. Even if the real, final behavior depends indeed on many mechanical constraints, varying on per sound bodies basis, nonetheless it can be grossly determined by the specific waveform in the digital signals. In this sense, a greater complexity can be achieved than with microcontrollers. In Rumentarium, the original design involving Arduinos allows only to vary the amount of direct current fed into the motors by a power supply: thus the only parameter that could be controlled was rotation speed. The control depended exclusively on the PWM modulation provided by the microcontrollers, as a result of the digital-to-analog conversion. The real transfer function for conversion is not known, and indeed a greater approximation is tolerated with microcontrollers than with sound cards. In short, in the audio physical computing scenario, algorithmic strategies for the generation of digital signals (that are straightforward processes in the electronic music domain) allow for a higher degree of control.

While still in the first scenario, it is indeed possible to use unipolar AC signals (i.e. only positive): an unipolar signal, varying only in the positive domain (the contrary would be the same in the opposite direction) will result in a speed rotation varying from null to the maximum, with no direction changes. In any case, if the aim is the recreation of a direct current, i.e. as in the original microcontroller setup, the most effective strategy is to use DC signals. This is what happens in the second scenario, where the rectifier intervenes. The ratio for using the rectifier –instead of directly connecting a DC output from the sound card into the amplifier– is to be found in the presence of the amplifier. From a computational perspective, the generation

of a DC-like signal is not an issue per se. However, even if DC-coupled sound cards are in use, the generation of DC signal can be prevented by the amplifier: designed to work in the analog audio domain, where direct current is considered a problem as it typically results from electric interferences (and can damage loudspeakers), the amplifier circuitry typically operates an automated correction step on the final signal, precisely in order to remove the (supposed) DC offset. This correction typically leads to output a silent signal. But a DC signal can be obtained by rectifying the AC signal from the amplifier. In the perspective of rectification, while different waveforms can be indeed used, the general solution is represented by a pulse waveform, oscillating between maximum values. In Figure 8b (bottom), the digital pulse wave signal is rectified after amplification: the signal resulting from rectification is a straight direct current. In this way, by varying the amplitude of the digitally-generated square wave, it is possible to continuously control motor speed, as the amount of DC current in output is varied proportionally to the amplitude of the square wave itself. In short, the amplitude envelope modulates the square wave that is, in turn, demodulated by the rectifier, so that the same envelope, once amplified, is returned in output. In this way, the same control architecture of the original microcontroller setup can be achieved.

6. CIFRE DEL COLPO: SOLENOIDS

In this section, audio physical computing as discussed before is applied to the control of solenoids. By discussing a music work, *Cifre del colpo*, it will be possible to introduce some aesthetic issues at the basis of the approach. *Cifre del colpo* (“Numbers/Ciphers of the beat”) is written for 8 percussions whose sound is captured through microphones and expanded by a set of other objects, activated by solenoids. These “Expanders” are intended as a landscape of accidental sound bodies, following the Rumentarium model. But while in the Rumentarium sound production is entirely electro-mechanical, in the *Cifre* sound generation is divided between a human player and a mechanical one.

An overview of the setup showing both elements and their placement, is depicted in Figure 9. The setup includes Percussions, Microphones, Sound card, Computer, Amplifier, Expanders. Percussions are indicated with capital roman numerals (I–VIII) and must be chosen following the criterium of sharing a clear common feature. Microphones are to be placed around Percussions, in order to uniformly sample the set. Figure 9 shows four microphones, but their number can vary. In short, the microphones are listening to the soundscape created by the percussion player and reply by expanding it through the sound bodies activated by the solenoids. The number of microphones determines the number of channels, indicated in Figure 9 with numbers (1–4). Each microphone is connected to a discrete input channel of the Sound card. In turn, the Sound card is connected to a computer analyzing the input signals coming from the microphones, and generating control signals for the Expanders. The output pipeline is the one described in Figure 8, in relation to the

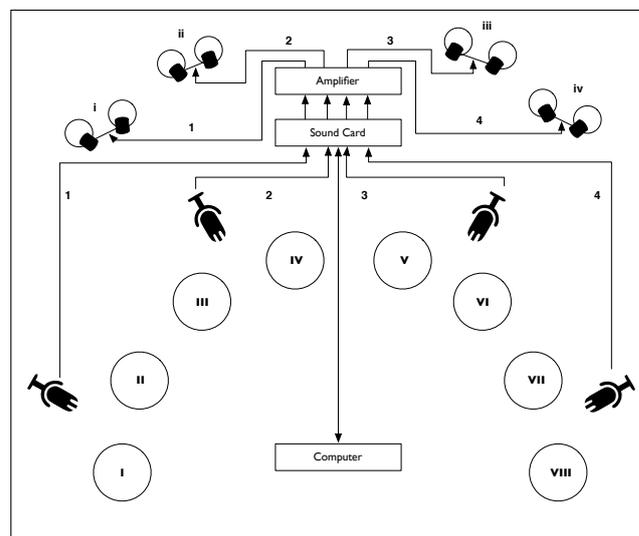


Figure 9. *Cifre del colpo*, setup.

first AC scenario. The sound card outputs another array of discrete channels, their number matching the number of input channels (hence, Figure 9 includes 4 output channels). Output channels are amplified by the Amplifier, so that the resulting audio signals can feed the solenoids in the Expanders. Expanders are indicated with lowercase roman numerals (i–iv). Sound bodies are grouped in Expander sets, and all the Expanders belonging to the same Expander set receives the same audio channel. Solenoids (and motors) are at the core of many projects involving musical robots. As an example, ModBots are digitally controlled acoustic percussive robots developed by Bill Bowen (2002), in collaboration with Lemur and installed at Angle Orensanz Foundation in 2002⁷. ModBots are miniature, modular instruments designed with an emphasis on simplicity, and making use of only one electromechanical actuator (a rotary motor, or linear solenoid)⁸. While Bowen’s Modbots are used mainly as components of sound installations, Lemur is the main technological provider of Pat Metheny recent Orchestrion project, where solenoid actuators are used interactively by the renowned guitar player in real time⁹. The ModBots project has been at the origin also of William Brent’s Ludbots (2008): LudBots have been used as instruments in live performances, but also as components of the “False Ruminations” installation¹⁰. All the previous projects use microcontrollers to drive the motors/solenoids. Coherently with the ecologic and low-profile assumption at the basis of the Rumentarium aesthetics, the solenoids intended to be used in the *Cifre* project are coil components used in hydraulic valve systems coupled with a simple bolt, that acts as an iron shaft to be pulled/pushed when the coil is given current. As electro-mechanical noise (both in acoustic and behavioral sense) is a key concept in the work, they are not intended to provide

⁷ <http://lemurbots.org/videoandaudio.html>

⁸ <http://public.bilbowen.net/home/digitally-controlled-acoustic-percussion>

⁹ <http://patmetheny.com/orchestrioninfo/qa.cfm>

¹⁰ <http://williambrent.conflations.com/pages/projects.html>



Figure 10. Testing a solenoid on a snare drum.

a precise output, rather to show complex and partly unforeseen results. They must be applied to Expanders following different strategies depending on empirical fine tuning. Figure 10 shows a test where a solenoid is suspended over a snare drum thanks to a clip microphone holder. When the coil is magnetized, the bolt is raised above the snare skin (the solenoid is connected by two alligator clips to the amplifier visible back on the left). Then, when magnetization is over, it falls on the skin generating sound. In order to obtain a random bouncing effect, a nut is placed between the skin and the bolt. The overall signal flow for the *Cifre* is shown in Figure 11. The process applies independently to each signal from each microphone (that is, in case of e.g. 4 microphones, there would be 4 parallel processes like the one in Figure 11). The audio physical computing scenario is the following. An onset detection algorithm is running on the computer. When an onset is detected in the input signal is , a pulse train t is digitally generated. The pulse train contains an unipolar square waveform of 0.25 seconds. The pulse train is converted, amplified and sent to the solenoid, so that it flips abruptly between no motion (amplitude is 0, and applied voltage is null) and maximum motion (where signal amplitude is at its maximum too). The pulse train waveform must have a frequency in the range $[1.0, 10.0]$, to be chosen by the interpreter. The lowest value results in a single beat, the highest in a tremolo-like effect. The amplitude of t is determined by amplitude tracking of input signal is , scaled by function f_{scale} , that truncates amplitude intensity (expressed in dB) in the range $[-30, 0]$, and linearly scales the result in the range $[0.1, 1]$, to be used as a multiplier for t amplitude. All these processes, integrating audio analysis on input signals from microphones and signal generation for the solenoids, are easily handled by a software application named *Guardian*, that has been specifically developed in SuperCollider for the work. Figure 12 shows the GUI for a 4-channel I/O setup, where GUI elements are the replicated for each channel. From top to bottom, GUI shows: a scoping window showing the input signal for the channel (channels 1 and 2 are receiving each an input sig-

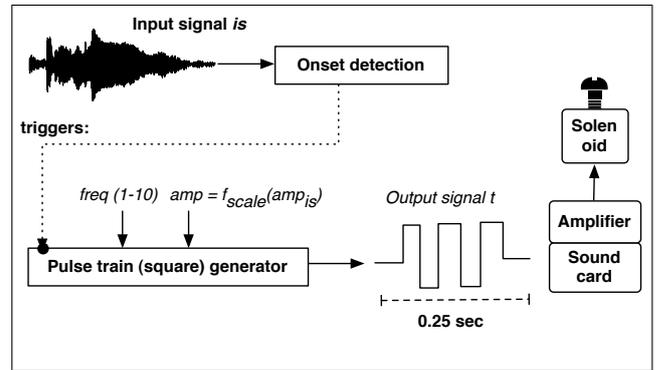


Figure 11. Onset detection, triggering and signal generation.



Figure 12. GUI for the Guardian application.

nal); four knobs/number boxes respectively for input signal volume, for onset detection threshold, for frequency of output square signal (0-10) and for output signal volume, a scoping window showing the output signal for the channel (channels 1 and 2 are outputting control signals as a response to onset detection on respective input). The application is not intended as an official version, but just as one of the possible implementations with respect to the description of the algorithm provided by the score.

7. CONCLUSIONS AND FUTURE WORK

Audio physical computing is not intended as a general approach to physical computing. Nevertheless, at least for the discussed specific tasks, it has proven to be able to solve many issues emerged while using microcontrollers. First of all, the overall hardware/software chain is reliable, as it uses very stable technologies (both on software and hardware side). Moreover, the setup is of immediate realization, as it uses standard audio connectors to link the element of the chain. The only element to be created from scratch is the rectifier, but its assembly is a trivial task. Probably, the most relevant benefit that comes from us-

ing audio signals as control signals for electronic devices is the deep integration between electronic music competences and physical computing, since strategies for algorithmic generation of digital signal can be directly applied to the control of physical objects. Finally, the setup has revealed a specific flexibility, as it allows to mix in output sound from electromechanical devices (like the ones discussed before), with digital sounds (and, moreover, in a sort of continuum, sounds not generated but amplified by motors can be added). The setup devised for *Cifre del colpo* is also passible of further developments in terms of sound installation. In particular, it can evolve into a feedback system, where the audio output of the sound bodies, captured in input by microphones, is mapped into audio signals for the sound bodies themselves, like in audio feedback installations (see [15]).

Acknowledgments

A thank is due to Francesco Richiardi for technical inspiration and support, to Enrico Cosimi for his suggestions on analog synths, and to Antonino Secchia for his dedication to the performance of *Cifre del colpo*. I am also grateful to the anonymous reviewers for their insightful comments.

8. REFERENCES

- [1] D. O’Sullivan and T. Igoe, *Physical Computing. Sensing and Controlling the Physical World with Computers*. Boston, Mass.: Course Technology, 2004.
- [2] T. Igoe, *Making Things Talk*. Beijing–Cambridge–Farnham–Köln–Sebastopol–Taipei–Tokyo: O’Reilly, 2007.
- [3] N. Collins, *Handmade Electronic Music. The art of hardware hacking*. New York–London: Routledge, 2006.
- [4] S. Goto, “The case study of an application of the system, ”bodysuit” and ”roboticmusic”: its introduction and aesthetics,” in *Proceedings of the 2006 conference on New interfaces for musical expression*, ser. NIME ’06. Paris, France, France: IRCAM & Centre Pompidou, 2006, pp. 292–295. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1142215.1142287>
- [5] A. Valle, “The Rumentarium project,” in *Proceedings of the international conference on Multimedia*, ser. MM ’10. New York, NY, USA: ACM, 2010, pp. 1413–1416. [Online]. Available: [art05609s-valle](http://portal.acm.org/citation.cfm?id=1142215.1142287)
- [6] P. Tamborrini, *Design sostenibile. Oggetti, sistemi e comportamenti*. Milan: Electa, 2009.
- [7] S. Seriff, *Recycled Re-seen*. Santa Fe: Museum of International Folk Art, Santa Fe, 1996, ch. Folk Art from the Global Scrap Head: The Place of Irony in the Politics of Poverty, pp. 8–29.
- [8] M. Banzi, *Getting started with Arduino*. Sebastopol: O’Reilly, 2009.
- [9] S. Wilson, D. Cottle, and N. Collins, Eds., *The SuperCollider Book*. Cambridge, Mass.: The MIT Press, 2011.
- [10] AMP2, “Hopeful Monster,” Die Schachtel-Zeit Imp1.9, Milan.
- [11] S. Kersten, M. A. Baalman, and T. Bovermann, *The SuperCollider Book*. Cambridge, Mass.: The MIT Press, 2011, ch. Ins and Outs: SuperCollider and External Devices, pp. 105–124.
- [12] J. S. Wilson, Ed., *Sensor Technology Handbook*. Burlington, MA – Oxford: Newnes, 2005.
- [13] H. Austerlitz, *Data Acquisition Techniques Using PCs*, 2nd ed. San Diego – London: Academic Press, 2003.
- [14] C. Roads, *The Computer Music Tutorial*. Cambridge, MA, USA: MIT Press, 1996.
- [15] A. Di Scipio, “Sound is the interface. sketches of a constructivistic ecosystemic view of interactive signal processing,” in *Proceeding of the XIV CIM 2003*, N. Bernardini, F. Giomi, and N. Giosmin, Eds., Firenze, 2003, pp. 128–131.

THE VOWEL WORM: REAL-TIME MAPPING AND VISUALISATION OF SUNG VOWELS IN MUSIC

Harald Frostel, Andreas Arzt, Gerhard Widmer

Department of Computational Perception
Johannes Kepler University, Linz, Austria
harald.frostel@jku.at

ABSTRACT

This paper presents an approach to predicting vowel quality in vocal music performances, based on common acoustic features (mainly MFCCs). Rather than performing classification, we use linear regression to project spoken or sung vowels into a continuous articulatory space: the IPA Vowel Chart. We introduce a real-time on-line visualisation tool, the *Vowel Worm*, which builds upon the resulting models and displays the evolution of sung vowels over time in an intuitive manner. The concepts presented in this work can be used for artistic purposes and music teaching.

1. INTRODUCTION

An important aspect in singing is the production of distinct, recognisable vowels. The work presented in this paper aims to automatically recognise and track two perceptually important qualities ('open-/closeness' and 'front-/backness') of vowels in sung music, in real-time (and, by implication, recognising the vowels themselves). This would have many applications in science, music teaching, and art.

In the speech research and phonetics communities numerous studies have focused on the relationship between acoustic signal parameters of (spoken) vowels and their phonological categories and perceivable qualities (e.g., [1], [2], [3], [4], [5], [6], [7]). In particular, Pfitzinger [3], [4], [5], [6] has recently shown that such automatic mappings from acoustic to articulatory features are possible and can even match the performance of trained phoneticians.

With this paper, we wish to introduce the Sound and Music computing (SMC) community to that body of work and demonstrate that vowel qualities can be recognised also in vocal music performance. We first present systematic experiments that corroborate Pfitzinger's findings, on a different vowel corpus. In particular, we show that an effective mapping can also be learned on the basis of Mel Frequency Cepstral Coefficients (MFCCs) (which are routinely computed in many SMC applications). We then present an experimental tool that tracks and visualises sung vowels over time – as trajectories in a common phonological 'vowel space' (see Section 2) – in a real-time, on-line

setting. In analogy to [8] we call this the *Vowel Worm*. (The figure in Section 5 and several demo videos (see Section 5) explain why.) It provides us with preliminary (though still somewhat anecdotal) evidence that this kind of mapping approach is indeed viable for the singing voice.¹

The focus of the work presented here is not on categorical *classification* of vowels but on a mapping into a continuous (and phonologically motivated) two-dimensional space. Real-time categorical vowel recognition can easily be built on top of this – either via distance-based classification in the visualisation space or by modelling the vowel classes as MFCC distributions (e.g., in the form of Gaussian Mixture Models) and performing maximum-likelihood classification.

Our work is motivated by an artistic goal (real-time, on-stage music visualisation), but it could also be useful in music teaching – in particular, as a feedback tool in the training of singers, as briefly discussed in Section 6.

The paper is organised as follows. First, in Section 2, we explain the articulatory space we use for mapping and visualisation of vowel quality. In Section 3 we present our methods of modelling the projection into this space. Training and evaluation of these models are described in Section 4. We give insight into how this model was realised and used to visualise the vowel trajectory in Section 5. Finally, in Section 6, we discuss the results and possible scenarios where our approach might be applicable.

2. THE PERCEPTUAL SPACE OF VOWELS AND THE IPA VOWEL CHART

In the literature, mainly two kinds of diagram or chart are used to illustrate and classify articulatory vowel quality.

One type is the *formant frequency space* (e.g., [6], [2], [1]). The simplest version is a two-dimensional diagram with the first and the second formants F_1 and F_2 defining the axes, and vowels positioned in the diagram according to their formant frequencies. Variations exist with respect to the scaling of the formant frequencies (e.g., Hertz or Bark) and/or the usage of differences between formant frequencies instead of absolute values.

The second type of diagram is the *Cardinal Vowel Diagram* [11] and its newer adaptation by the International

¹ This is notable because in [9] and [6] it was shown that the fundamental frequency (F_0) has a significant effect on the formant frequencies and, in particular, on the 'vowel height' (which is one of the articulatory dimensions we wish to recognise). In singing, this effect is expected to be much more pronounced than in speech, which was the focus of previous research.

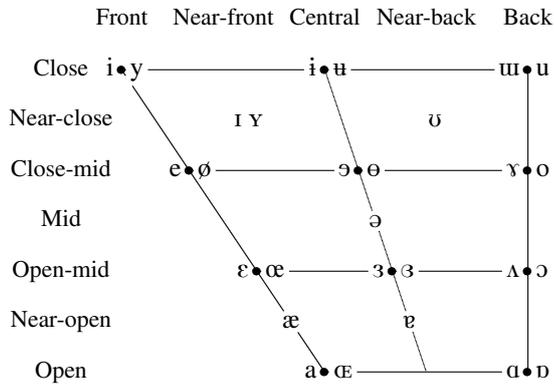


Figure 1. The IPA Vowel Chart [10]. The horizontal axis depicts the *vowel backness* and the vertical axis the *vowel height*. Vowels at the right and left of bullets are *rounded* and *unrounded*, respectively.

Phonetic Association (IPA), the vowel quadrilateral or *IPA Vowel Chart* [10] as shown in Figure 1. In our work we use the latter as a space to visualise vowel quality.

The IPA Vowel Chart locates vowels in terms of the tongue position required for their production. One dimension refers to the *backness* of the vowel, ranging from *front* to *back*. If the tongue or its highest point is placed near the front of the mouth (the hard palate), the vowel is labelled as a front vowel, whereas if the highest point of the tongue is placed at the back, narrowing the pharynx, the vowel is labelled as a back vowel.

The second dimension is *height*. If the tongue is near the roof of the mouth, the vowel height is described as *close*.² A vowel produced with maximum distance between tongue and palate is described as *open*. The eight primary cardinal vowels [i], [e], [ɛ], [a], [ɑ], [ɔ], [o], and [u] define the reference points in this chart. All other vowels can be placed in positions between them [10].

A third, partially independent dimension is the *lip rounding*. A vowel is called *rounded* if the lips are rounded during its production, and *unrounded* if the lips are relaxed.

3. VOWEL QUALITY PREDICTION

The objective of the work described here is to develop a system that can recognise, track, and visualise vowel qualities in sung music by mapping them into the IPA chart in real-time using a suitable regression model that is based on common audio features as routinely used in Music Information Retrieval (MIR) and SMC.

3.1 The Space

In order to build such a model, it is necessary to define a space on the basis of the IPA Vowel Chart. We simply place the vowel chart in a two-dimensional *Cartesian coordinate system* as shown in Figure 2. The proportions of the vowel chart used here are 2:3:4 for the bottom, right, and top sides respectively [12]. The backness coordinates

² Not closed, as one might expect from the naming of the other extreme ('open'). The vowel height dimension is sometimes also called 'closeness'.

of our space range from 0 (front) to 4 (back), and the height coordinates range from 0 (open) to 3 (close). Each vowel is therefore represented by a distinct point in this space.

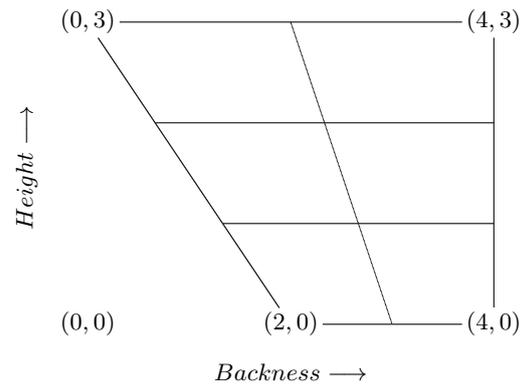


Figure 2. Coordinates of the vowel chart space used for regression with some sample points.

3.2 Multiple Linear Regression

As a predictor, we decided to use *multiple linear regression*. The advantages of this, compared to other methods like *local regression*, for instance, are its simplicity, robustness, and efficient implementation for real-time prediction.

$$y = \sum_i x_i \beta_i + \varepsilon = \langle \mathbf{x}, \boldsymbol{\beta} \rangle + \varepsilon \quad (1)$$

As shown in equation 1, multiple linear regression is basically the inner product of a (feature) vector \mathbf{x} (regressor) and a regression coefficients vector $\boldsymbol{\beta}$ plus an error term ε . It is necessary to extend \mathbf{x} (the features used as regressors) by a further dimension to model the constant term in the linear equation. For each of the two vowel dimensions (backness and height) a separate regression model is generated.

3.3 Features

As regressors we use the following features (or subsets of these):

- **Mel Frequency Cepstral Coefficients (MFCCs) [13].** The MFCCs are calculated using 40 mel spaced triangular filters covering the spectrum up to 8000 Hz. No pre-emphasis is used, and the areas of the filters are not normalised; in other words, all filters have the same height. A Hamming window is used to calculate the spectrum.
- **Linear Prediction Filter Coefficients (LPGCs) [14].** Linear prediction of order 13 is performed, resulting in 14 filter coefficients.
- **Fundamental Frequency (F0).** To obtain the fundamental frequency (and to decide whether one is present at all), we use a (real-time) F0 estimation algorithm developed earlier and also implemented in our visualisation tool (Section 5). The underlying algorithm builds upon a combination of approaches

presented in [15] and [16]. Several scalings of F_0 are used, namely *Hertz*, *logarithm*, *mel scale*, and *equivalent rectangular bandwidth (ERB) scale*.

Altogether, this results in 58 features (40 MFCCs, 14 LPFCs, 4 representations of F_0).

4. EXPERIMENTS

4.1 Corpus and Data Generation

We have not been able to find an annotated vowel database for sung vowels. Creating such a database is very labour-intensive, since in addition to recording the sung vowels, it also requires classification by several phoneticians who place them at the right positions in the IPA Vowel Chart, as done by Pfitzinger in [4] for spoken vowels. Thus, to build and validate our models, we relied on an existing database for spoken vowels.

We used the vowel corpus created for the *Vocal Joystick Project*³ [17], which consists of a large amount of recorded monophthongs (pure vowel sounds with no changes in articulation) and vowel-to-vowel transitions (articulation moves from one vowel to another). The corpus features 9 vowels, namely [i], [e], [æ], [a], [ɑ], [o], [u], [ɪ], and the schwa [ə], spoken by multiple speakers (male and female with different native languages) and recorded at various sound levels, intonations, and durations [18]. In addition, each recording was judged by a phonetician as to whether it is of acceptable quality (i.e., close enough to the target vowel). Figure 3 shows the vowels covered by the corpus on the IPA Vowel Chart.

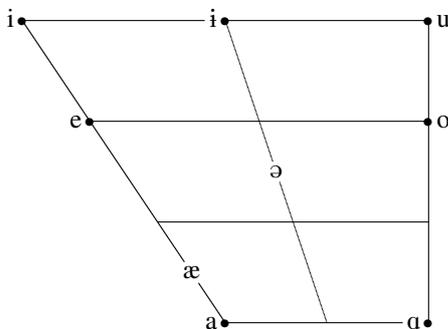


Figure 3. Vowels covered by the Vocal Joystick Corpus (cf. Figure 1).

For training and evaluation of our models, we used only utterances of monophthongs (no vowel-to-vowel transitions) of acceptable articulatory quality. After removing problematic utterances (i.e., those too far from the target vowel), we only kept speakers that still covered all 9 vowels. This resulted in a reduction from originally 92 speakers to 56. To create our data set, 150 uniformly distributed random points in time were generated for all corresponding utterances per speaker and vowel. Only voiced segments were taken into account, that is, segments with a detected fundamental frequency. This resulted in a total of 75,600 samples (56 speakers \times 9 vowels \times 150 samples). From these

³ Freely available at <http://www.vocaljoystick.org/>

random points in time, the features (MFCCs, LPFCs, and F_0) were extracted.

4.2 Experimental Results

As mentioned above, our data covers 56 different speakers. We performed a 56-fold cross-validation for our evaluation. For every fold, the samples of one speaker were left out for testing while the rest of the data was used for training the regression models. After training, the models were used to predict the backness and height of the samples from this speaker, resulting in a *leave-one-speaker-out* cross-validation.

As evaluation measures we used the *correlation coefficient* r and the *root mean square error (RMSE)* for each of the two dimensions backness and height. The *RMSE* was calculated from the results in a normalised space, which means that the backness and height predictions were divided by 4 and 3, respectively, to obtain a chart space ranging from 0 to 1 in each dimension.

| Features | t_{win} | r_b | $RMSE_b$ |
|--|-----------|---------|----------|
| MFCC ₁₋₄₀ , LPFC | 93ms | 0.8854 | 16.09% |
| MFCC ₁₋₄₀ , LPFC, F_{0Hz} | 93ms | 0.8853 | 16.10% |
| MFCC ₁₋₄₀ , LPFC | 46ms | 0.8826 | 16.27% |
| MFCC ₂₋₂₅ | 93ms | 0.8659 | 17.32% |
| MFCC ₂₋₂₅ , F_{0ERB} | 93ms | 0.8656 | 17.33% |
| MFCC ₂₋₂₅ | 46ms | 0.8608 | 17.61% |
| MFCC ₂₋₁₃ | 46ms | 0.8572 | 17.82% |
| MFCC ₁₋₆ | 23ms | 0.8130 | 20.15% |
| Baseline | – | 1.6e-13 | 34.61% |

Table 1. Features and results for backness regression models sorted by r_b .

| Features | t_{win} | r_h | $RMSE_h$ |
|---|-----------|--------|----------|
| MFCC ₁₋₄₀ , LPFC, F_{0Hz} , F_{0Log} , F_{0ERB} , F_{0Mel} | 93ms | 0.8554 | 20.36% |
| MFCC ₁₋₄₀ , LPFC, F_{0ERB} | 93ms | 0.8551 | 20.38% |
| MFCC ₁₋₄₀ , LPFC | 93ms | 0.8540 | 20.45% |
| MFCC ₂₋₂₅ , F_{0ERB} | 93ms | 0.8526 | 20.53% |
| MFCC ₂₋₂₅ , F_{0ERB} | 46ms | 0.8502 | 20.68% |
| MFCC ₂₋₄₀ | 93ms | 0.8501 | 20.69% |
| MFCC ₂₋₂₅ | 46ms | 0.8479 | 20.83% |
| LPFC, F_{0ERB} | 93ms | 0.6701 | 29.17% |
| Baseline | – | –8e-17 | 39.28% |

Table 2. Features and results for height regression models sorted by r_h .

We tried several window sizes, zero-padding sizes, and feature combinations to determine which settings perform best. Tables 1 and 2 show the results for several feature combinations and window sizes (t_{win}) but list only a subset of all tested combinations. Zero-padding does not seem to have any significant influence on the quality of the MFCCs. For performance reasons, we thus omitted zero-padding in the final results. The window size, however, does have an

impact: the larger the window, the better the regression. This is not surprising, as only monophthongs were used for training, and therefore a larger window covers more information and might cancel out variations in the signal. Furthermore, if a sample happens to be at the beginning or end of a vowel, a larger window captures more of the relevant signal. The baselines shown in Tables 1 and 2 represent a predictor that outputs the average backness and height.

The best *backness* correlation ($r_b = 0.8854$) was obtained with a window size of 93 *ms*, all MFCCs and LPFCs as features. Adding the fundamental frequency did not lead to any improvement. Without LPFCs, the best results were obtained with MFCCs 2 to 25 (excluding the first coefficient). The prediction of backness seems to be very robust in terms of the feature combinations. The worst result with only 6 MFCCs and a window of 23 *ms* still achieved a correlation coefficient of $r_b = 0.8130$.

Vowel *height* seems to be more sensitive to the choice of feature combination. Again, the best result was obtained with all features, but also including all scalings of F_0 . The height predictions improved with the usage of F_0 . This finding is in agreement with Pfitzinger's results [3], [5], [6], which showed that F_0 influences the perceived vowel height. However, the impact is rather small in our case. Also, the scaling of F_0 has no major influence. The best results were obtained by using the F_0 in ERB scale. The reason for the limited effect of the fundamental frequency might be that the MFCCs themselves encode some amount of pitch information.

For comparison, in [5], Pfitzinger achieved results of $r_b = 0.964$, $r_h = 0.903$ and $r_b = 0.965$, $r_h = 0.960$ without and with F_0 , respectively. However, it is difficult to compare these results with ours. Pfitzinger used only 12 German speakers and different vowels. The vowel stimuli were presented to 40 trained phoneticians, each of whom assigned the stimuli to precise positions in the vowel chart (whereas we were limited to assigning training vowels to their grid positions in the chart, because we only had vowel labels as ground truth). It could thus be argued that the training data used in [4] is substantially more refined than the data available to us.

Note that vowel quality assessment is generally not unambiguous. Dioubina and Pfitzinger [12] stated that the judgement of vowel quality by phoneticians is influenced by their native language. In addition, according to [4], even skilled phoneticians cannot reliably repeat their own judgements after some time.

Overall, we conclude from the experiments for the purposes of our project that (1) the qualities backness and height of spoken vowels can be predicted reasonably well by linear regression, and (2) this can be done by using MFCCs, which are routinely computed and used in MIR and SMC applications as underlying audio features. Whether these results can be extended to *sung* vowels in music will need to be established quantitatively in future experiments – should an annotated corpus of sung vowels becomes available. In Section 5, we give some anecdotal evidence of this by supplying examples of our vowel visualiser in action.

4.3 Final Model

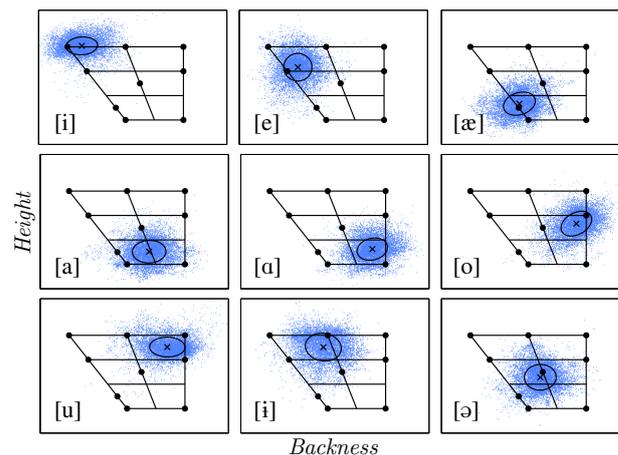


Figure 4. Scatter plots of the predictions of the final model for all 9 vowels (cf. Figure 3). Each dot marks the predicted point of one test sample. Each cross and circle indicates the mean and standard deviation of all predictions of one class.

We used 24 MFCCs (coefficients 2 to 25) with a window size of $t_{win} = 46$ *ms* as the final model parameters in the backness prediction. For height, two models were generated: one based only on the same 24 MFCCs as for backness, and one based on the 24 MFCCs plus the F_0 in ERB scale. The system switches automatically between the models, depending on the presence of a valid F_0 . The chosen models constitute a trade-off between run-time and accuracy. A longer signal window raises computational costs in the calculation of the spectrum and the MFCCs, and also generally the tracker's latency. An effect similar to that of a larger window can be obtained by smoothing the final predictions of backness and height over time. Figure 4 shows scatter plots of the predictions of the trained models.

5. THE VOWEL WORM

Our ultimate goal is the visual tracking of vowels (and other aspects of singing) in artistic musical contexts. From previous experiments with categorical vowel classification, we learned that a simple textual display of the currently recognised vowel is not very useful. Vowel changes happen too fast for the viewer to process and validate these impressions. Also, classifying vowels into nominal classes has several drawbacks. Spoken or sung vowels are almost always a mixture of adjacent vowel classes. Moreover, classification limits the set of vowels to those used in training. If there are, for instance, only 9 vowel classes in the training data (as in the Vocal Joystick corpus), the classifier will recognise only those. With regression, on the other hand, only a subset of vowels is needed to develop a model that is (at least in theory) capable of projecting any vowel or input into a vowel space, even if it was not available during training.

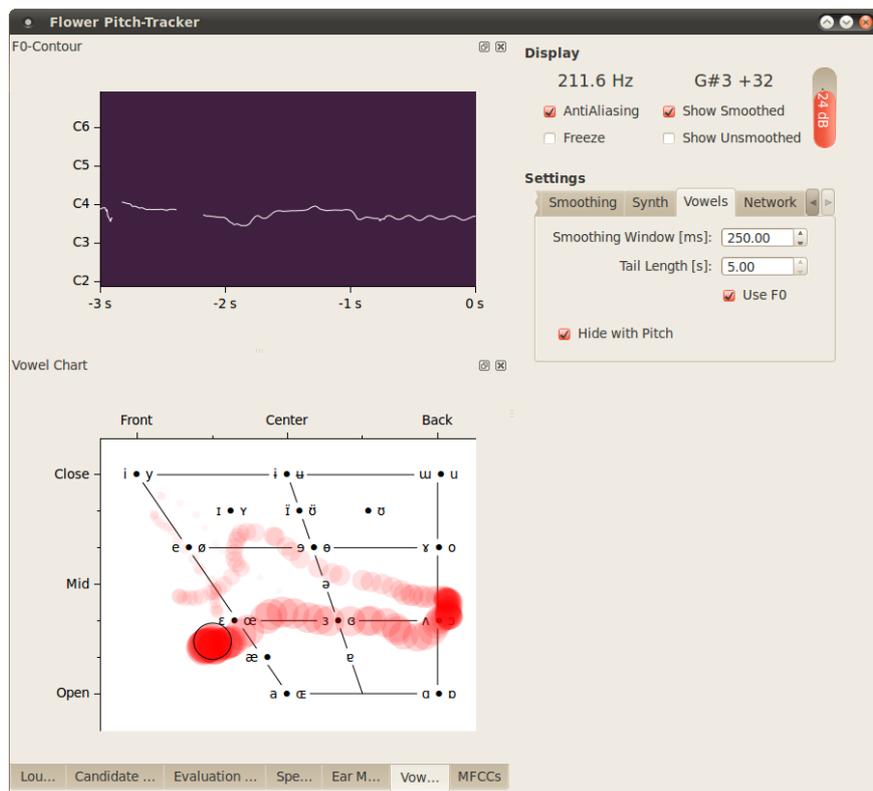


Figure 5. Screenshot of the real-time visualisation of the Vowel Worm. The upper left panel depicts the pitch contour of the last 3 seconds – as computed by our on-line pitch tracker. The bottom left panel shows the trajectory of the sung vowels (Vowel Worm). On the right, the Vowel Worm settings can be adjusted.

These considerations led us to developing the *Vowel Worm*.⁴ It not only displays the current vowel in a more intuitive manner by positioning it in the vowel chart plane, it also captures the evolution of the sung vowels over time. The Vowel Worm uses the regression models described above to perform real-time mapping of an incoming audio signal (ideally a solo singing voice) onto the IPA Vowel Chart visualisation plane. The current position in the chart is indicated by a filled circle, while instances further in the past appear smaller and fainter. Figure 5 shows a screenshot of our visualisation tool during a live performance. The top left panel shows the estimated F_0 -contour for the last 3 seconds, as determined by our pitch tracker. The Vowel Worm is located at the bottom left.

The Worm takes the MFCCs and the fundamental frequency F_0 (computed on-line) as input. It then calculates backness and height via the model’s regression formula. Depending on the presence of a valid F_0 , the implementation chooses the appropriate regression model for predicting the height. Backness and height are further smoothed over time. Smoothing is done for each dimension separately by averaging over the last predictions. This improves the visual stability of the projection, which otherwise tends to jitter. Typical smoothing windows range from 150 *ms* to 350 *ms*. Other settings that can be adjusted during run-time are the tail length of the worm (i.e., the time span into

the past that is visualised), whether F_0 is to be used, and whether the worm should hide in the absence of a fundamental frequency.

To give the reader an impression of the visualisation, we have generated a few *screen shot videos* of the Worm in action. They can be found at <http://www.cp.jku.at/projects/realtime/vowelworm.html>. In these examples, the input audio stream comes from an audio file, but the system works in exactly the same way with real-time input via microphone, for instance.

6. CONCLUSIONS AND DISCUSSION

This paper has addressed the problem of real-time vowel quality recognition and tracking, and introduced a particular way of mapping and visualising the development of sung vowels over time. The main result, as we see it, is that two central articulatory features of (spoken) vowels seem to be reliably predictable from standard MFCC features and that – on the basis of our unsystematic qualitative experiences with the Vowel Worm – models learned from spoken vowels appear also to apply (perhaps unexpectedly) well to sung vowels. Carrying out systematic quantitative experiments to support this would require the availability of precisely annotated musical corpora.

The concrete goal of the present project is to develop real-time music analysis and tracking technology that can be used to control live on-stage visualisations of large musical works (e.g., operas). To that end, we are also working on

⁴ Concept and name were inspired by previous work on real-time visualisation of expressive performance parameters in the *Performance Worm* [8].

tracking other parameters (including exact score position) in, for instance operatic singing. For artistic visualisation purposes, the current recognition capabilities of the Vowel Worm may be considered sufficient.

Beyond artistic visualisation projects, we envision application in a vocal quality visualiser, particularly in didactic settings, and as a feedback tool for the training of singers, actors, etc. A prerequisite for this, however, would be precise quantitative experiments that establish whether, and to what extent, the placement predicted by learned models are reliably correct, which in turn depends on the availability of high-quality training and validation corpora of sung vowels (of various musical styles).

Acknowledgments

This research is supported by the City of Linz, the Federal State of Upper Austria, and the Austrian Research Fund (FWF) under grant TRP109-N23. The FLOWER real-time audio processing framework, upon which the Vowel Worm is built, is being developed by Martin Gasser (Austrian Research Institute for Artificial Intelligence, Vienna), supported by FWF project Z159.

7. REFERENCES

- [1] W. Klein, R. Plomp, and L. C. W. Pols, "Vowel Spectra, Vowel Spaces, and Vowel Identification," *The Journal of the Acoustical Society of America*, vol. 48, no. 4B, pp. 999–1009, 1970.
- [2] M. Aylett, "Using Statistics to Model the Vowel Space," in *Proceedings of the Edinburgh Linguistics Department Conference*, 1996, pp. 7–17.
- [3] H. R. Pfitzinger, "Dynamic Vowel Quality: A new Determination Formalism based on Perceptual Experiments," in *4th European Conference on Speech Communication and Technology (EUROSPEECH '95)*, vol. 1, Madrid, Spain, Sep. 1995, pp. 417–420.
- [4] —, "Acoustic Correlates of the IPA Vowel Diagram," in *Proceedings of the 15th International Congress of Phonetic Sciences*, vol. 2, Barcelona, Spain, Aug. 2003, pp. 1441–1444.
- [5] —, "The /i/-/a/-/u/-ness of Spoken Vowels," in *8th European Conference on Speech Communication and Technology*, Geneva, Switzerland, Sep. 2003, pp. 809–812.
- [6] —, *Speech Production and Perception: Experimental Analyses and Models*. Berlin: ZAS Papers in Linguistics, 2005, vol. 40, ch. Towards Functional Modelling of Relationships between the Acoustics and Perception of Vowels, pp. 133–144.
- [7] S. Ran, B. Millar, and P. Rose, "Automatic Vowel Quality Description using a Variable Mapping to an Eight Cardinal Vowel Reference Set," in *Proceedings of the 4th International Conference on Spoken Language (IC-SLP 96)*, vol. 1, Oct. 1996, pp. 102–105.
- [8] S. Dixon, W. Goebel, and G. Widmer, "The Performance Worm: Real Time Visualisation of Expression based on Langner's Tempo-Loudness Animation," in *Proceedings of the International Computer Music Conference (ICMC)*, Göteborg, Sweden, Sep. 2002, pp. 361–364.
- [9] J. Sundberg, *The Science of the Singing Voice*. Northern Illinois University Press, 1987.
- [10] International Phonetic Association, *Handbook of the International Phonetic Association : A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press, Jun. 1999.
- [11] D. Jones, *An Outline of English Phonetics*, 9th ed. W. Heffer & Sons Ltd., 1962.
- [12] O. I. Dioubina and H. R. Pfitzinger, "An IPA Vowel Diagram Approach to Analysing L1 Effects on Vowel Production and Perception," in *7th International Conference on Spoken Language Processing*, vol. 4, Denver, Colorado, USA, Sep. 2002, pp. 2265–2268.
- [13] S. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, Aug. 1980.
- [14] B. S. Atal and M. R. Schroeder, "Predictive Coding of Speech Signals," in *Proceedings of the 1967 IEEE Conference on Communication Processing*, 1967, pp. 360–361.
- [15] A. Camacho and J. Harris, "A Pitch Estimation Algorithm Based on the Smooth Harmonic Average Peak-to-Valley Envelope," in *IEEE International Symposium on Circuits and Systems (ISCAS 2007)*, May 2007, pp. 3940–3943.
- [16] H. Frostel, "Real-time Fundamental Frequency Estimation of the Human Voice," Master's thesis, Johannes Kepler University, Linz, Austria, 2009.
- [17] J. A. Bilmes, X. Li, J. Malkin, K. Kilanski, R. Wright, K. Kirchhoff, A. Subramanya, S. Harada, J. A. Landay, P. Dowden, and H. Chizeck, "The Vocal Joystick: A Voice-Based Human-Computer Interface for Individuals with Motor Impairments," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, Oct. 2005, pp. 995–1002.
- [18] K. Kilanski, J. Malkin, X. Li, R. Wright, and J. Bilmes, "The Vocal Joystick Data Collection Effort and Vowel Corpus," in *Proceedings of the International Conference on Spoken Language Processing*, Pittsburgh, Pennsylvania, USA, Sep. 2006.

SONIC GESTURES AS INPUT IN HUMAN-COMPUTER INTERACTION: TOWARDS A SYSTEMATIC APPROACH

Antti Jylhä

Department of Signal Processing and Acoustics
Aalto University, School of Electrical Engineering
antti.jylha@aalto.fi

ABSTRACT

While the majority of studies in sonic interaction design (SID) focuses on sound as the output modality of an interactive system, the broad scope of SID includes also the use of sound as an input modality. Sonic gestures can be defined as sound-producing actions generated by a human in order to convey information. Their use as input in computational systems has been studied in several isolated contexts, however a systematic approach to their utility is lacking. In this study, the focus is on general sonic gestures, rather than exclusively focusing on musical ones. Exemplary interactive systems applying sonic gestures are reviewed, and based on previous studies on gesture, the first steps towards a systematic framework of sonic gestures are presented. Here, sonic gestures are studied from the perspectives of typology, morphology, interaction affordances, and mapping. The informational richness of the acoustic properties of sonic gestures is highlighted.

1. INTRODUCTION

The connection between gesture and sound has been a point of intensive study during the past decade in the field of sound and music computing (SMC). We have witnessed numerous musical performances, software applications, and human-computer interfaces highlighting the use of gestural control of sound output. Also theoretical advances in action-sound relationship and sound-related gesture research have been presented, making use of both results and theories from the field of human-computer interaction (HCI) and classical theories on sound objects, and applying these to SMC especially in the domain of gesture in music [1, 2, 3, 4, 5].

The mainstream of gesture research in the field of SMC has concentrated on "traditional" viewpoints of gestures and their acquisition. This is to say that although definitions of gesture differ dependent on the context, gesture is seen as a human-generated perceivable physical action, which is often analyzed by means of haptic input or computer vision. For example, in the context of sonic interaction design (SID), most of the presented interaction

paradigms and interfaces rely on haptic and/or visual input, which is analyzed to inform sound production. In musical controllers, it is usually the physical gesture that is mapped to the sound output, either as sound-generating or sound-modifying action.

However, SID has been defined as studying sound as the conveyor of information, aesthetics, and emotions, which does not imply a one-way approach to the use of sound at the interface. Instead, it can be interpreted to also consider sound propagating to the other direction in the interaction loop, i.e., from the human to the computer. Only recently the sounds generated by humans have started to gain more attention in the field in the context of controlling interfaces and applications. Looking back, isolated examples and studies on sonic input can be found, but a common perspective on using sound as a key input modality has not been presented.

This study concentrates on the notion of sonic gesture as input in HCI, which is an interesting topic for several practical reasons. First, sounds do not require specialized hardware to acquire, as most computational devices, be they computers or mobile devices, are equipped with a microphone. Rather, the challenge is in the processing of the sounds to acquire meaningful information from them [2]. Second, sonic gestures facilitate remote interactions, i.e., the user does not need to touch the operated device. Third, sound as an input modality can work in situations, where looking at the device is not possible (eyes-busy situations, visual impairment). Fourth, some sonic gestures can provide an alternative means of accessing computers and applications for people with motor impairment.

This work will discuss the use of sonic gestures as input in HCI and SMC fields, basing the discussion on previous studies and examples of sonic gestures in action. While a large portion of previous studies are musically oriented, this study considers sonic gestures in general, stripped from the constraint of their use exclusively in musical contexts. The aim is to take steps towards a systematic approach to sonic gestures in terms of the types of interaction and information different sonic gestures afford.

2. SONIC GESTURES

This work defines sonic gesture as a sound-producing action generated by a human in order to convey information to a computational system. This definition differs from previous definitions in that the gesture itself is always a sound producing action and does not necessitate an instru-

ment for production, although sonic gestures can be instrumental, too. Furthermore, while there is a vast and growing body of research looking into gesture from the musical perspective, in this study the sonic gesture itself does not necessarily comprise musical elements.

Considering the above definition, it is important to note that the perspective of this study differs from the classical action-sound perspective. Here, the sound always occurs prior to computation, i.e., this work does not consider as sonic gesture for example wielding an accelerometer-equipped controller in the air and mapping this motion into sound synthesis parameters. In this work, the information conveyed lies within the human-generated sound itself, and the examination focuses on its acoustic properties and the use of these for informing or controlling an interactive system, rather than trying to infer the gesture behind the sound.

While higher-level aspects of sonic gestures, such as emotion, social connotations, or Chion's concept of *ergo-audition* [6] (the experience of hearing the sounds of one's own acting) are definitely relevant for utilizing sonic gestures in context, from HCI perspective capturing this information from sound with a computational system is still mostly a challenge of the future. Nevertheless, when sonic gestures are performed in interactive contexts, there are actually two levels of feedback the user gets: the sound and sensation from the sonic gesture itself, and the feedback from the computational system. The former is in practice always multisensory, containing typically auditory and haptic components, while the modalities of the latter vary dependent on the form, function, and design of the interactive system.

The simple case of a hand clap is a good basic example of a sonic gesture. It is clear that it is a sound-producing action - with a very distinct sound - generated by a human. As will be discussed below, this simple gesture can convey lots of information with its acoustic properties. While a hand clap can be considered a gesture by itself, the information it conveys is ultimately dependent on the context in which it is produced. It can be argued that sonic gestures become meaningful only when they have been associated with a meaning, which in HCI is achieved usually by means of mapping the gesture or some of its properties to a command on the computational device.

Jensenius has categorized definitions of gesture into three groups: gesture as communication, gesture for control, and gesture as mental imagery [3]. In communication, gesture is seen as means for social, interpersonal interaction, whereas mental imagery refers to studying gestures as mental processes. In this study, the focus is mainly on gestures for control, which can be seen as the traditional HCI perspective for gestures. However, as we shall see from examples, some approaches to sonic gesture also relate closely to both communication and mental imagery, even if the aim is in control. Also, control in this work refers not only to giving commands but to convey information that is essential for interaction in a broader context.

In contrast to the gesture definition of Cadoz [7], which excludes all vocal sounds, sonic gestures can be produced

also vocally. Indeed, non-speech utterances, humming, and mouth-generated sounds provide a very rich gestural repertoire. Their utility in designing sonic interactions has been discussed by Ekman and Rinott [8] in their influential work on vocal sketching. Dessein and Lemaitre [9] have explored the capability of humans to imitate everyday sounds vocally, and found out that there exists a strong connection between the classification performed by humans and that performed for real everyday sounds by acoustic descriptors. In addition, as shown by Sporka [10], there is a lot of intuitive information in how people communicate by pitch alone to indicate confirmation, negation, uncertainty, and surprise, among others. While it is a very simple acoustic descriptor, pitch can be utilized in numerous ways in interactive systems, too.

Van Nort has studied sonic gestures from a musical perspective [4]. He defines sonic gestures in the context of instrumental excitation and interaction design, and carefully dissects the gestures into possible control structures based on their acoustic morphology. He presents a perspective on mapping as more than just the link between control and output, highlighting the importance of mental images in the perception on musical dynamics and the gestures used in music production.

2.1 Previous studies on sonic gesture interfaces

The range of sonic gestures is broad and several studies have proposed interfaces using some type of sonic gesture for a particular problem. While the examples here are by no means exhaustive, they show the variety of different gestures and approaches to their utilization, which will be used as basis for discussion in Section 4.

Vesa and Lokki have presented a music player control interface using finger snaps [11]. The system utilizes two microphones integrated to the headphones of the user and is capable of detecting which snaps occur on the left or right side of the user's head or in front of it. This information is mapped to previous/next track and play/pause functionalities found in all music players nowadays.

Jylhä and Erkut have developed a hand clap interface for sonic interactions with the computer [12]. From a stream of percussive sound events, the interface can extract information on the event type (i.e., hand configuration) and tempo. This information can then be used to indicate control information in various applications. It has been demonstrated on giving discrete commands to the system, controlling the tempo of music, and entraining a virtual audience to the user's clapping. More recently, the interface has been applied and extended to an interactive Flamenco hand clapping tutor application, in which also accentuation (clap strength) and temporal deviation of the user's clapping are extracted [13]. This information is applied to inform rhythmic output from the system, and to monitor the performance of a learning clapper.

Rocchesso, Polotti, and Delle Monache have studied continuous sonic interactions based on kitchen activities and sounds [14]. As one case example, they consider carrot cutting, a rhythmic activity, which they sonify with several different feedback strategies. As one input modality, they

utilize the contact sound resulting from the knife hitting the table, and perform beat-tracking on the sound. Providing rhythmic sonic feedback with an adaptive tempo and upbeat rhythm seemed to result in the most relaxed action by the cutter.

Vocal Joystick [15] is an interface enabling the user to control for example the mouse cursor by vowel sounds. From vowels, the interface extracts energy, pitch, and vowel quality. Energy is mapped to the velocity of cursor movement, while vowel quality is mapped with a continuous two-dimensional mapping into movement direction. The Vocal Joystick has been shown to compete with eye-tracking based cursor movement interfaces.

Another mouse-replacement interface has been presented in [16], based on humming and hissing. A four cell mouse grid is used, and a cell is selected by low-frequency or high-frequency humming. Hissing is detected and mapped to a mouse click event.

Sporka [10] has presented several methods and applications around using pitch-based vocal input in HCI, including target acquisition by absolute and relative pitch, mouse cursor control by whistling or humming using pitch and loudness parameters, non-speech control of keyboard emulation by mapping three-element pitch patterns to keyboard keys thus forming an alphabet of sonic gestures, and controlling two computer games by vocal input. The methods have been designed especially for hands-busy situations and people with motor impairment.

Hämäläinen has presented computer game applications incorporating vocal input as part of the interface [17]. In one game, shouting is used to control the fire-breathing of a dragon avatar, while in other ones voice pitch controls the avatars' movements.

Billaboop is an interface which allows the user to play virtual drums by beatboxing sounds [18]. The system captures the sonic gestures of the user and by means of machine learning triggers drum sounds corresponding to the detected sounds. The system is also capable of reacting to table drumming in the same way. Recently, a mobile application called BoomClap¹ from the same origin has been presented. The application can be taught which sounds the user wants to use in the interaction.

Considering instrumental sonic gestures, Scratch Input demonstrates how scratches on surfaces can form a rich sonic gesture repertoire [19]. The sounds are captured by a contact microphone and different gestures are recognized by their sound trajectory. Given that structure-borne sound travels far and is highly tolerant for unwanted environmental sounds, the technique is relatively robust.

As an innovation to musical applications for mobile devices, the iPhone Ocarina application presents an interface where the blowing of the user to the microphone of the device acts as excitation of sound [20]. This is a good example of a natural interface, where the interaction with the computational device is very close to that with a real ocarina.

3. TYPOLOGICAL AND MORPHOLOGICAL CONSIDERATIONS

As discussed for example by [5] and [4], gestures can be divided into three categories based on their macro-level morphology: impulsive, iterative, and sustained gestures. Iterative and sustained gestures have also been labeled continuous gestures [1], but as argued by Van Nort, iterative and continuous gestures can be seen as separate categories [4]. Isolated hand claps and finger snaps, for example, are impulsive sonic gestures, whereas whistling and humming are sustained. Continuous hand clapping with a relatively constant tempo is an iterative gesture, consisting of sequential impulsive gestures. It is noteworthy, however, that in principle any basic gesture type - be it impulsive or sustained as an isolated case - can be sequentially produced to form iterative gestures.

Typologically, it is clear that a hand clap is a different type of gesture than a whistle. However, it is possible to expand the gestural typology by considering different types of the same gesture class as their own subtypes. For example, it has been shown that different hand configurations in clapping result in audibly different sounds and that it is possible by machine learning techniques to also differentiate between these types computationally [21, 22]. Furthermore, it is possible to consider such a typology as a continuum, as is done for example in the Vocal Joystick example for vowels [15]. While this continuum may rely on anchors, for example eight vowels of spoken language, to form a basis for the mapping space, the "in-between" vowels can be used to provide a continuum rather than a class-based typology. Similarly for pitch, it is possible to produce melodies by sounds of different constant pitches following each other, or to continuously vary the pitch, e.g., with a glide up and down in pitch.

Considering acoustic morphology further, sonic gestures can be categorized in several ways. There are unpitched sonic gestures such as hand claps, finger snaps, and table taps, and pitched ones like whistling and humming. On the other hand, the shape of sonic gestures can be static, e.g., humming with a constant pitch, or dynamic, e.g., humming with a varying pitch. It is possible to dwell deeper into the acoustic morphology, looking at Schaefferian principles of sound objects, as has been discussed by Van Nort [4] in the context of musical gesture, which would apply for the most part also to the morphologies of general sonic gestures. However, in this study the focus remains more on a macro-level.

It is also possible to see a connection between the sonic gesture typology and Gaver's map of everyday sounds [23], which examines the sounds generated by interacting materials starting from their fundamental sources (solids, gasses, liquids), and proceeding through basic sound-producing events into temporal patterning and more complex sounds. A similar approach can also be envisioned for sonic gestures, grouping them based on the basic-level events, temporal patterning, and combinatory events of multiple gestures.

An important aspect to consider in sonic gesture discussion is the sound-producing body. For sound-producing

¹ <http://billaboop.com/en/boomclap>

gestures, a categorization into empty-handed and instrumental gestures has been proposed [2]. Ballas [24] has labeled both of these as self-produced sound, including all sounds produced by the body or body movements, with or without interacting with an external surface or object. In the context of sonic gestures, empty-handed gestures can be considered as all sound-producing actions that the human is able to produce without an external sounding body. For example hand claps, finger snaps, whistling, all vocal sounds, and body tap sounds can be considered empty-handed gestures, even if hands can be used in their production. Instrumental sonic gestures are actions, in which the human interacts with a secondary physical object to generate sounds, e.g., footsteps and scratches or knocks on surfaces. Here, we consider as first-order instrumental sonic gestures those sound-producing actions, that involve direct human interaction on a secondary sounding body. Tapping a table or scratching a wall are first-order sonic gestures, as is blowing into the microphone of a mobile device as in [20] to create turbulent air flow sound. Second-order instrumental sonic gestures involve interacting with a secondary sounding body through a proxy object, for example hitting a table with a pen or throwing a ball to a wall. This class is so vast, however, that it is not discussed in this study.

4. EXTRACTABLE MAPPING PARAMETERS OF DIFFERENT SONIC GESTURES

Designing an interface around sonic gestures the designer is faced with the questions of what parameters of the sound need to be computed and how they can be mapped to the system functionality and output to provide meaningful interaction. Looking at the variety of sonic gestures, it is clear that different gestures can provide different types of information and, thus, are applicable for different purposes. Here we take a detailed look at an exemplary set of sonic gestures and present a set of parameters that can be computed from each gesture type, summarized in Table 1. The upper part of the table enlists empty-handed gestures, while the lower part considers first-order instrumental gestures. These parameters are non-exhaustive and relatively low-level, and it is in most cases possible to compute higher-level parameters as well.

In Table 1, every listed gesture has been categorized by the relevant temporal forms. As discussed above, it is noteworthy that any listed basic action can afford iterative gestures. Sequentially produced percussive gestures have been more widely used, but there is no reason why whistling or humming, for example, couldn't be produced iteratively as well. This also results in the fact that any gesture, when produced iteratively, can convey temporal parameters such as tempo, temporal deviation, acceleration slope etc. These continuous parameters can then be mapped to continuous commands and actions in the system. For example, monitoring the tempo of a clapping user can be used in a musical system to inform the tempo of the sound output as in [13]. As tempo is a continuous parameter, it could also be used to inform other than rhythmic functions in a system requiring continuous control. Considering richness of

| Sounding action (basic gesture) | Temporal form | Extractable parameters |
|---------------------------------|-----------------------------|---|
| hand clap | impulsive, iterative | clap type, patterns, tempo, temporal deviation, acceleration, volume |
| finger snap | impulsive, iterative | tempo, temporal deviation, acceleration, patterns |
| body tap | impulsive, iterative | type, tempo, temporal deviation, acceleration, patterns, volume |
| whistling | sustained, iterative | pitch, duration, slope, pattern, volume, tempo |
| vocal: humming etc. | sustained, iterative | pitch, duration, slope, pattern, volume, tempo, timbre |
| vocal: impulsive | impulsive, iterative | type, tempo, deviation, acceleration, patterns, volume |
| vocal: fricatives | sustained, iterative | type, duration, timbre, tempo, volume |
| vocal: vowels | sustained, iterative | pitch, type, duration, timbre, volume, tempo |
| breathing | sustained, iterative | type, duration, timbre, tempo, volume |
| footsteps | impulsive, iterative | type, tempo, patterns, volume |
| knocks and taps | impulsive, iterative | type, tempo, deviation, acceleration, patterns, volume |
| scratches | sustained, iterative | type, shape, duration, tempo, patterns, volume |
| blowing turbulence | sustained, iterative | duration, patterns, tempo, volume |

Table 1. A set of empty-handed (top) and instrumental (bottom) sonic gestures with different morphologies, including the basic parameters that can be extracted from each gesture. **Bold** face signifies that the gesture, temporal form, or parameter has been utilized in one or more of the interfaces and applications summarized in section 2.1, while the rest of the items are considered feasible in practice as well.

information, it should be noted that the iterative stream is still a result of concatenating basic gestures, and can incorporate informational parameters obtainable from the basic sounds, such as different hand configurations varying in the stream.

For all temporal forms, an obvious piece of information lies in the very occurrence of the sound-producing event. As exemplified above and in [12], for example individual hand clapping sounds can be detected to give discrete commands or trigger actions in computational systems. Also, the gesture type, e.g. hand configuration, can be recog-

nized to provide a set of discrete commands. While a computational system may only be reliable in recognition of a finite set of gesture types resulting in a finite gesture dictionary, concatenating basic gestures into patterns can broaden up the set of possible commands.

Sustained gestures always have a finite duration, which can be used as one computational parameter in a system. Pitched sustained gestures can convey information by pitch in various ways as discussed above. Short melodies can be mapped to discrete commands or functions, while gliding pitches can be used to tweak a continuous parameter in the system. In addition, the sound volume or its variation, and timbre can be tracked to enrich the information flow. Unpitched sustained gestures also are characterized by duration, and as shown by Harrison and Hudson [19], can be used to perform recognizable gestures based on temporal and timbral trajectories.

Looking at the parameter set, it can be argued that most of the sound-producing actions can be used for both discrete and continuous interactions. For impulsive actions, this usually requires performing an iterative gesture rather than one impulsive instance of the basic sound. For sustained actions to be used for discrete interactions, the solution is to consider them as objects rather than dynamic trajectories. This all boils down to mapping and selecting gestures with computable parameters that map well to the desired output parameters.

The gestures and their extractable parameters can also be studied in a multi-level hierarchy reflecting the complexity of the gestures and the extractable parameters at each level. This approach is presented in Table 2. At the lowest level, we have simple sonic gestures like individual hand claps, hums, and scratches. On the next level, here defined as the dynamic level, the body of the sounds can change its shape during the gesture, as for example in a whistle with a continuously changing pitch. Above these is the iterative level, i.e., all the iterative gestures. The highest level in this hierarchy is the compound level, which includes any combinations of the lower-level gestures, and can in theory provide an infinite group of potential gestures.

A possibility yet largely unexplored is to build interfaces around compound gestures and simultaneously occurring sonic gestures of different types. Compound sonic gestures could be gestural patterns, in which several gestures of different type follow each other (a finger snap followed by a whistle, for example). To facilitate richness of information at the interface, it would also be feasible to design interfaces where a sustained pitched gesture is used to control a continuous action, and an impulsive gesture that could overlap in the stream with the sustained gesture to indicate a discrete command. This approach would allow gestural multi-tasking.

Considering different sonic gestures, it is clear that they have different constraints in how they can be physically produced. Hand clapping or table tapping is easier to perform with fast tempos than finger snapping, for example. The same applies for sustained gestures, where for example the natural pitch range for humming is limited and even varies between different people, as shown by Sporka in

| Level of complexity | Impulsive | Sustained |
|---------------------|---|--|
| Compound | Any of the below and their combinations | |
| Iterative | Tempo, temporal deviation, acceleration/deceleration, patterns Example SGs: Walking with a constant tempo, clapping a pattern of different hand clap types | Tempo, temporal deviation, acceleration/deceleration, patterns, melodies Example SGs: Scratching the table in a repetitive motion, humming a melody |
| Dynamic | | Trajectories of changing pitch, timbre, volume, etc. Example SGs: Whistling with a rising pitch, scratching the table in an arc motion |
| Basic | Type, volume, timbre, direction Example SGs: Hand clap, finger snap | Type, pitch, duration, volume, timbre, direction Example SGs: Whistling with a constant pitch |

Table 2. A multi-level presentation of extractable informational parameters from impulsive and sustained sonic gestures (SGs). The levels indicate the complexity of the gestures (and the required processing algorithms). On the basic level, we find simple gestures like individual claps, snaps, and hums. The dynamic level does not exist for impulsive gestures in this case, as their body cannot be dynamically varied after the sound has been generated. The compound level is a placeholder for arbitrary combinations of the lower-level gestures.

[25], who found that the average comfortable pitch range is 12.7 semitones. Personal adjustability or system adaptability in fine-tuning the mapping can be useful in sonic interfaces.

There are also computational constraints to consider. For example, it is in general not feasible to implement a real-time algorithm reliably differentiating between an arbitrary number of impulsive gesture types. Therefore, the interface designer needs to select an optimal set of gestures for the task. If needed, the gesture typology can be extended with compound gestures or gesture patterns.

An interesting prospect in using sonic gestures lies in sound-based positioning of the sound-performing human. With an array of microphones, it is possible to detect the

direction, from which the sound arrives. While this was already demonstrated by two microphones in [11] for a single user, it is possible for example to separate the sound streams of different users with positioning information from a larger array.

5. DISCUSSION

It is without question that sonic gestures can bear lots of information useful in human-computer interaction. This is not to say that sound should or could always be used as the only input modality in interactive applications, but rather that the interaction could be enriched by more broadly acknowledging the use of sonic gestures as one input modality in the sensory fusion. Also, as sonic gestures are by nature typically embodied actions, they have potential in creating "natural" interactions. This, however, is dependent on finding suitable interaction primitives that result in aligned multisensory perception, as discussed in [14] in the context of continuous sonic interaction.

Designing interfaces around sonic gestures can be seen as closely related to the "traditional" design of sonic interactions. Indeed, for example basic design has been proved to be a usable tool for SID in designing sonic feedback [26, 14], and it can be hypothesized that similar techniques could be used to design interfaces with sonic input.

This study does not discuss the computational methods for information acquisition from sonic gestures. In general, it can be stated that an algorithm for detecting sonic gestures needs to be specialized into certain gesture types, for example classifying different impulsive gestures or tracking the pitch of humming. However, several algorithms suitable for sonic gestural interfaces are already available from different fields of study. For example in the field of music information retrieval, more and more focus has recently been put to implementing real-time algorithms for sound recognition, tempo and beat tracking, pitch tracking, etc. These tools can be efficient also in the acquisition of information from inherently non-musical sonic gestures, as exemplified for example in [22].

One important notion in discussing the utility of sonic gestures is to acknowledge their social acceptability. As it is generally understood that noise pollution is nowadays everywhere especially in urban life, do we want to add to the chaotic soundscape people interacting with their devices by clapping their hands or hissing through their teeth? This question is interesting and challenging for interface designers, who need to take into account the potential contexts where their designs are applied, and that a cultural change to facilitate the use of new interaction schemes ubiquitously takes time. It may be that sonic gesture interfaces have most applications in private conditions, or in social interaction applications where the users occupy the same space. Utilizing less intrusive sonic gestures and placing the microphone close to the sound production may provide a solution.

6. CONCLUSIONS

This study has demonstrated that sonic gestures, as sound-producing and information-bearing actions of a human, may be used in many ways to realize new kinds of interfaces for human-computer interaction, and that they are able to convey a very rich set of information. While some of the reviewed applications are musical, the gestures themselves are inherently often non-musical, at least until the context of interaction is introduced. Different sounds bear different kinds of information, which can be mapped in a desired way to the system output. While similar studies are known in the field of gestural interfaces, for sonic gestures a solid body of work demonstrating where different sonic gestures may be useful has not been previously presented.

An important aspect to consider when designing interfaces around sonic gestures is to aim for maximally natural and intuitive interaction. While it is entirely possible to derive continuous parameters from iterative gestures, for example, it does not mean that the mapping to any continuous output parameter is meaningful. To derive more comprehensive guidelines for facilitating the use of sonic gestures in the fields of HCI and SMC, one potential approach could be design patterns [27], capturing the use of sonic gestures in context to highlight good use scenarios, available computational techniques, and gestural relations. However, to date the body of work on sonic gestures for system input is not broad enough for deriving a comprehensive set of patterns.

An important prospect for future is to combine the presented gesture and parameter taxonomy with the physiological limitations of the production of different sonic gestures, e.g., what is the natural range of tempos for clapping and the natural pitch ranges in humming and whistling. This would underline the dynamic range of each gesture type and provide more guidelines for interface designers.

Acknowledgments

This work has been supported by the Graduate School at Aalto University School of Electrical Engineering, the Academy of Finland (project SCHEMA-SID), and the Finnish Foundation for Technology Promotion. The author would like to thank Dr. Cumhuri Erkut for his support.

7. REFERENCES

- [1] C. Cadoz and M. Wanderley, "Gesture-music," in *Trends in Gestural Control of Music*. Paris, France: IRCAM - Centre Pompidou, 2000, pp. 71–93.
- [2] E. Miranda and M. Wanderley, *New digital musical instruments: control and interaction beyond the keyboard*. Madison: AR Editions, Inc., 2006.
- [3] A. Jensenius, *Action-sound: Developing methods and tools to study music-related body movement*. Faculty of Humanities, University of Oslo, 2008.
- [4] D. Van Nort, "Instrumental Listening: sonic gesture as design principle," *Organised Sound*, vol. 14, no. 02, pp. 177–187, 2009.

- [5] R. Godøy and M. Leman, *Musical gestures: Sound, movement, and meaning*. New York: Taylor & Francis, 2009.
- [6] M. Chion, *Le Son*. Paris: Editions Nathan, 1998.
- [7] C. Cadoz, “Instrumental Gesture and Musical Composition,” in *Proc. Intl. Computer Music Conf.*, Cologne, Germany, 1988, pp. 1–12.
- [8] I. Ekman and M. Rinott, “Using vocal sketching for designing sonic interactions,” in *Proc. 8th ACM Conf. Designing Interactive Systems*, Aarhus, Denmark, 2010, pp. 123–131.
- [9] A. Dessen and G. Lemaitre, “Free classification of vocal imitations of everyday sounds,” in *Proc. Sound and Music Computing Conf.*, Porto, Portugal, 2009.
- [10] A. Sporcka, “Pitch in non-verbal vocal input,” *ACM SIGACCESS Accessibility and Computing*, no. 94, pp. 9–16, 2009.
- [11] S. Vesa and T. Lokki, “An eyes-free user interface controlled by finger snaps,” in *Proc. 8th Intl. Conf. Digital Audio Effects (DAFx)*, Madrid, Spain, 2005, pp. 262–265.
- [12] A. Jylhä and C. Erkut, “A hand clap interface for sonic interaction with the computer,” in *Proc. Conf. Human Factors in Computing Systems (CHI)*, Boston, MA, USA, 2009, pp. 3175–3180, presented in interactivity.
- [13] A. Jylhä, I. Ekman, C. Erkut, and K. Tahiroğlu, “Design and Evaluation of Rhythmic Interaction with an Interactive Tutoring System,” *Computer Music J.*, vol. 35, no. 2, pp. 36–48, 2011.
- [14] D. Rocchesso, P. Polotti, and S. Delle Monache, “Designing continuous sonic interaction,” *Intl. J. Design*, vol. 3, no. 3, pp. 13–25, 2009.
- [15] J. Bilmes, J. Malkin, X. Li, S. Harada, K. Kilanski, K. Kirchhoff, R. Wright, A. Subramanya, J. Landay, P. Dowden *et al.*, “The vocal joystick,” in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, Toulouse, France, 2006, pp. I–625–I–628.
- [16] S. Chanjaradwichai, P. Punyabukkana, and A. Suchato, “Design and evaluation of a non-verbal voice-controlled cursor for point-and-click tasks,” in *Proc. 4th Intl. Conv. Rehabilitation Engineering & Assistive Technology*, Las Vegas, NV, USA, 2010, pp. 48:1–48:4.
- [17] P. Hämäläinen, “Novel applications of real-time audiovisual signal processing technology for art and sports education and entertainment,” Ph.D. dissertation, Helsinki University of Technology, 2007.
- [18] A. Hazan, “Performing Expressive Rhythms with BillaBoop Voice-Driven Drum Generator,” in *Proc. 7th Intl. Conf. Digital Audio Effects (DAFx)*, Naples, Italy, 2004.
- [19] C. Harrison and S. E. Hudson, “Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces,” in *Proc. 21st annual ACM Symp. on User Interface Software and Technology*, ser. UIST ’08, 2008, pp. 205–208.
- [20] G. Wang, “Designing smiles iphone ocarina,” in *Proc. Intl. Conf. New Interfaces for Musical Expression*, Pittsburgh, PA, USA, 2009.
- [21] A. Jylhä and C. Erkut, “Inferring the hand configuration from hand clapping sounds,” in *Proc. 11th Intl. Conf. Digital Audio Effects (DAFx-08)*, Espoo, Finland, 2008, pp. 300–304. [Online]. Available: http://www.acoustics.hut.fi/dafx08/papers/dafx08_52.pdf
- [22] U. Şimşekli, A. Jylhä, C. Erkut, and A. Cemgil, “Real-Time Recognition of Percussive Sounds by a Model-Based Method,” *EURASIP J. Advances in Signal Processing*, 2011, special Issue on Musical Applications of Real-Time Signal Processing.
- [23] W. Gaver, “What in the world do we hear?: An ecological approach to auditory event perception,” *Ecological psychology*, vol. 5, no. 1, pp. 1–29, 1993.
- [24] J. A. Ballas, “Self-produced sound: tightly binding haptics and audio,” in *Proc. 2nd Intl. Conf. Haptic and Audio Interaction Design*. Berlin / Heidelberg: Springer-Verlag, 2007, pp. 1–8.
- [25] A. Sporcka, “Non-speech Sounds for User Interface Control,” Ph.D. dissertation, Faculty of Electrical Engineering, Czech Technical University, 2008.
- [26] K. Franinovic and Y. Visell, “Strategies for sonic interaction design: from context to basic design,” in *Proc. 14th Intl. Conf. Auditory Display*, Paris, France, 2008.
- [27] J. Borchers, “A Pattern Approach to Interaction Design,” *AI & Society Journal of Human-Centred Systems and Machine Intelligence*, vol. 15, no. 4, pp. 359–376, 2001.

IMPROVING PERFORMERS' MUSICALITY THROUGH LIVE INTERACTION WITH HAPTIC FEEDBACK: A CASE STUDY

Tychonas Michailidis
Birmingham Conservatoire
Birmingham City University
tychonas@me.com

Jamie Bullock
Birmingham Conservatoire
Birmingham City University
jamie.bullock@bcu.ac.uk

ABSTRACT

Physical interaction with instruments allows performers to express and realise music based on the nature of the instrument. Through instrumental practice, the performer is able to learn and internalise sensory responses inherent in the mechanical production of sound. However, current electronic musical input devices and interfaces lack the ability to provide a satisfactory haptic feedback to the performer. The lack of feedback information from electronic controllers to the performer introduces aesthetic and practical problems in performances and compositions of live electronic music.

In this paper, we present an initial study examining the perception and understanding of artificial haptic feedback in live electronic performances. Two groups of trumpet players participated during the study, in which short musical examples were performed with and without artificial haptic feedback. The results suggest the effectiveness and possible exploitable approaches of haptic feedback, as well as the performers' ease of recalibrating and adapting to new haptic feedback associations. In addition to the methods utilised, technical practicalities and aesthetic issues are discussed.

1. INTRODUCTION

This paper presents an overview of a study that investigates whether incorporating haptic feedback into musical input devices can result in creative musical outcomes for composers and performers working with computers and sensor-based technology.

Traditionally, instrumental performers require an intimate relationship with their instrument, developed through a long process of development and exploration of this bidirectional relationship [1]. This relationship creates a cause-and-effect feedback loop between the performer and instrument, which is constantly developed and adjusted while playing. The instrument reacts to the energy it receives from the performer by producing both, aural and haptic feedback. Through instrumental practice, the performer is able to learn and internalise these responses.

Copyright: © 2011 Michailidis et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Appraisal of current musical input devices and controllers shows that the received haptic feedback information is often limited, and does not provide the necessary level of *feeling* required from performers as happens with traditional instruments [3]. An experiment conducted by O'Modhrain and Chafe shows how force feedback improves the ability of the performer to control digital musical instruments such as the theremin [8]. Electronic controllers capture the performance gestures and process them through a computer that reacts to the prior decisions of the composer or programmer. The physical nature of such controllers or devices does not allow a bidirectional relationship with the performer, due to a physical decoupling of controllers and sound producing components. Furthermore, the mapping strategies employed between the controller and the audio processing can change arbitrarily, increasing the difficulty of constructing a reliable familiar feedback channel for performers.

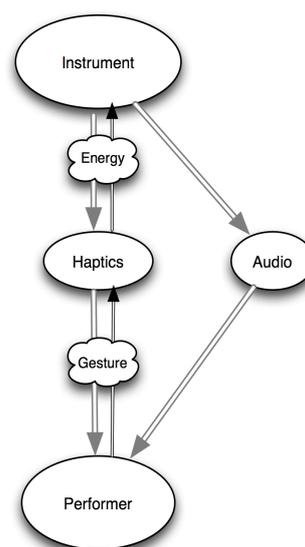


Figure 1. Shows the cause-and-effect feedback loop between the performer and the instrument.

2. CONTROLLING SOUND

Electronic controllers provide the means by which performers' physical gestures are converted into data accessible to use in conjunction with computers. Components like sensors, switches, faders and video cameras might be

used individually or in combination with each other. For example, the widely-used Nintendo Wii remote offers a combination of sensors, switches, an infrared camera and a wireless connection with which to transfer data to a computer. Bonger provides further discussion of the most commonly-used sensors for music applications [1].

In most cases, electronic controllers are made of plastic, a material that is unlikely to react to the energy provided by the performer. This raises concerns about the performers' experience and related feedback. Chu mentions additional concerns about computer-generated sound being disembodied from the physical object, problematising the formation and control of the sonic properties by the performer [4]. In addition, Tanaka suggests the importance of haptic feedback in creating music coherence in performances [9].

Complications arise upon considering the mapping relationship between the controller and the sound source. This significant aspect of electronic music has been addressed extensively by Hunt, Kirk, Miranda, and Wanderley [5]. Mapping strategies and the possibilities of sound control in real time introduce additional difficulties in the development and use of *controllers as instruments*. Looking at the mapping strategies and sonic possibilities, there are no conventions as to what electronic controllers can affect. However, this flexibility provides opportunities for composers to use the same controller over and over again with different sound results. Consequently, performers face a situation where the development of performance skills, based on the audible feedback, is very unlikely. The creators of such devices often perform with their custom made controllers because they are able to familiarise themselves most to the relationship between controller and created sound [1].

With traditional instruments, the laws of acoustics play a major role in regards to their construction, functionality and sound quality. The physical properties of the instrument, in relation with the aural and haptic feedback, allow detailed exploration of their sonic properties. Two main concerns emerge from this investigation of electronic controllers in music performances:

- The absence of haptic feedback encourages a situation where the performer is only able to have a passive understanding of the sound generated, and
- the constant remapping approaches that the performers experience do not contribute toward a deeper understanding of the relationship of gesture to sound.

These two situations greatly reduce the ability of the performer to effectively realise the musical requirements of the composer.

3. CASE STUDY

3.1 Hypothesis

It is common for composers to combine live electronics with other instruments to create their desired musical result. However, hardware and space requirements of such live electronics components create rehearsal diffi-

culties, especially if the performer does not have their own equipment for the electronics or is unfamiliar with the technology involved. As a result the electronic aspects of pieces receive limited rehearsal. The rehearsal time available for live electronic aspects can often be as little as 2-3 hours 'on the day'. This study will test if incorporating haptic feedback in performances can improve the overall control, perception and musicality of the electronics by instrumental performers—taking into account the limited amount of time available.

3.2 Method

This study is aimed towards a practical utilisation of live electronic performing practice through sensor technology via haptic feedback channels. Different qualitative methods, like interviews and discussions, were employed in this study to examine participants' performing experiences. Six trumpet players, divided into two groups of three, volunteered to take part in a series of semi-open interviews and performing tests. All participants, were undergraduates studying at the Birmingham Conservatoire (Birmingham, UK), were in different academic years, and of both classical and jazz backgrounds. They were from 19 to 22 years of age, and spent between 15 and 25 hours playing their instrument each week. None of them had any prior experience in performing with live electronics. This excludes the possibility of a priori knowledge from influencing the outcome of the study. Each interview, including the performing tests, was approximately one hour and thirty minutes long, after which each participant was compensated with £5. All interviews were recorded with their permission.



Figure 2. (Top) Inputs and outputs of the Arduino prototype box, (bottom left) glove with pressure sensors attached and (bottom right) vibrating motors with and without rubber shield.

3.3 Hardware Implementation

The prototype box, created by one of the authors, uses an

Arduino Diecimila¹ board, an open source prototyping device. The board is capable of receiving up to six analogue inputs and thirteen digital inputs. The thirteen digital inputs also serve as outputs, of which six can provide Pulse Width Modulation (PWM). Connectivity with a computer is through USB, allowing both power to the board as well as data transfer. The board is housed within a plastic box fitted with female mini-jack connections [see Figure 2].

A pressure sensor glove was created with three sensors attached to the fingertips. As an output source, the PWM function is used to individually control vibrating motors. All sensors and motors use an 1/8" jack adaptor to connect to the Arduino box. Rubber covers were attached to each motor to create a larger surface area as well as to protect them from damage the while in use. The three vibrating motors are attached on the left hand of the performer in different places wrist (inside), forearm (inside) and bicep (inside) [see Figure 4]. The placement was determined through experimentation with a trumpet player (who was not included in the study's participants) in order to ensure comfort, effectiveness, and recognisability of the vibrations produced. In addition, a microphone, sound card, laptop, and speakers were used.

Example 1



Example 2



Example 3



Example 4



Example 5



Example 6



Figure 3. Music excerpts composed for the performance test.

3.4 Methodology

3.4.1 Preliminary Interviews

The subjects were interviewed before and after a performance test. First, general questions were asked regarding the performance background of each participant, including the amount of weekly practice, how long they have played trumpet, the genre of music they usually perform, and if they play any other instruments. Following this were questions addressing their understanding of live electronics and computer music in general.

3.4.2 Performance Test

The performing portion of the study was divided into two tests, A and B, performing the six musical examples in each test. Both tests use the glove having the pressure sensors controlling the effects. Test A was indicated to be as a standard approach using live electronics while test B utilised haptic feedback. Group one played first the example with the standard approach and then all examples with the haptic approach. Group two performed first the haptic approach and then the standard approach [see Table 1].

| | | TEST | |
|---------|---|------|--|
| Group 1 | A | B | |
| Group 2 | B | A | |

Table 1. Shows the order of the tests for each group.

This enabled us to compare the result of adding haptic feedback to both new and previously-learned systems. The brief musical examples provide a range of musical variables, including articulation, note range, phrasing and dynamics [see Figure 3]. The tempo of the examples was unspecified, allowing for free interpretation, which was explicitly encouraged. The composition process was influenced from the trumpet fingerings, as they affected the relationship of the sensors by the notes being played. In example 4, the music requires the performer to use only fingers one and two that control the reverb and frequency shifting effects. In combination with the long notes and the absence of timing the performer is expected to concentrate on how the effect changes with the vibrating relationship. Example 3 was composed to examine how the vibrating functions might work in fast musical passages, and to test the performer's awareness of the vibration.

Max/MSP² programming environment was used for receiving sensor data and transmitting data to the vibrating

¹ www.arduino.cc/

² http://cycling74.com/

motors. Incoming sound was processed through Ableton Live³, modified by the values received from the pressure sensor glove. A one-to-one mapping was implemented between sensor input and sonic effect. Three different effects were used throughout the study. The participants wore the pressure sensor glove on their right hand, which also operated the trumpet's valves. The pressure sensor on the first finger correlated to the amount of reverb added, the second finger affecting frequency shifts, and the third finger controlling the amount of a chorus effect.

The vibrating motors also make use of a direct one-to-one mapping of input to output. In test B, where the haptic feedback layer was added, each sensor's data received from the glove correspond linearly to one vibrating motors. This relationship was explained to the participants as "the more you press, the more it vibrates". A calibration function was created to provide the maximum and minimum values received from each trumpet player before the tests began. This allows the individual calibration of the motors according to the pressure that was applied to each value from the performer. Sound received from the trumpet was monitored in the computer through the microphone. The performer controlled all the parameters of the effects in both tests. Each performing test lasted around 25 minutes, and included two play-throughs of each musical example.

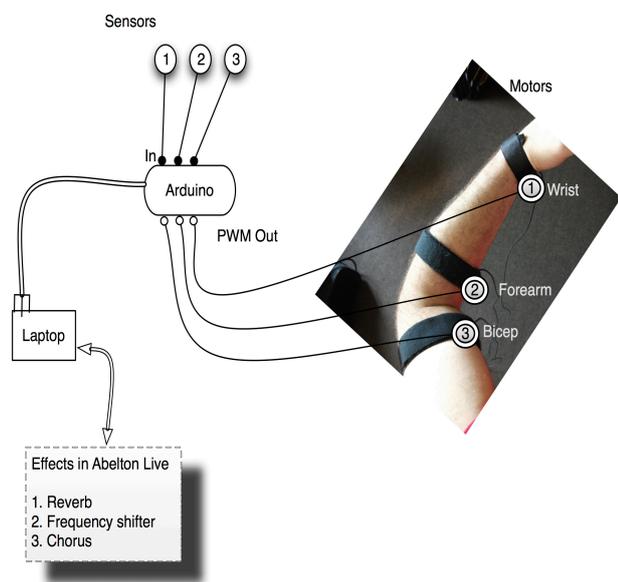


Figure 4. Overview of entire haptic feedback system.

3.4.3 Final interviews

The final set of questions was about the performers' understanding and experience they had while performing the two tests. Participants were asked a variety of questions, including: which system (that with or without haptic feedback) they would prefer to practice with; the difficulty of the two tests; the usability of the technology and hardware used; and their understanding of the sensors'

mapping to sound processing and vibrating feedback. In addition, they were asked to evaluate how fast they could adapt, if possible, to the artificially-created haptic relationship and which approach they would prefer to use in concerts.

4. RESULTS AND DISCUSSION

All six participants strongly agreed that haptic feedback created a more understandable relationship between their actions and the ensuing electronics. Apart from the ability to measure the amount of effects processing through vibrations, the participants all mentioned one essential difference between the two tests: the use of haptics allowed them to know definitively whether the electronic effects were active or not. This observation is important in that none of the performers had previous experience with live electronics. Additionally, the lack of audible confirmation about their action, in this case controlling the effects, was evened out through the haptic feedback channels. As mentioned previously, instrumental performers are used to the sensing feedback when they play their instruments. All performers mentioned that the calibrating effect was important in order to accommodate the amount of vibration received from the motors.

Four performers immediately became aware of the expressive possibilities while using the sensor glove with the vibrating feedback. They noticed that while the expression generally comes from the mouth, having the glove one is also expected to think about the pressure applied on the valves. One performer commented that "...expression comes from the mouth and you have to think not only how use the mouth but also the finger pressure to allow expressive changes of the sound". Another performer observed that "...with a bit of practice (using the vibrating motors) I can learn to manipulate it properly". They also noted that "you had something coming back, you could feel and you know physically if something was happening or not". One musician indicated that he could not hear the individual effects in test A but when he could feel it, in test B, he could then press the valves more-or-less accordingly. Another performer suggested the following during the interview: "From doing this now, I don't think that I will need additional practice time to get used to the motors. You could feel individually the effects through the vibrating feedback where in the run without the motors I was not able to know what was happening".

Four of the six performers indicated that, given the option, they would choose to use haptic feedback in the preparation and performance of live electronic works. To them, there was a substantial difference between test A and test B. Specifically, they mentioned the awareness of control they had through experiencing haptic feedback. With the remaining two performers, one preferred to focus only on the notated music having someone else to control all aspects of the computer processing. The remaining one had no distinct preference between the two systems.

³ <http://www.ableton.com/>

Furthermore, results of this study support the hypothesis that incorporating haptic feedback in live electronic performances may improve the overall control, perception and musicality of the electronics by instrumental performers. Even though we had two different groups with no prior experience in live electronics insufficient evidence was acquired to provide statistically significant results regarding the amount of improvement, control and perception in performances. However, qualitative responses elicited through interview give an early indication that the application of the haptic feedback system significantly improved the way the performers respond musically to the live electronics. The performers displayed an improved understanding of their actions in relationship with the pressure sensors and resulting sound produced. Consequently, our findings support the theory that haptic feedback can enhance musicians' expressivity in performances involving live electronic music.

The performers were questioned about how they perceived the basic understanding of the data flow from the controller, the sensor glove, to the resulting sound. Interestingly the performers having the haptic approach (test B) first, formed a clearer understanding overall. In addition, the participants were also asked if they thought that an understanding of the technology involved could improve their approach in performances. None of the performers were able to fully confirm this theory given the short amount of time available.

The results of this study suggest that haptic feedback has the ability to provide a framework for experimentation and improvisation with live electronics. After completing the tests, four of the performers asked us to further explore the haptic relationships. At one point, a performer realised that pressing the valves halfway through, the sensors were activated providing data to the computer. When asked, the performer mentioned that the vibrating feedback made him aware of the sensitivity of the pressure sensors. He was then able to slide between notes, using the half valve technique, creating interesting and unanticipated musical results with the effects. Another performer realised that it was not necessary to press the valves to activate the pressure sensors. Consequently, the performer was able to play with all three effects by pressing on the hard surface of the trumpet. However, this also meant the performer was only able to play notes within the trumpet's natural harmonic series.

Overall participants reported that the glove was comfortable enough and did not produce any problems while performing even in fast passages.

4.1 Future work

Future work will develop the technical aspect of the device used in order to minimise minor technical issues as well as increase functionality. On the current hardware an external driver should be added between the Arduino's PWM output and the vibrating motor in order to securely provide more power to the motors, as power management was not optimised in the current device. Additionally, a

new version is planned that includes a wireless Bluetooth connection as well as battery power [7]. The wireless hardware will provide flexibility of movement in performances with no need to wear the glove or attach the motors while on stage. The issue of latency between the sensors and the vibrating feedback should be explored further to minimise the response time as well as creating a more consistent device. However, it should be noted that none of the participants reported any noticeable latency problems when asked. Latency issues might be more apparent when vibrating feedback is used to indicate sections, cues or tempo in the score, as this would require temporal synchronisation to be accurate.

It is anticipated that using the sensor glove with the trumpet, composers will explore creative ways of musical expression in relationship with the fingering, the effects processing, and the haptic feedback provided to the performer. In addition, providing haptic feedback regarding electronic effects, composers can utilise vibration as a channel of communication between the performer and the computer to inform them of specific temporal cues, duration of events, functionality of running computer processes, as well as the positioning of electronic sound in space. Moreover, vibrating motors can be attached on more than one performer creating a haptic feedback network channel that can provide information to the performers independently or allow the exchange of information and gestures within the ensemble.

As discussed earlier, another study using the same hardware could examine the difference, if any, in the performing aspect of a piece with and without haptic feedback from the audience's perspective. Additionally, audio input could be utilised as another method to control the haptic feedback provided to the performers.

5. CONCLUSIONS

In this paper we have presented a study that attempts to establish whether adding haptic feedback to live electronics control improves the musicality of performer interaction. Our results suggest that adding haptic feedback to a glove-based controller can significantly improve a performer's understanding the relationship between control sensors and resulting sound produced. Additionally, the use of haptics suggested new musical possibilities not previously considered by the performers using non-haptic systems. Although using haptic feedback introduces an additional layer of complexity in live electronics systems, we consider it essential to pursue further research in this area so that standard methods of providing haptic feedback can be established. With haptic feedback in the control path, interaction is enriched allowing performers and composers to develop new relationships with live electronics practice.

Acknowledgments

The authors would like to thank Murphy McCaleb for his fruitful thoughts and discussions during the study and all the performers that took part in the interviews.

6. REFERENCES

- [1] Bonges, B. Physical Interfaces in the Electronic Arts Interaction Theory and Interfacing Techniques for Real-time Performance. In: M. M. Wanderley and M. Battier, eds. *Trends in Gestural Control of Music*. [CDROM], Paris: IRCAM Centre Pompidou. 2000.
- [2] Castagne, N. Cadoz, C. Florens, J. Luciani, A. Haptics in Computer Music: a Paradigm Shift. In *Proceedings of EuroHaptics, Munich Germany, June 5-7, 2004*.
- [3] Chafe, C. Tactile Audio Feedback. *Proceedings of the International Computer Music Conference (ICMC)*, 1993, pp.76-79.
- [4] Chu, L.L. Haptic Feedback in Computer Music Performance. *Proceedings of the International Computer Music (ICMC)*, 1996, pp. 57-58.
- [5] Hunt, A. and R. Kirk “Mapping strategies for musical performance”, in *Trends in Gestural Control of Music*, eds. M. Wanderley and M. Battier, IRCAM, Paris, 2000, pp. 231–258.
- [6] Miranda, E. R. and Wanderley, M.M., eds. *New digital musical instruments: control and interaction beyond the keyboard*. Middleton, Wisconsin: A-R Editions, Inc. 2006.
- [7] Modlier, P. and Myatt, T. Haptic Feedback for Improved Positioning of the Hand for Empty Handed Gestural Control. *Proceedings SMC'07, 4th Sound and Music Computing Conference, 2007*, Lefkada, Greece.
- [8] O'Modhrain, S. and Chafe, C. Incorporating Haptic Feedback into Interfaces for Music Applications. In *Proceedings of ISORA, World Automation Conference, 2000*.
- [9] Tanaka, A. Musical Performance Practice on Sensor-based Instruments. In: M. M. Wanderley and M. Battier, eds. *Trends in Gestural Control of Music*. [CDROM], Paris: IRCAM Centre Pompidou. 2000.
- [10] www.crackle.org/ (Michel's Waisvisz web site)

WHERE DO YOU WANT YOUR EARS? COMPARING PERFORMANCE QUALITY AS A FUNCTION OF LISTENING POSITION IN A VIRTUAL JAZZ BAND

Adriana Olmos

Centre for Intelligent Machines
McGill University
aolmos@cim.mcgill.ca

Paul Rushka

Schulich School of Music
McGill University
paul.rushka@mail.mcgill.ca

Doyuen Ko

Schulich School of Music
McGill University
doyuen.ko@mail.mcgill.ca

Gordon Foote

Schulich School of Music
McGill University
gordon.foote@mcgill.ca

Wieslaw Woszczyk

Schulich School of Music
McGill University
wieslaw@music.mcgill.ca

Jeremy R. Cooperstock

Centre for Intelligent Machines
McGill University
jer@cim.mcgill.ca

ABSTRACT

This study explores the benefits of providing musicians with alternative audio rendering experiences while they perform with a virtual orchestra. Data collection methods included a field study with a large jazz band and a pilot study in which musicians rehearsed using a prototype that presented two different audio rendering perspectives: one from the musician's perspective, and a second from the audience perspective. The results showed that the choice of audio perspective makes a significant difference in some musicians' performance. Specifically, for some musicians, e.g., lead trumpet players, an acoustically natural mix results in improved performance, for others, e.g., drummers, it was easier to play along with the artificial "audience" perspective. These results motivate the inclusion of a music mixer capability in such a virtual rehearsal scenario.

1. INTRODUCTION

Ensemble rehearsal is a demanding activity for musicians, one in which they must deal not only with the complexities of their own part, but also, coordinate with the performance of other musicians. It is challenging for many musical groups to find sufficient opportunities for the entire ensemble to practice together. This challenge led to our work on the Open Orchestra Project, which simulates the ensemble rehearsal experience, using *both* high-definition video and high-resolution audio, rendered from the perspectives of individual instrumentalists. In other words, the musician sees the conductor and relevant part of the orchestra on a panoramic video display, and hears the rest of the orchestra, with his or her own part removed. The result combines the experience of ensemble rehearsal with the convenience and flexibility of solo study.

In normal ensemble rehearsal and performance, musicians

see and hear the other instruments depending on their physical location within the orchestra. For example, in a large jazz band, a lead trumpet, surrounded by other trumpet players and positioned directly behind the trombones, primarily hears the brass section. The result, both in terms of relative loudness and arrival times of the sounds from the various instruments, is very different from the experience of a lead alto saxophone player, or for that matter, of an audience member. In the context of ensemble training with the Open Orchestra system, we consider whether it is better for the musicians to practice with the sounds of the other instruments reproduced in this natural manner, from their intended position, or as a more balanced mix, along the lines of that produced for a commercial recording. Specifically, we are interested in determining which option is preferred by the musician, which option is considered more realistic, and how this choice impacts the quality of the musician's performance.

Our initial hypothesis was that although lacking in the aesthetics of the audience experience, an audio image of the orchestra, rendered from the musician's individual perspective, would be the most desirable, since this provides the necessary audio cues to interact with one's closest orchestral neighbours, critical to an effective ensemble performance.

2. RELATED WORK

A variety of previous systems have been developed to present musicians with an experience of performing with an orchestra. One of the best known examples is perhaps "Music Minus One",¹ which consists of prepared recordings of a musical program from which an instrument or voice is missing. A musician may practice and learn the omitted performance, accompanied by the recording of the ensemble, much like karaoke systems. However, systems like this suffer from an absence of visual cues, and a limited control of the ensemble sound, providing only an audio image from a predetermined audience perspective. Another group of such systems supports real-time accompaniment

Copyright: ©2011 Adriana Olmos et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

¹ <http://musicminusone.com>

by synthesis [1, 2, 3], anticipating the performer's trajectory through a non-improvised musical piece.

Audio spatialization and sound image rendering have long been the subjects of intensive study [4]. Considerable effort has been devoted to software systems that simulate acoustic environments, based on psychoacoustic models related to the perception of sound sources by the human ear [5]. These models led to techniques for sound localization using headphones or a limited number of loudspeakers, typically exploiting interaural level differences (ILD), interaural time differences (ITD), and sound filtering techniques such as reverberation, to recreate the impression of distance and direction.

Numerous examples of such work can be found in interactive games, virtual reality, electroacoustic composition and audio conferencing [6, 7, 8]. Audio spatialization has also been employed in previous systems intended for musical practice and performance. For example, Schertenleib et al. [9] provided music amateurs the opportunity to conduct a group of musicians and produce a new kind of multimedia experience, rendering the orchestra from a central position, with attention to both visual realism and 3D sound rendering.

Other examples of spatialized audio rendering for immersive virtual environments include the work of Naef et al. [10], who optimized their system for rendering moving sound sources in multi-speaker environments using off-the-shelf audio hardware. Wozniowski et al. [11] proposed a framework for creation of perceptually immersive scenes, in which the entire environment functions as a rich musical instrument, with spatialization of sound sources an important element for musical applications. Somewhat closer to a real world reproduction, Martens and Woszczyk [12] accurately mapped and recreated nine virtual rooms in which Haydn's music would have been played. This work was carried out in the context of re-creating a small concert performance as it would have been experienced acoustically in the eighteenth century.

Musicians are influenced by both internal timing variances and external latencies while they try to coordinate their timing [13]. The former are attributed to performer anticipation and delays associated with expressive performance, errors in performance, and random timing variations due to physiological and biological constraints. External latencies are the result of audio propagation through the air as sound travels from the instrument to the performers' ears. Players in a small chamber ensemble typically experience such latency of 5-10 ms, whereas a double bass player could encounter latency of 30-80 ms in the sounds from the percussion section of a large orchestra (e.g., symphony). During their training, musicians develop various techniques that allow them to overcome these latencies, following or isolating certain sounds or instruments in order to coordinate the timing of the musical passage that they are performing. For a simple rhythmic clapping task, Chafe et al. [14] offer a detailed analysis of the effects of such latency. Given his central position with regard to the ensemble, the conductor's role in this process is thus to coordinate the musicians, adjusting not just the tempo, but

also ensuring a suitable balance between the different instruments.

Despite the large body of work in the domain of spatialized audio and its importance to music, little attention has been given to alternative audio experiences, or audio rendering perspectives for a musician sitting in a particular orchestral position. Specifically, there does not appear to be any prior work investigating the effects of audio perspective on the musician's performance, that is, whether it makes a difference if the sound is rendered "naturally" from the position of that performer, or from some other perspective such as that of the audience.

3. METHODOLOGY

We investigated the above question through both a field study in the context of orchestral rehearsal, as well as via an experimental study. The former involved observations and recording of the behaviour and actions of the orchestral participants within their work context, without interfering with their activities. These observations were complemented by exposing two of the musicians to an audiovisual "music minus-one" type of system, aiming to elicit conversations between the musicians and the design team. The experimental study involved a pilot experiment in which musicians were asked to rehearse with a prototype built to test two different audio rendering perspectives or conditions, one rendered from the musician's perspective, the other from the audience perspective.

Following a discussion of the musicians' preferences and the quality of their performance, as assessed by a big band Jazz conductor, we discuss the implications of these results in relation to our ongoing work on the design of a virtual orchestral rehearsal system.

3.1 Observing real and virtual rehearsal sessions

To support the exploratory nature of our initial research, a quick ethnographic model was employed in the early stages. This involved fly-on-the-wall [15] observations within a real scenario, and also employed a mock-up prototype of the system that allowed the musicians to rehearse with a recording from a previous rehearsal session. These observations were complemented with conversations with the conductor and musicians. The field study was carried out over a full three-month academic term with the McGill Jazz Orchestra I, an ensemble of 18 students, the most experienced jazz band in the Schulich School of Music. After the fly-on-the-wall observation sessions, our written notes were integrated into a presentation² to a user group in order to solicit their feedback. This user group consisted of conductors and professional musicians involved in teaching and mentoring activities at the university level. Although informal, this stage was valuable since it provided a general understanding of the orchestral rehearsal process at the outset of the study. Observations clarified initial assumptions regarding the importance of audio and visual cues and helped inform the design of the pilot experiment,

² <http://tinyurl.com/5wgm42b>

described in Section 3.2, intended to explore various audio rendering conditions.

The main findings from the early observations are now summarized. Students in the jazz band rehearsed in three modalities: on their own at home, in groups (depending on the instrument or musical voice they played), and with the full orchestra and conductor.

Quoting a trumpet player from the Jazz ensemble:

“[In the orchestra] I practice for listening... it is a team work. At home is more for learning the piece... If you focus and play while listening, all the music comes together...”

During the rehearsals, musicians add comments to their own music parts. These typically consist of descriptions or guidelines from the conductor’s feedback regarding how to play a music section. When the orchestra is learning a new piece, the conductor might choose to play back a recording of the music piece that they are about to learn. Other times, they start with a session of music reading, and together decide how to play or interpret certain difficult parts of the piece. Some of these instructions make it to the music part in the form of annotations, while others are simply memorized and indicated by the conductor’s gestures while performing.

There is no question that playing within the ensemble involves a team effort to interpret the piece as a whole. Audio and visual cues are important elements of the ensemble’s rehearsal dynamics. As part of our early observations, we wanted to expose the musicians to the experience of rehearsing with a playback of their previous rehearsal session. This was done in part to prompt them to express their thoughts regarding the concept, as well as to elicit feedback about potential future improvements based on an early prototype.

Although initially skeptical, two of the musicians agreed to spend an hour of rehearsal time with a playback of their previous rehearsal session. To their surprise, the experience proved to be much better than they had expected; significantly, they were able to “pause” the conductor, assimilate his feedback, and repeat a section of music, incorporating the guidelines or instructions provided into their practice. Conversely, during actual “live” rehearsals, these capabilities are not possible.

Of direct relevance to our question of rendering perspective, the musicians immediately identified (by ear) the position from which the recording was made and were able to articulate the differences from what they would normally experience in real life, e.g., hearing more of the lead trumpet. The students also expressed an interest in hearing how they sound with the whole band from an audience perspective, which is not possible from their position in the ensemble.

3.2 Pilot study: comparing two audio image conditions

Based on the observations from the field study, above, we designed our pilot study to address the following questions:

1. While rehearsing with a high-fidelity simulator, which audio image is preferred by the musician, an egocentric perspective or one rendered from the audience position, and what accounts for this preference?
2. Would the preferred audio image be regarded as the most realistic, i.e., in relation to a real-world orchestral environment?
3. How does the choice of audio image rendering affect the performance of the musician?

The pilot study was conducted with eight jazz musicians, four from the McGill Jazz Orchestra I, who played in the recordings used in the prototype, and the remainder from other jazz bands. Both groups consisted of trumpet, trombone, sax and drums players. All the participants were enrolled in a university music program at either the Masters or undergraduate level.

The musicians were exposed to two conditions: an unmodified binaural recording, captured from the musician’s perspective, and an “audience” mix, equivalent to that of a central audience member’s perspective. The music piece used in this experiment was a recording of the McGill Jazz Orchestra I in Tana Schulich Hall, playing Nestico’s “Basie Straight Ahead”. Multiple binaural recordings were made using a Neumann KU 100 dummy head with binaural stereo microphones. Each of the musician’s position (lead trumpet, lead trombone, lead alto (sax) and drums) was substituted, one at a time, during the band’s performance. A Sony PMW-EX3 HD camcorder was located beside the dummy head, facing the conductor, to record the video from the same musician’s perspective. For the generation of the audience mix, close microphone placement on all the main orchestra sections (trumpet, trombone, sax, bass, and main) allowed for independent control of the balance of various sections. The mix was created by a sound recording engineer, aiming to produce an audio image from a central audience position, similar to the conductor’s perspective. The audio mixes and video were synchronized manually using Final Cut Pro by examining the onsets of the audio waveforms recorded both by the built-in camera microphone and separately by microphones covering the instrument sections. The synchronized audiovisual content was then validated by the conductor of the jazz band.

3.2.1 Experimental design

The musicians were asked to rehearse with the orchestral simulator, which was similar to the one presented in Figure 1. The experimental sessions lasted approximately 80 minutes, including a break of 10 minutes at the half-way point. Rehearsals were carried out in four blocks of two trials (eight trials in total). For each block, musicians played the entire song twice, once with the binaural recording and once with the audience mix, with the order of presentations balanced across blocks, conditions, and musicians. The musicians were not informed as to which recording was presented at each trial. At the end of each block, the musicians were given the following questionnaire:

1. Which audio track allowed you to perform to the best of your ability?



Figure 1. Trumpet player rehearsing with an early prototype of the Open Orchestra Project

2. Which track felt more realistic, as related to your experience with a real orchestra?
3. Rate the selected track in terms of realism on a scale of 1 (unrealistic) to 5 (identical to a real orchestra).

4. ANALYSIS AND FINDINGS

The responses to the questions above, along with unsolicited comments, were then analyzed.³ In addition, audio recordings of the musicians' rehearsals were evaluated by a conductor who was not involved in the original performance.

4.1 Musicians' perceptions

Overall, musicians preferred to rehearse with the binaural recording in 20 of the 32 blocks across all the subjects, but this trend was not statistically significant ($\chi^2(1) = 2.000$, $p = 0.15$). Independently of which audio condition was chosen, musicians considered their preferred choice to be the most realistic ($\chi^2(1) = 21.125$, $p < 0.0001$). This can be observed in Figure 2, where the proportions of binaural preference are presented in response to the first two questions from the questionnaire above. Accordingly, musicians rated the chosen audio recording similarly in terms of realism, regardless of whether it was the binaural (Mean = 3.5, SD=0.49) or the audience mix (Mean=3.6, SD=0.43). This finding could be explained through the concept of processing fluency, which relates to the ease with which information is processed in the mind. Research in psychology has shown that processing fluency influences different kinds of judgments. For instance, perceptual fluency contributes to the experience of familiarity and positive affect [16].

4.2 Expert review

All of the trial recordings were evaluated by a conductor naive to the experimental set up and audiovisual condi-

³ The data from these sessions is available from <http://tinyurl.com/2fnp9ob>.

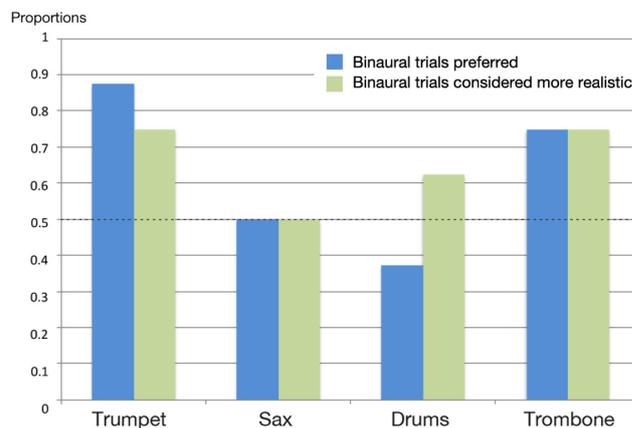


Figure 2. Responses across musicians for preference and perceived realism of the binaural condition.

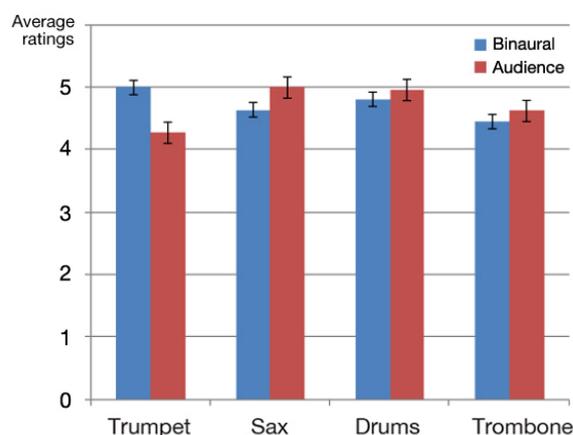


Figure 3. Average ratings per instrument for the binaural and audience audio mix. Error bars indicate the standard error.

tions. The conductor was asked to assign a rating from 1 to 5 to each of the parameters of articulation, time feel, pitch (intonation), note shape (inflection), sound quality, dynamic contrast, and rhythm. These parameters were selected based on various conversations with the conductor of the McGill Jazz Orchestra I. To avoid bias, the recordings were rated in a randomized order.

A total of 17 out of the 32 blocks were assigned higher average ratings when performing with the binaural recording, but the difference was not statistically significant. Refining the analysis by instrument can be more instructive. As seen in Figure 3, the average performance of the trumpet players proved to be significantly better when rehearsing with the binaural mix ($t(7) = 2.938$, $p = 0.013$). The Bonferroni corrected p-value was marginally non-significant. However, the result was consistent with the general preference of the trumpet players for the binaural recording, as indicated in Figure 2.

4.3 Discussion

While the limited number of participants in this pilot study precludes us from stating any strong conclusions, some interesting trends are suggested by the observations. For the lead trumpet in a jazz band, the binaural audio image seems to make a significant difference and helps improve the performance. However, for the lead alto sax players, the lack of clear preference between the binaural and audience mix might be explained by the fact that their sitting position in the band is central, adjacent to the conductor. From that position, they typically receive a more balanced sound, similar to what the audience would hear. The drummers sit to the side of the brass section and mostly hear their own sound, relying on the conductor's gestures for guidance. As one drummer player mentioned "as long as I can hear the bass, I am happy." This might also explain the lack of a clear preference between the binaural and audience mix, given that in both, the bass can be heard clearly. Nevertheless, the drum players were able to identify the audience recording, labelling it a bit less realistic from the experience encountered in a typical orchestral situation because they could hear the band more clearly. Despite its artificiality, these musicians noted that it was easier to play with the audience mix. On the other hand, trombone players were comfortable with either recording but indicated a preference to hear a bit more of the brass section, commensurate with their natural experience.

It is important to mention that the audio conditions were independent of the fixed video display perspective for each instrument. In other words, the video content rendered to the musicians was always acquired from the perspective of that instrumentalist, regardless of whether the binaural or audience audio mix was used. One could argue that this performer-centric video perspective might have influenced the results in favour of selecting the binaural mix as the most realistic, since this is the one with which it was congruent. The motivation for using this same video perspective regardless of the audio environment was to ensure that the musician had visual access to the gestures from the conductor, which were unavailable from the audience perspective. Without such a view of the conductor, the experience would almost certainly have felt less natural. It would be interesting to consider a further audio-only experiment of the two conditions to remove the potential confound of audiovisual congruency from these results.

In any case, the results from this study suggest that the value of providing a dedicated audio image to the musicians, rendered from their own placement within the orchestra, is dependent on the individual instrument. Perhaps even more importantly, providing a mixer capability appears to be desirable. This mixer would provide, as a starting point, a default audio setup that resembles the rendering from the position of the given instrumentalist, but allowing fine tuning of what the musician hears in an ensemble.

5. SUMMARY AND FUTURE WORK

The body of work presented here investigated the idea of providing musicians with different audio experiences while performing with a virtual orchestra. Two audio rendering perspectives were presented to a group of eight jazz musicians, one at the time, sitting in a particular orchestral position (lead trumpet, lead alto sax, lead trombone, drums). We investigated the effects on the musician's performance while rehearsing with both an audio perspective provided "naturally" from the position of that performer and that from the audio perspective of an audience member.

While there was a slight preference towards the audio experience rendered from the musician's perspective, this was not significant across all instruments tested. However, there was a significant difference in the performance by the lead trumpet players while rehearsing with the audio rendered from their perspective. These results seem to suggest that the value of providing a dedicated audio image to the musicians is dependent on the individual instrument position. Our ongoing work is examining the customization of these audio parameters for a given musician based on recommendations from a mentor or conductor. We expect that this approach will foster an interesting learning environment in which the musician could practice and improve his skills while performing within an orchestra context.

Future work will involve a larger study including other genres of music and expanding the number of participants, as well as the number of expert reviewers or conductors assessing performance of the musicians. As noted above, we are also interested in the experimental outcome of an audio-only presentation of the two renderings, without the potential confound of a video perspective that is congruent with only one of the conditions.

6. ACKNOWLEDGEMENTS

The research described here was funded under a Network Enabled Platforms (NEP-2) program research contract from Canada's Advanced Research and Innovation Network (CANARIE). The project is being developed in collaboration with colleagues at the Centre for Interdisciplinary Research in Music, Media and Technology (CIRMMT) at McGill University. In particular, the authors would like to thank Antoine Rotondo and Nicolas Bouillot for their help in the set up of the recording sessions, Mick Wu for valuable discussions on the experimental design and analysis, the CIRMMT technical staff for their continual assistance, and John Roston, for his important role in this work.

7. REFERENCES

- [1] R. Christopher, "Demonstration of music plus one: a real-time system for automatic orchestral accompaniment," in *Proceedings of the 21st National Conference on Artificial Intelligence*. AAAI Press, 2006, pp. 1951-1952.
- [2] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," in *Proceedings of the International Computer Music Conference*, 1984, pp. 193-198.

- [3] B. Vercoe, "The synthetic performer in the context of live performance," in *Proceedings of the International Computer Music Conference*, 1984, pp. 199–200.
- [4] J. Blauert, *Spatial Hearing: The Psychophysics of human sound localization*. The MIT Press, 1997.
- [5] J. Chowning, "The simulation of moving sound sources," *JAES*, vol. 19, no. 1, pp. 2–6, 1971.
- [6] G. Walker, J. Bowskill, M. Hollier, and A. McGrath, "Telepresence: Understanding people as content," *Presence: Teleoperators and Virtual Environments*, vol. 9, no. 2, pp. 119–136, 2000.
- [7] R. Kilgore and M. Chignell, "The vocal village: Enhancing collaboration with spatialized audio," in *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*. Washington, DC, USA: AACE, 2004, pp. 2731–2736.
- [8] W.-G. Chen and Z. Zhang, "Highly realistic audio spatialization for multiparty conferencing using headphones," in *IEEE International Workshop on Multimedia Signal Processing*, 2009.
- [9] S. Schertenleib, M. Gutierrez, F. Vexo, and D. Thalmann, "Conducting a virtual orchestra," *IEEE Multimedia*, vol. 11, no. 3, pp. 40 – 49, 2004.
- [10] M. Naef, O. Staadt, and M. Gross, "Spatialized audio rendering for immersive virtual environments," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. ACM, 2002, pp. 65–72.
- [11] M. Wozniowski, Z. Settel, and J. R. Cooperstock, "A framework for immersive spatial audio performance," in *New Interfaces for Musical Expression*. IRCAM - Centre Pompidou, 2006, pp. 144–149.
- [12] W. Martens and W. Woszczyk, "Virtual acoustic reproduction of historical spaces for interactive music performance and recording," *Acoustical Society of America*, vol. 116, no. 4, pp. 2484–2485, 2004. [Online]. Available: <http://link.aip.org/link/?JAS/116/2484/4>
- [13] C. Bartlette, D. Headlam, M. Bocko, and G. Velickic, "Effect of network latency on interactive musical performance," *Music Perception*, vol. 24, pp. 49–59, 2006.
- [14] C. Chafe, J.-P. Cécères, and M. Gurevich, "Effect of temporal separation on synchronization in rhythmic performance," *Perception*, vol. 39, no. 7, pp. 982–92, 2010.
- [15] IDEO, *IDEO Method Cards: 51 Ways to inspire Design*. William Stout Architectural Books, 2003.
- [16] R. Reber, P. Winkielman, and N. Schwarz, "Effects of perceptual fluency on affective judgments," *Psychological Science*, vol. 9, pp. 45–48, 1998.

THE EYEHARP: AN EYE-TRACKING-BASED MUSICAL INSTRUMENT

Zacharias Vamvakousis
Universitat Pompeu Fabra
Roc Boronat 138
08018 Barcelona, Spain
zackbam@gmail.com

Rafael Ramirez
Universitat Pompeu Fabra
Roc Boronat 138
08018 Barcelona, Spain
rafael.ramirez@upf.edu

ABSTRACT

In this paper we present the EyeHarp, a new musical instrument based on eye tracking. The EyeHarp consists of a self-built low-cost eye-tracking device which communicates with an intuitive musical interface. The system allows performers and composers to produce music by controlling sound settings and musical events using eye movement. We describe the development of the EyeHarp, in particular the construction of the eye-tracking device and the design and implementation of the musical interface. We conduct a preliminary experiment for evaluating the system and report on the results.

1. INTRODUCTION

Traditionally, music performance has been associated with singing and hand-held instruments. However, nowadays computers are transforming the way we perform and compose music. Recently, music performance has been extended by including electronic sensors for detecting movement and producing sound using movement information. One early example of this new form of music performance is the theremin and terpsitone [1]. More recent examples of new music performance paradigms are systems such as The Hands [2] and SensorLab [3]. The creation of these kinds of musical electronic instruments opens a whole new door of opportunities for the production and performance of music.

Eye tracking systems provide a very promising approach to real-time human-computer interaction (a good overview of eye tracking research in human-computer interaction can be found in [4]). These systems have been investigated in different domains such as cognitive psychology where eye movement data can help to understand how humans process information. Eye tracking systems are also important for understanding user-device interaction and to allow physically disabled people to communicate with a computer using eye movements.

In this paper, we present the EyeHarp, a new music instrument based on eye tracking. We have built a low-cost tracking device based on the EyeWriter project [5] and im-

plemented various musical interfaces for producing sound. The resulting system allows users to perform and compose music by controlling sound settings and musical events using eye movement.

The rest of the paper is organized as follows: Section 2 describes the background to this research. Section 3 presents the EyeHarp, in particular it describes the construction of the eye-tracking device, the design and implementation of the musical interface, and the evaluation of the system. Finally Section 4 presents some conclusions and future work.

2. BACKGROUND

2.1 Eye tracking systems

Several approaches for detecting eye movement have been proposed in the past. These have included electrophysiological methods [6,7], magnetic search coil techniques [8], infrared corneal reflectance and pupil detection methods.

Electrophysiological methods involve recording the difference potentials generated between electrodes placed in the region around the eyes. However, this method has been found to vary over time, and is affected by background activation of eye muscles [7]. The disadvantages of search coil systems are that its use involves quite invasive procedures, and it relies on expensive hardware (i.e. around US\$40,000).

In recent years video-based eye movement detection has gained popularity due to the fact that it offers a solution to some of the limitations of other methods. For instance, it allows reliable tracking of the pupil as well as tracking of the iris as it rotates torsionally around the optic axis [9,10] at rates of up to 250 frames per second. However, one limitation of this type of system is the need for greater intensity of infrared illumination to allow adequate passing of light from the eye to the camera sensor.

Combined pupil detection and corneal reflection techniques are becoming more and more popular lately for interactive systems. The reason is that with this combined method the head of the user does not have to be fixed.

2.2 Eye-tracking-based music systems

The first system using eye tracking devices to produce music in real time was proposed by Andrea Polli in 1997 [11]. Polli developed a system which allowed performers to access a grid of nine words spoken by a single human voice

by making saccadic to nine different directions. After trying different artistic implementations Polli concluded that improvising with the eye-tracking instrument could produce the same feeling for the performer as improvisation with a traditional instrument [11].

In 2001 she performed “Intuitive Ocusonics”, a system for sound performance using eye tracking instruments to be performed live. Instruments were played using distinct eye movements. Polli’s compositions responded to video images of the eye, not specifically the pupil center which are parsed and processed twelve times per second using the STEIM’s BigEye software (www.steim.org). With this technology it is impossible to calibrate the pupils position to the computer screen coordinates, thus the user does not have precise control of the system.

Hornof et al. [12] propose a system based on a commercial eye tracking system, the LC Technologies Eye-gaze System, which provides accurate gazepoint data using the standard pupil-center corneal-reflection technique. In the system, the coordinates of the user’s gaze are sent to MAX/MSP for generating sound. They study both the case of using fixation detection algorithms for choosing an object and the raw data from the eye tracker. When trying to implement an eye-piano they report that the musicians that tried the system preferred to work with the raw data instead of a dispersion-based fixation-detection for playing the notes. The problem with fixation detection is that it reduces the temporal control, which is very critical in music. A velocity-based fixation-detection algorithm is suggested instead. They do not consider other techniques, such as blink detection, as a method for choosing objects. The authors consider designing more interactive tools using Storyboarding. The performer moves an eye-controlled cursor around on the screen, and makes the cursor come into direct visual contact with other visual objects on the screen, producing a visual and sonic reaction. The user interacts with objects that appear on the screen, through a series of interaction sequences (like a scenario).

Hornof and Vessey in a recent technical report evaluate four different methods for converting real-time eye movement data into control signals (two fixation based and two saccade-based methods). They conduct an experiment comparing the musicians’ ability to use each method to trigger sounds at precise times, and examined how quickly musicians are able to move their eyes to produce correctly-timed, evenly-paced rhythms. The results indicate that fixation based eye-control algorithms provide better timing control than saccade based algorithms, and that people have a fundamental performance limitation for tapping out eye-controlled rhythms that lies somewhere between two and four beats per second [13]. Hornof claims in [12] that velocity-based (as opposed to dispersion-based) fixation-detection algorithms work better for rhythmic control with the eyes. Fixation-detection algorithms typically employ a minimum fixation duration of 100 ms which would impose an upper bound of ten eye-taps per second.

Kim et al. [14] present a low cost eye-tracking system with innovative characteristics, called Oculog. For selecting objects, blink detection is implemented. The data from

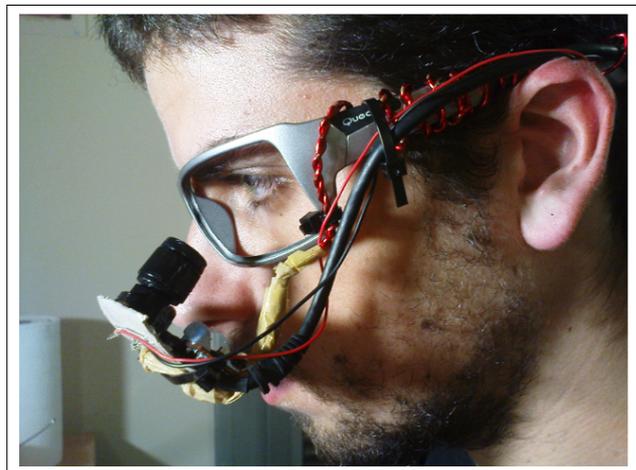


Figure 1. The PlayStation Eye digital camera is modified so as to be sensitive to infra-red light and mounted along with two infra-red leds on a pair of sun-glasses.

the eye tracking device are mapped to PureData for generating and interacting with four sequences. In their user interface the performer’s field of vision is divided into four discrete quadrants. The direction of eye movement detected by the Oculog camera software is encoded as a combination of horizontal position (pitch) and vertical position (velocity): pitch 0 is produced by looking to the extreme left, note number 127 to the extreme right; velocity 0 is produced by looking down, velocity 127 by looking up. Assigned to each quadrant is a real-time tone generator. Each tone generator is driven by a cyclic sequence. Oculog also detects torsional movement of the eye, but this is not mapped to any control feature. The authors claim that eye tracking systems are appropriate for micro-tonal tuning (they used a 15 note scale).

3. THE EYEHARP

3.1 Eye tracking device

There are a number of commercial systems available specifically designed to enable people to communicate using their eyes. However, these systems are expensive, costing in the range of US\$20,000. In order to create a reproducible system we decided to make the most simple and inexpensive eye-tracking head-set possible. We built our own eye tracking system based on the EyeWriter project [5]. Thus, the resulting system emphasizes low-cost and ease of construction and as a consequence has several limitations such as robustness and appearance. Figure 1 shows the eye tracking device used in this work.

In order to read the input from the eye tracking device, we have used the libraries developed in the EyeWriter project. The eye-tracking software detects and tracks the position of a pupil from an incoming camera or video image, and uses a calibration sequence to map the tracked eye/pupil coordinates to positions on a computer screen or projection. The pupil tracking relies upon a clear and dark image of the pupil. The eye tracking device includes near-infrared leds to illuminate the eye and create a dark pupil effect.

This makes the pupil much more distinguishable and, thus, easier to track. The software dealing with the camera settings allows the image to be adjusted with brightness and contrast to get an optimal image of the eye. When initializing the system, calibration takes place displaying a sequence of points on the screen and recording the position of the pupil at each point. The user focuses on a sequence of points displayed in the screen presented one by one. When the sequence is finished, the collected data are used to interpolate to intermediate eye positions.

3.2 Music interface

The ultimate goal of this project is to create a real musical instrument with the same expressive power as traditional musical instruments. The implemented instrument should be suitable for being used as a musical instrument for performing in a band, as well as a standalone composition tool. The following decisions have been taken in the EyeHarp design:

- More than one different layer should be available. One of them could be used for building the rhythmic and harmonic musical background, and another for playing accompanying melodies on top of the musical background.
- The performer should be able to control in real time the rhythmic, harmonic and melodic aspect of his/her composition, as well as to control the timbre of the instrument. The instrument's timbre is determined by having control over (i) the spectral envelope, (ii) the attack-decay time of the produced sound. In addition, the performer should have control over the articulation and other temporal aspects of sound such as glissando and vibrato.
- The buttons on the screen for playing a note should be big enough in order to reduce the possibility of playing neighbor notes, due to errors of the eye tracking system. To save space and avoid dissonant notes the produced instrument should be diatonic (like e.g. the harmonica). The user should be able to determine the musical mode while performing.
- Temporal control in music is crucial. This is why we should avoid using blink detection or fixation detection algorithms for playing real-time melodies. Music should be controlled by making use of just the user's gaze. Thus, the process of designing an eye tracking musical instrument is similar to designing an instrument in which the input is a pencil (eye gaze) drawing on a paper (screen), where the pencil should always be in touch the surface of the paper. Consequently the performer should be able to play every pair of notes with a straight saccade eye movement without activating any other note. This would allow working with the raw data of the eye tracker and skip the use of any fixation detection algorithm that would increase the response time of the instrument [12].

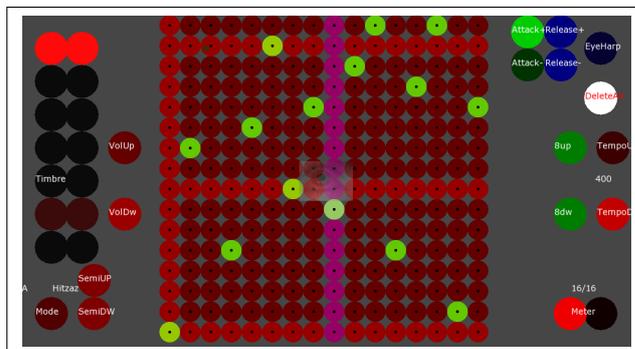


Figure 3. The EyeHarp Melodic Step Sequencer. Time Signature 16/16

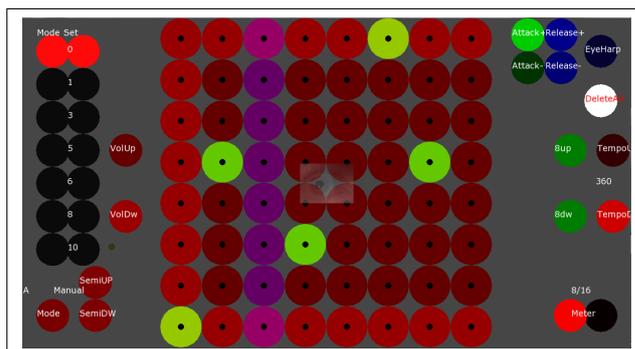


Figure 4. Setting the musical mode manually. In this case [0,1,3,5,6,8,10] corresponds to the mixolydian mode.

3.2.1 The EyeHarp Melodic Step Sequencer: Building the harmonic and rhythmic background

Various interfaces based on step sequencers are available in different environments. Two commercial examples are the Tenori-on [15] and Max For Live Melodic Step Sequencer [16]. The EyeHarp Melodic Step Sequencer is implemented using similar ideas (see Figure 2).

In the center of the screen there is a small transparent section which shows an image of the eye as captured by the camera in real-time. This is crucial for live performances, as it helps the audience to correlate the eye movements to the produced music. A small green circle indicates the user's detected gaze point. Each circle corresponds to a note. A note is selected when the user remains looking at it for more than one second. When a note is active, the color of the corresponding circle is green. Only one note can be selected for each step of the sequence. To deactivate a note the user has to look at it again for more than a second. At the center of every circle, each of which corresponds to a note, there is a black dot which helps the user to look at the middle of each circle. In every column we have the notes of the selected key with their pitch rising with direction from down to up.

The purple line in Figure 2 is moving from left to right with a speed related to the selected tempo. When the line hits one of the green circles, the corresponding note is played. So in this grid, in the horizontal dimension we have time and in the vertical dimension pitch (down→ low pitch, high→ high pitch). The horizontal brighter lines are for

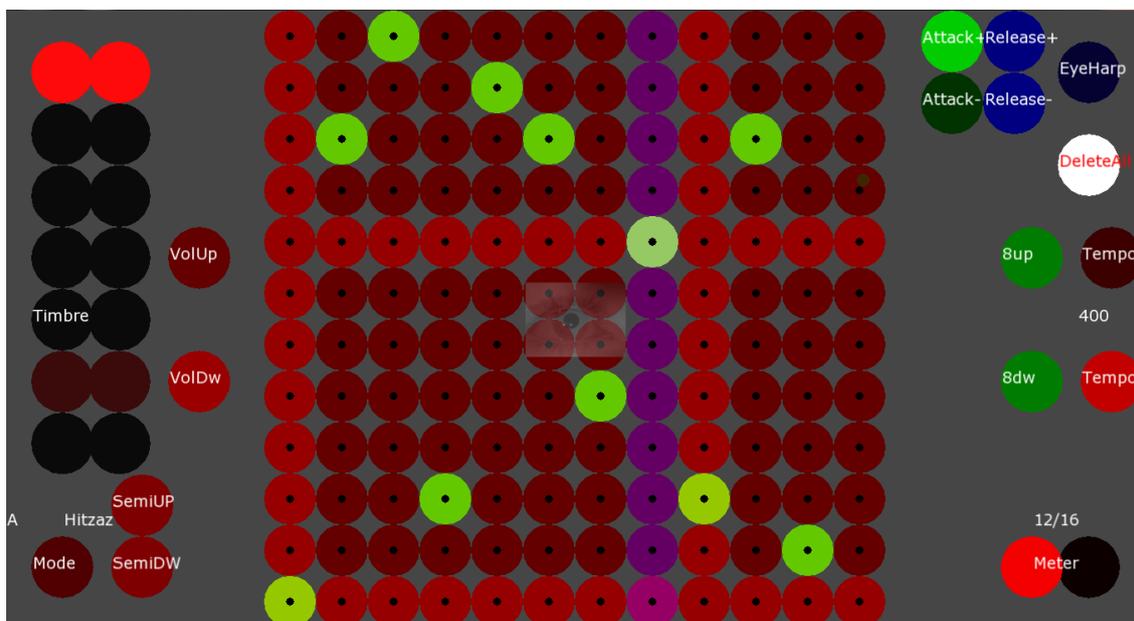


Figure 2. The EyeHarp Melodic Step Sequencer. Time Signature 12/16

helping the user to understand where a new octave starts (every 5 or 7 notes depending on the mode) The vertical brighter lines repeat every eight time steps and help to visualize beats.

On the left and right of the "score" region described above, there are various buttons (i.e. circles) affecting different sound and musical aspects of the composition. On the left of the score region there are two circle buttons for controlling the volume. Again a time threshold is used for triggering a volume change: every 0.25 seconds that the user keeps looking at the "volumeUp" button, the volume is increased one step. The color of the corresponding button gets brighter or darker corresponding to its value. When the volume reaches its maximum value the "VolumeUp" button is bright red and "VolumeDown" button is black. That way the user has feedback about when a control parameter has reached its minimum or maximum value. The same applies to most of the circle buttons for controlling all different input variables: "SemitoneUp" - "SemitoneDown", "AttackUp" - "AttackDown", "ReleaseUp" - "ReleaseDown", "TempoUp" - "TempoDown", "OctaveUp" - "OctaveDown", "MeterUp" - "MeterDown". The "MeterUp" - "MeterDown" buttons change the dimensions of the sequencer's grid. That way the time signature of the composition changes as well (ranges from 1 to 64).

The two columns of circles at the left of the screen are for controlling the amplitude of each of the seven first harmonics of the synthesized sound. The left column is for decreasing the contribution of each harmonic and the right for increasing it. The "mode" button is for switching between different musical (scale) modes. The available modes are the major, Hitzaz¹ and pentatonic. The "mode" can also be set to "manual". In this case the buttons that nor-

¹ Hitzaz is a mode used in Eastern music (e.g. Greece, Turkey and some Arabic countries) and flamenco music

ally were for setting the amplitude of the harmonics can be used for setting the musical mode manually: each note of the scale can be assigned to any semitone. For example in Figure 4 the mode is set manually to the mixolydian. Transposition to all the different semitones is available as well. Finally "DeleteAll" sets all the notes to inactive. The "EyeHarp" button is for switching to the EyeHarp layer for playing a melody on top of the composed loop.

The EyeHarp Step Sequencer is not designed for playing real time melodies, but for building the harmonic and temporal background of the composition. The decisions mentioned at the beginning of this section apply mostly to the next proposed layer for playing real time melodies.

3.2.2 The EyeHarp

Playing Melodies in Real Time The EyeHarp interface was designed having in mind that it can be controlled with or without a gaze fixation detection algorithm. A velocity based fixation detection algorithm can be optionally activated. The velocity is computed by two successive frames and is given by the equation:

$$Velocity = \frac{\sqrt{(x_{t+1} - x_t)^2 - (y_{t+1} - y_t)^2}}{dt} \quad (1)$$

where $x_t, y_t, x_{t+1}, y_{t+1}$ are the screen coordinates of the gaze detected in each frame, and dt is the time between two successive frames. If the fixation-detection algorithm is not activated, the response time is expected to be equal to dt . If the fixation-detection algorithm is active the response time of the system will range from $2 \cdot dt$ to $4 \cdot dt$. In any eye tracking device, noise will be registered due to the inherent instability of the eye, and specially due to blinks [17]. The EyeWriter software used for tracking the gaze coordinates, provides some configurations that can help to reduce that noise. By setting the minimum and maximum

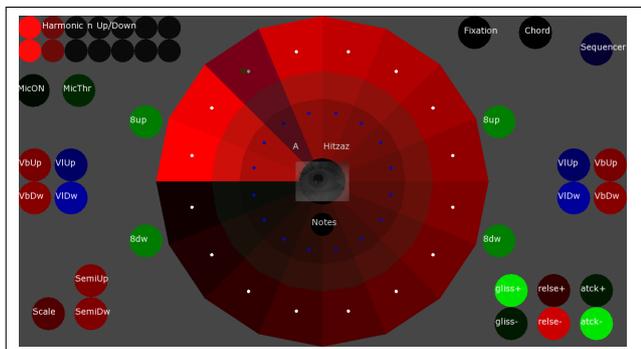


Figure 6. The EyeHarp without chords.

of the pupil' s size to the proper values the system might ignore the blinks in most of the cases. Another possible adjustment is the smoothing amount. The smoothed coordinates are given by the equation:

$$x_n = S * x_{n-1} + (1 - S) * Gx_n$$

$$y_n = S * y_{n-1} + (1 - S) * Gy_n$$

where x, y are the smoothed gaze values, Gx_n, Gy_n are the raw data of the gaze detection and S is the smoothing amount. $0 \leq S \leq 1$. For maximum temporal control, the smoothing amount should be set to zero.

The PlayStation Eye camera that is used in this project is capable of capturing standard video with frame rates of 75 hertz at a 640×480 pixel resolution. The program has been tested on a Intel Core i5 460M processor with 4GB of RAM and an nVIDIA GeForce GT 330M graphic card. For the sound to be generated smoothly, the refresh rate is set at 30 frames/second. Thus, without the fixation-detection algorithm the response time is 25ms, while with the fixation algorithm it is 50-100ms.

Spatial Distribution of the Notes The EyeHarp layer is displayed in Figure 5 and Figure 6. In order to be able to play the instrument without a fixation-detection algorithm all the notes are placed at the periphery of a circle. In the middle of this circle there is a small black circle, where the performer' s eye is displayed. If the performer looks at this circle, then the played note is released. The fixation detection algorithm is always active for this specific region. The reason for this is that the user should be able to play any melodic interval without accidentally releasing the played note. So the release region in the center is triggered only when a fixation is detected. Using that spatial distribution, the user can have control over the articulation of the sound (staccato, legato). To play staccato, after triggering a note the user' s gaze should quickly return to the center of the circle in order to release it soon. One more advantage of that spatial distribution is that all the notes are relatively close to each other, so it is easy to play every possible melodic interval. At the center of every note there is a white spot that helps the user to focus on it. A note is triggered immediately when the gaze of the user is detected inside its region. Almost no controls are placed in the region inside the circle. The user' s gaze can move freely inside this region without triggering anything. A second row of blue dots are placed inside this region.

Before playing a note the user can first look on the corresponding blue spot inside the circle and then play it by looking at the white spot placed at the periphery. This way of "clicking" provides an optimum temporal control, since the note is triggered exactly when the user looks at it. If the fixation-detection algorithm is inactive, the response time is only limited by the frame rate. The dark color indicates a low pitch, while a bright color indicates a higher pitch. So the pitch increases in a counter clockwise order, starting from the most left note of the circle. If the gaze of the performer is between the small black circle in the center and the notes at the periphery of the main circle, nothing happens. The last triggered note will keep on being generated until a new note is played or it is released. As already mentioned the instrument is diatonic, so every seven or five (for pentatonic) notes we have a new octave.

Spatial distribution of the control buttons All the control buttons work in the same way as described in the step sequencer layer: there is a time threshold -different for every button- for moving the corresponding variable one step up or down.

The only control button that is inside the main circle is the one that deactivates all the notes. Obviously, there was a need for the gaze to move outside the circle in order to change several aspects of the synthesized sound. If the fixation-detection is inactive, in order to go outside the circle without triggering a note, the notes should be deactivated first. They can be activated again by looking at the black circle in the middle of the interface.

On the upper right corner of the screen, there is the "fixation" button for activating or deactivating fixation-detection. Next to it, there is the "chord" button. When it is active the notes at the upper part of the circle are assigned for changing the harmonies of the Step Sequencer. The user can build an arpeggio in the sequencer layer and then change the harmonies of his composition in the EyeHarp layer. The closest buttons to the main circle are the ones for changing octaves, and they are placed close to the lowest and highest pitches of the interface. If the fixation-detection is not active, when pressing any of the buttons for changing octave, the notes are automatically deactivated, so the user can enter inside the main circle again without triggering any note accidentally.

As it can be seen in Figure 5, there are buttons for adjusting the glissando, attack, release, volume, vibrato, amplitude of each harmonic, tonality (semitone up, down and "mode"), and switching to the sequencer layer. The two layers have their own sound properties, apart from the ones related to the tonality. That means that the timbre, articulation, temporal aspects of the sound, octave of each layer can be set to different values (e.g. choose a percussive timbre for the sequencer and a harmonic timbre for the melody). The user can also activate the microphone input for blowing in the microphone and having control over the amplitude of the melody that he is performing (a very dynamic microphone is recommended). The minimum sound level to be considered as an input can be set through the "MicThr" button.

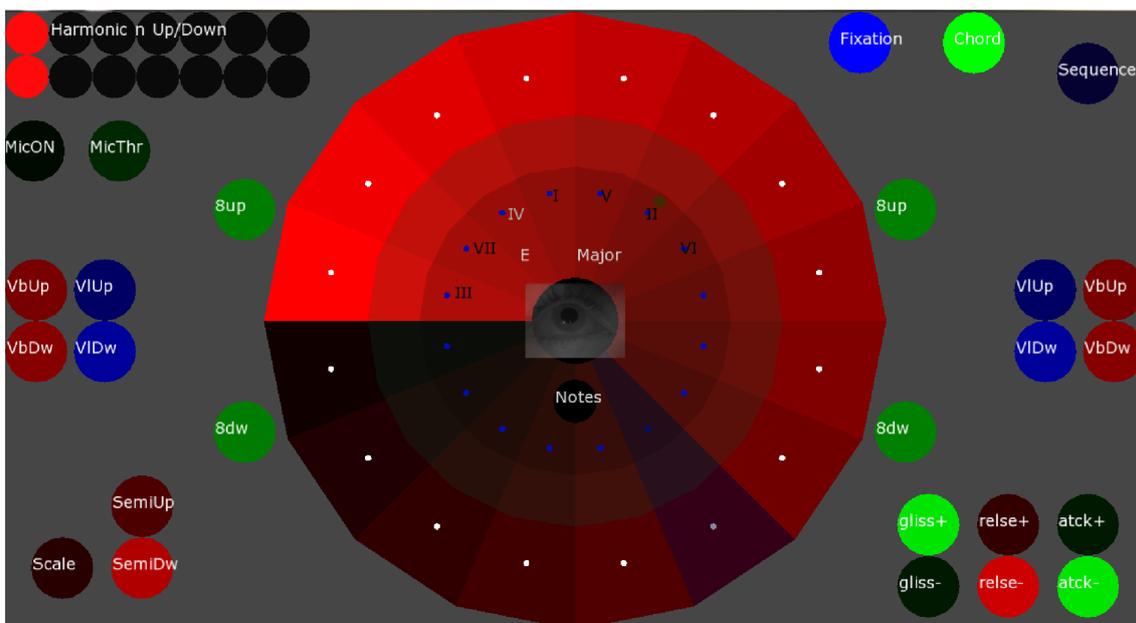


Figure 5. The EyeHarp layer. The selected scale is E major. The 4th chord of the key is played (A minor) and the melody note is do. The user is about to play the second chord of the key (F# minor).

3.3 Implementation

The interface and sound synthesis of the EyeHarp were implemented in Openframeworks [18], an open source C++ toolkit for creative coding. Openframeworks is used in all the stages of the system: (i) tracking the pupil of the eye and calibrating (based on the EyeWriter Project), (ii) designing the different modes of the EyeHarp (iii) synthesizing the sound.

3.4 Evaluation

Evaluating a new musical instrument is a difficult task. Ideally, the instrument should be evaluated at different stages. It should be evaluated on how accessible or “playable” it is for novice performers, how easy/difficult it is to improve with experience, and what is the potential of the instrument performed by experts. As a preliminary evaluation we have asked two people, one person completely novice to the instrument (playing the EyeHarp for the first time), and another more experienced person who had spent many hours using the EyeHarp, to each perform two tasks: perform a two octave scale using the EyeHarp interface as accurate and speedy as possible, and generate a note pattern on the EyeHarp melodic step sequencer as speedy as possible. In addition, for comparison purposes we have asked the same two people to perform the same tasks using a video-based head tracking software [19]. Figure 7 shows the results of the experiment.

Both (the experienced and novice) participants agreed that proficiency in the EyeHarp improves with practice. Observing the participants interact with the EyeHarp after the experiment, it seems that the fixation-detection algorithm is indeed very helpful for a novice user and can be activated for increasing the spatial accuracy of the system. The smoothing amount can be adjusted as well. The user can

| User | Tracking | Two octave scale in the EyeHarp | | Arpeggio in the Step Sequencer |
|--------|----------|---------------------------------|----------|--------------------------------|
| | | Seconds | Accuracy | Seconds |
| Expert | Eye | 12 | 100% | 15 |
| | Head | 36 | 100% | 18 |
| Novice | Eye | 18 | 94% | 30 |
| | Head | 40 | 75% | 29 |

Figure 7. Eye-Tracking and Head-Tracking for an experienced and a novice user.

choose between better spatial (not pressing notes accidentally) or temporal control by adjusting these two parameters.

It has to be noted that the accuracy of the implemented eye-tracking device was not explicitly evaluated (this is out of the scope of this paper). However, the EyeHarp interface can be used along with more accurate commercial eye tracking systems. It is very likely that the temporal and spatial control would be even better in that case.

Probably the best way to evaluate the potential of the EyeHarp as a musical instrument is to listen to performances produced using the instrument. The reader may listen (and watch) one such performance at:

<http://www.dtic.upf.edu/~rramirez/eyeharp/EyeHarpDEMO.wmv>

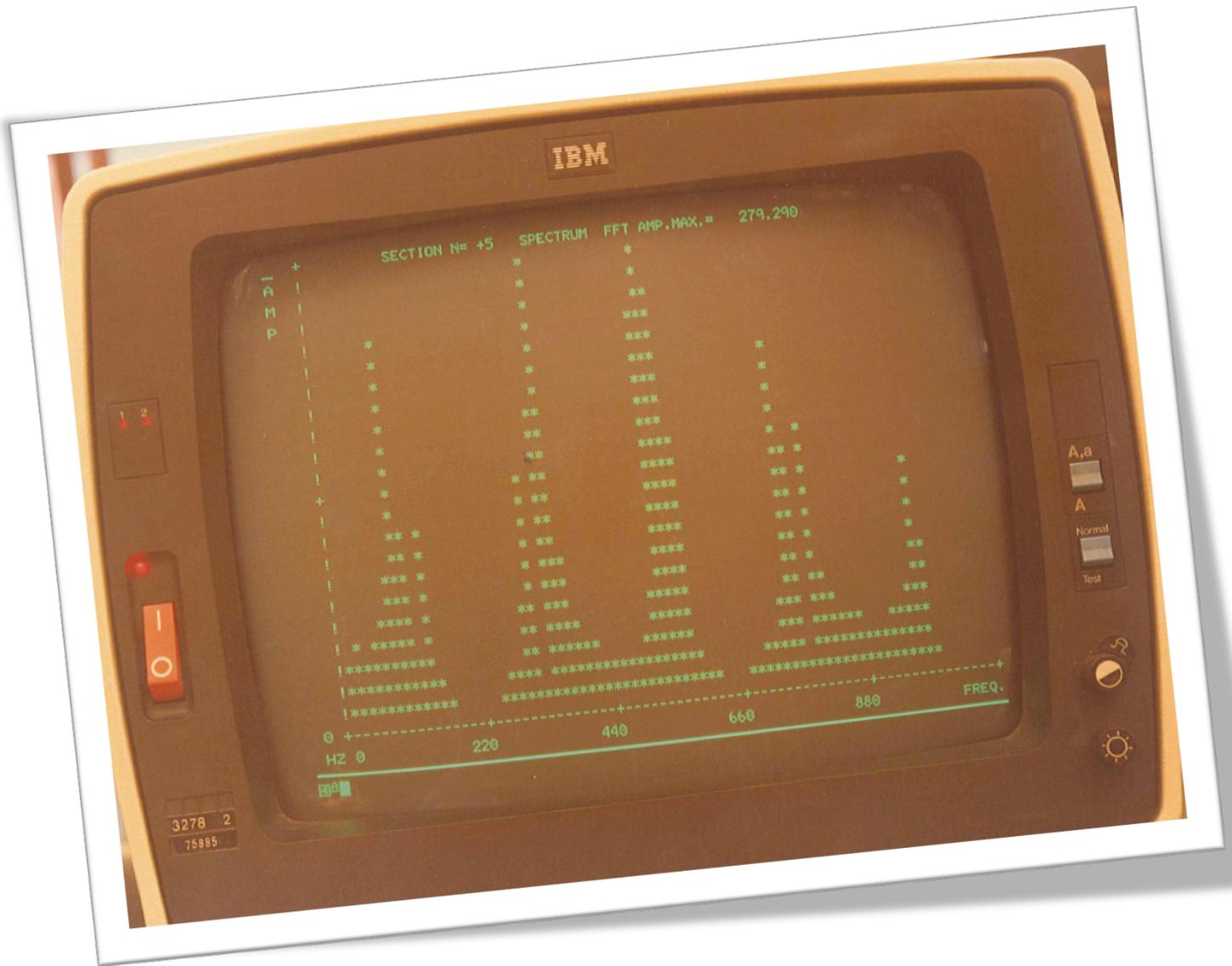
4. CONCLUSIONS

We have presented the EyeHarp, a new musical instrument based on eye tracking. We have built a low-cost eye-tracking device which communicates with a melody and step sequencer interface. The interface allows performers and composers to produce music by controlling sound settings and musical events using eye movement. We have described the development of the EyeHarp, in particular design and implementation of the melody and step sequencer interface. Finally, we have conducted a preliminary experiment for evaluating the system and compare its usability with a similar video-based head tracking controller. The results are encouraging but are still preliminary because the evaluation included only one experienced performer and one novice performer. The EyeHarp interface is still under development and many aspects, such as the choice of colors and spatial distribution of the control buttons, are still under reconsideration.

The eyeWriter project team has provided “a low-cost eye-tracking apparatus and custom software that allows graffiti writers and artists with paralysis resulting from Amyotrophic lateral sclerosis to draw using only their eyes”. The eyeHarp is a musical instrument that could give to these people the opportunity to express themselves through music, but can also be used by anyone as musical instrument in a traditional way.

5. REFERENCES

- [1] B. M. Galayev, “Light and shadows of a great life: In commemoration of the one-hundredth anniversary of the birth of Leon Theremin,” *Pioneer of Electronic Art. Leonardo Music Journal*, pp. 45–48, 1996.
- [2] M. Waiswicz, “The hands: A set of remote MIDI controllers.” in *Proceedings of the 1985 International Computer Music Conference*. San Francisco: Computer Music Association, 2003, pp. 573–605.
- [3] A. Tanaka, “Musical technical issues in using interactive instrument technology.” in *Proceedings of the International Computer Music Conference*. San Francisco: International Computer Music Association, 1993, pp. 124–126.
- [4] R. J. K. Jacob and K. S. Karn, “Eye tracking in human-computer interaction and usability research: Ready to deliver the promises,” in *The Mind’s Eyes: Cognitive and Applied Aspects of Eye Movements*. Elsevier Science, H. D. J. Hyona, R. Radach, Ed., 2003, pp. 573–605.
- [5] The eyeWriter project. [Online]. Available: <http://www.eyewriter.org/>
- [6] J. A. Werner, “A method for arousal-free and continuous measurement of the depth of sleep in man with the aid of electroencephalo-, electrooculo- and electrocardiography (eeg, eog, and ekg),” in *Z Gesamte Exp. Med.*, 1961, vol. 134, pp. 187–209.
- [7] S. Iwasaki, L. A. McGarvie, G. M. Halmagyi, A. M. Burgess, J. Kim, J. G. Colebatch, and I. S. Curthoys, “Head taps evoke a crossed vestibulo-ocular reflex.” in *Neurology (in press)*, December 2006.
- [8] D. A. Robinson, “A method of measuring eye movement using a scleral search coil in a magnetic field,” *IEEE Trans Biomed Engineering*, pp. 137–145, 1963.
- [9] S. T. Moore, T. Haslwanter, I. S. Curthoys, and S. T. Smith, “A geometric basis for measurement of three-dimensional eye position using image processing. vision research,” in *Vision Research*, 1996, vol. 36, no. 3, pp. 445–459.
- [10] D. Zhu, S. T. Moore, and T. Raphan, “Robust and real-time torsional eye position calculation using a template-matching technique.” in *Comput. Methods Programs Biomed.*, 2004, vol. 74, pp. 201–209.
- [11] A. Polli, “Active vision: Controlling sound with eye movements,” *Leonardo*, vol. 32, no. 5, pp. 405–411, 1999, seventh New York Digital Salon.
- [12] A. Hornof, “Bringing to life the musical properties of the eyes,” University of Oregon, Tech. Rep., 2008.
- [13] A. J. Hornof and K. E. V. Vessey, “The sound of one eye clapping: Tapping an accurate rhythm with eye movements,” Computer and Information Science University of Oregon, Tech. Rep., 2011.
- [14] J. Kim, “Oculog: Playing with eye movements,” in . Nime 07, 2007.
- [15] Y. Nishibori and T. Iwai, “Tenori-on,” in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME06)*, Paris, France, 2006.
- [16] Max for live. [Online]. Available: <http://www.ableton.com/maxforlive/>
- [17] A. Duchowski, *Eye Tracking Methodology: Theory and Practice*, 2nd ed., Springer, Ed., 2007.
- [18] openframeworks. [Online]. Available: <http://www.openframeworks.cc/>
- [19] M. Betke, J. Gips, and P. Fleming, “The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities,” *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 10, no. 1, p. 110, 2002.



POSTER PRESENTATIONS

IMPROVING TEMPO-SENSITIVE AND TEMPO-ROBUST DESCRIPTORS FOR RHYTHMIC SIMILARITY

Andre Holzapfel, Arthur Flexer and Gerhard Widmer

Austrian Research Institute for Artificial Intelligence (OFAI)

aholza@inescporto.pt, arthur.flexer@ofai.at, gerhard.widmer@jku.at

ABSTRACT

For the description of rhythmic content of music signals usually features are preferred that are invariant in presence of tempo changes. In this paper it is shown that the importance of tempo depends on the musical context. For popular music, a tempo-sensitive feature is improved on multiple datasets using analysis of variance, and it is shown that also a tempo-robust description profits from the integration into the resulting processing framework. Important insights are given into optimal parameters for rhythm description, and limitations of current approaches are indicated.

1. INTRODUCTION

Determining the similarity between two pieces of music is one of the core problems in Music Information Retrieval (MIR). Methods to estimate such similarity usually consider the timbre of music, *i.e.* the instantaneous sound characteristics that are contained in a sample. Similarity measures based on timbre can be improved by adding the aspect of rhythmic similarity [1]. However, while the meaning of timbre similarity is somehow intuitive, rhythmic similarity is a more abstract concept. In Cooper and Meyer [2], rhythm is defined as the way one or more unaccented beats are grouped in relation to an accented one. Furthermore, meter is defined as the measurement of the number of pulses between more or less regularly occurring accents. Even though rhythm can be perceived without the existence of a meter, in this paper we will restrict to music signals that have a meter. As soon as we impose this restriction, each piece of music is characterized by a frequency of pulsation (*i.e.* a pulse-tempo), that determines how fast the accents in the metrical structure are performed. Thus, in order to achieve high similarity values for similar pieces that are performed at different tempi, one approach is to make descriptions of rhythmic content independent of this pulse-tempo. Such descriptors were *e.g.* proposed by Peeters [3] and Jensen *et al.* [4]. These descriptors are based on periodicity representations: Given a music signal, the periodicities caused by its regularly occurring accents are estimated. Then it is tried to make

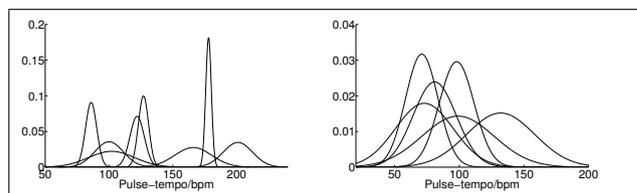


Figure 1. Tempi of the Ballroom (left) and the Turkish Art music (right) datasets modeled by Gaussian distributions.

either these representations or the applied similarity measure between them robust to tempo changes. An important question that will be addressed in this paper is whether such invariance is desirable in every context, or if there are cases in which a certain sensitivity of the descriptors to changing pulse-tempo is of advantage for the rhythmic similarity measurements.

In order to get an understanding of the significance of this question, let us have a look at two music collections: First, a collection of eight western Ballroom dances that is widely used in experiments in the MIR research community (*e.g.* in [3]), and, second, a collection of Turkish Art music divided into six metric classes that has been compiled by Holzapfel and Stylianou [5]. The pulse-tempo of all pieces is known for these collections. In Figure 1, the pulse-tempo in beats per minute (bpm) of all pieces in each contained class was modeled by a Gaussian distribution. For the Ballroom collection it is obvious that tempo can serve as a valuable information in order to differentiate between samples of different dances, a fact that was observed by Dixon *et al.* [6] for this dataset. The tempo distributions of the Turkish Art music collection, however, reveal opposite conditions for a good similarity measure. On this collection, it appears to be a good choice not to consider tempo information, because distributions have large overlaps and standard deviations.

Thus, depending on the type of music samples we want to compare we would either choose to discard tempo information, or to use it for improving our similarity measure. However, in most cases an annotated ground truth of the pulse-tempo is not given, and it must be estimated from the audio signal. Including estimated instead of annotated tempo information will lead to a decreased performance of the similarity measure, as shown recently by Peeters [3]. This is due to the fact that the tempo estimation is subject to halving- and doubling errors, and its accuracy depends strongly on the signal characteristics [7]. For that reason, it would be desirable to have two types of descriptors at

hand. In the first case, when we want to discard tempo information, a descriptor that completely ignores tempo information would be preferred, as *e.g.* for Turkish and Arabic art music. In the second case, we would prefer a descriptor which remains invariant for a small range of tempo changes, and which automatically varies in presence of larger tempo changes. To give an example, for Hip Hop music one would like to have descriptors that do not vary when the same beat is used in another track with a difference of only 5 beats per minute, but the contained shuffled grooves would appear altered and of different character when changed by 20 bpm.

For that reason, it was chosen to contrast two different techniques for rhythmic similarity estimations. The first was presented by Holzapfel and Stylianou [8] and is based on the Scale Transform Magnitudes (*STM*). This method was shown to be invariant to tempo changes. The second method was introduced by Pohle *et al.* [1], and applies descriptors that are referred to as Onset Patterns (*OP*). Large changes in tempo lead to a shift in these descriptors, but small changes in tempo leave this representation almost unchanged as shown in an example in Section 3. For both descriptors, no estimation of the pulse-tempo from the signal is necessary.

In this paper, with the availability of multiple datasets, it was feasible to conduct a series of analyses of variance (ANOVA) [9] in order to find improved parameters for rhythm descriptors. Improvements are related to optimal multi-band processing schemes, length of applied analysis windows, and the resolution which is necessary to obtain a good similarity descriptor. Our experimental setup can serve as an example of how to obtain optimal system parameters when several data sources are given. Until now, such parameters are usually found in a trial and error procedure, and not in a rigorous statistical setting as in our contribution.

Optimal parameters will be obtained by performing ANOVA on the *OP* computation, but it will be shown that the *STM* based rhythm descriptors profit from the obtained system improvement in the same way. This confirms that the found processing framework is generalizable and can be applied to other descriptors as well. We will then contrast the performance of the *OP* and *STM* descriptors on various datasets in order to verify the correctness of our hypothesis about the context-dependent meaning of pulse-tempo for rhythmic similarity.

The following Sections of this paper are structured as follows: In Section 2 experimental methods are detailed. Datasets are described, it is detailed how conclusions about the accuracy of descriptors are obtained, and *STM* and *OP* descriptors will be outlined, with emphasis on the method to improve the *OP* framework. Then, in Section 3, the different degree of robustness to tempo changes of *OP* and *STM* descriptors will be clarified in some examples. The results of the analyses of variance and comparisons between *STM* and *OP* features are given in Section 4, and Section 5 concludes the paper.

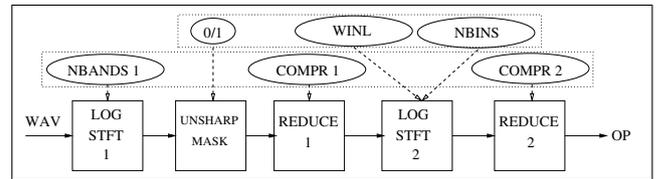


Figure 2. *OP* computation and system parameters.

2. EXPERIMENTAL SETUP

2.1 Rhythm Content Description

In Figure 2 the computation of Onset Patterns (*OP*) is summarized. The computations are symbolized by the bold rectangular boxes, and the dotted rectangular boxes show the parameters that will be evaluated in separate ANOVAs. As indicated by the dotted boxes, parameters are grouped into two sets. The first set, *NBANDS1*, *COMPR1* and *COMPR2* are related to the multi-band processing. The second set, *WINL*, *NBINS* and the usage of unsharp masking are related to the *OP* computation. A simultaneous analysis of those factors and all their interactions would be too challenging. We believe that factors in the two parameter sets are sufficiently independent to be improved separately. The rather coarse grid of factor levels (see Section 4) is also due to considerations of tractability. In the following paragraph, the computation of *OP* will be described and the meaning of the mentioned parameters will be explained.

Input to the first computation in Figure 2 is a monophonic piece of music sampled at 22050 Hz. The input is transformed into the frequency domain using a STFT with a 46.4 ms length Hanning window with half overlap. The magnitude of the transform is then processed by a filterbank in order to obtain coefficients on a logarithmic axis. The number of bands on this axis is denoted as *NBANDS1* in Figure 2, and was set to 85 by Pohle *et al.* [1]. In each of these bands, a masking can be computed in order to accentuate instrument onsets by emphasizing transient regions in the signal. This masking applies a moving average filter with a length of 0.25 s to each band and then half-wave rectifies the output. In this paper we will retain the notation of unsharp masking for this process which was used in [1]. Then the logarithm of the signal is computed and the *NBANDS1* bands can be reduced by the factor *COMPR1*. In [1], 85 bands were reduced to 38, which results in a compression of $COMPR1 = 85/38 \approx 2.24$. Then, a second STFT is computed on each band in order to obtain a description of the periodicities contained in this band. Such a description will be referred to as periodicity spectrum. The periodicity spectral magnitudes are mapped onto a logarithmic axis by applying a filter bank. In this computation, it was decided to evaluate the optimal analysis window length of the STFT (*WINL*) and the number of bins per octave that are obtained from the filter bank, the original values were 6s and 5 bins per octave [1]. The periodicities are described in five octaves from 30 to 960bpm. It should be pointed out that no zero padding was used in the STFT's, and a Hanning window of *WINL* length in seconds with a shift of half a second was applied to obtain the periodicity spectra. In the final stage of the *OP* computation,

it was tried to reduce the number of bands again, in order to obtain more compact descriptors. This results in a two stage compression scheme, starting from NBANDS1 bands. The rhythm of a whole sample is described by the mean of the *OP* obtained from the various segments of this sample.

A method that is robust to tempo variance in a very wide range is the description based on Scale Transform Magnitudes (*STM*) as proposed by Holzapfel and Stylianou in [8]. The computation of these descriptors was left exactly as explained therein, and its basic computation steps are depicted in Figure 3. The first step is a computation of a spectral flux based Onset Strength Signal (*OSS*). Within moving windows of eight seconds length, autocorrelation coefficients are computed and then transformed into the scale domain by applying a discrete Scale Transform. For one sample, the mean of the *STM*'s obtained from all the analysis windows are the *STM* descriptors of the rhythmic content of a sample. For the exact computation parameters please refer to [8]. However, it should be pointed out that the final descriptors do not contain separate information from various bands as for the *OP*. In order to im-

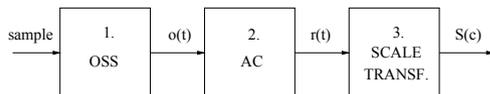


Figure 3. Computational steps of *STM* rhythm descriptors.

prove the existent descriptors, the two parameter groups in the *OP* computation depicted in Figure Figure 2 are evaluated in a two stage analysis of variance. In the first stage, the optimal parameters for the multi-band parameter set (NBANDS1, COMPR1, COMPR2) are evaluated in an ANOVA with the observations being the 1-nearest-neighbor classification accuracies on three datasets using Onset Patterns. For this, the parameters from the second set were set to the values applied in [1] (usage of unsharp masking, WINL = 6s, NBINS = 5). After deciding on the values for the parameters for the first set, these values are fixed and optimal values for the second set are found using a second ANOVA again with the observations being the accuracies on the same three datasets using Onset Patterns. Then, the improved processing framework for *OP* will be applied to *STM* in order to prove the validity of the obtained parameters also for these descriptors.

2.2 Datasets and evaluation

In order to improve the system depicted in Figure 2, three data sets will be used. The first, DBall, is the widely used Ballroom dataset, consisting of 8 classes with 698 ballroom dance excerpts of 30s length. The second dataset, DLat, was presented by Silla *et al.* [10], and contains 3226 files of Latin dance music in 10 classes. Finally, a third dataset, DPop, was compiled that consists of 347 excerpts from popular music samples organized into 15 different

classes that are related to rhythmic concepts (*e.g.* *Break Beat* and *Jive*). In order to investigate the different demands on the context of traditional music, two more datasets will be used to compute similarity measurements. The first, DCrete, was used by Holzapfel and Stylianou [8] and contains 180 short excerpts of six different dances commonly encountered in the island of Crete in Greece. The second traditional dataset, DTurk, contains 288 audio samples synthesized from melodies of Turkish art music. The pulse-tempo distributions of this dataset are shown in Figure 1, and further details on the dataset and the synthesis method are given in [8].

As the application for the proposed features is music similarity, the features will be evaluated in a 1-Nearest-Neighbor classification in a *leave-one-out* scheme. The distance between features will be Euclidean distance in all cases. The obtained classification accuracies will be used to find an optimal computation setup, and will serve as a way to contrast the performance of different features when applied to music of different style.

3. TEMPO ROBUSTNESS

In order to show the influence of tempo changes on the *OP* and *STM* descriptors, a simple experimental setting was chosen. From each class of DBall, DLat and DPop one song was chosen and its tempo was manipulated without changing pitch using the audacity audio editor. The tempo of each song was changed by +/-20%, +/-10%. This results in five tempo variants for each song, including the original tempo. Following this procedure, 22 songs in five tempo variants were obtained. Note that for classes which appear in several datasets (*e.g.* *Tango*), only one sample was used. The accuracy of correctly identifying a song in a 1-NN classification was determined. This means that it was determined how often the nearest neighbor is indeed a tempo changed version. Additionally, the average ratio of the distances between a song and all different songs and distances between a song and its tempo variations was computed. For example, if this ratio equals 2, the distance of one song to a different song is on average two times larger than the distance of a song to its tempo variants. Hence, larger numbers of this ratio indicate a better robustness to the tempo changes. The applied features are the *OP* and *STM* with the original parameters as presented in [1] and [8], respectively. The results shown in Table 1 clearly show the supe-

| | <i>OP</i> | <i>STM</i> |
|----------|-----------|------------|
| ACCURACY | 70.0 | 83.6 |
| RATIO | 2.26 | 3.03 |

Table 1. Song identification in presence of tempo changes

rrior tempo robustness of the *STM* features, both in terms of ratio and in terms of accuracies. However, we should have a closer look at the effect of small tempo changes on the *OP* features. This effect is visualized in Figure 4, where the low coefficients of a periodicity spectral magnitude of a Cha-cha-cha sample is shown as a bold line.

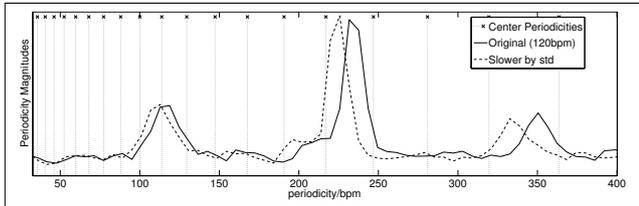


Figure 4. Excerpt of periodicity spectral magnitude of a Cha-cha-cha sample. Small changes in tempo lead to minimal change in *OP* due to the log-filterbank (center frequencies shown as dotted vertical lines).

This piece has an annotated tempo of 120 bpm, and the standard tempo deviation of this class in DBall is 5.6bpm. The dashed periodicity magnitude has been derived from the same piece, when its tempo has been changed by this standard deviation using audacity. The dotted vertical lines denote the positions of the log-filterbank center frequencies that map the periodicity spectral magnitudes to a logarithmic axis (LOG-STFT2 in Figure 2). It is obvious that this change in tempo leads to a minimal change in the resulting descriptors due to the coarse frequency resolution. This confirms that the *OP* descriptors are robust to tempo changes within certain limits that are determined by the NBINS parameter in Figure 2.

4. RESULTS

As explained in Section 2, two ANOVAs were performed as indicated by the dotted boxes in Figure 2. The results will be analyzed starting with the multi-band processing scheme.

4.1 Multi-band Processing ANOVA

We performed a four-way analysis of variance (ANOVA) with the following factors: “Number of bands” (NBANDS1, 4 levels: 16, 32, 64, 128), “Compression 1” (COMPR1, 3 levels: 1, 2, 4), “Compression 2” (COMPR2, 3 levels: 1, 2, 4), “Data set” (DS, 3 levels: DBall, DLat, DPop). We also looked into possible two-factor interactions. The dependent variable is the accuracy resulting from 1-Nearest-Neighbor classification. As can be seen in Table 2, all main effects as well as two-factor interactions are significant at the .05 error level (see last column, $P_{rob} > F$ smaller than 0.05).

| Source | Sum Sq. | d. f. | Mean Sq. | F | Prob>F |
|----------------|---------|-------|----------|---------|-----------|
| NBANDS1 | 0.0369 | 3 | 0.0123 | 76.82 | 8.469e-22 |
| COMPR1 | 0.0104 | 2 | 0.005 | 32.42 | 1.296e-10 |
| COMPR2 | 0.0037 | 2 | 0.0018 | 11.6 | 4.624e-05 |
| DS | 0.5807 | 2 | 0.2903 | 1810.14 | 1.079e-59 |
| NBANDS1*COMPR1 | 0.0179 | 6 | 0.0029 | 18.64 | 1.099e-12 |
| NBANDS1*COMPR2 | 0.0052 | 6 | 0.0008 | 5.48 | 1.125e-04 |
| NBANDS1*DS | 0.0405 | 6 | 0.0067 | 42.17 | 4.803e-21 |
| COMPR1*COMPR2 | 0.0031 | 4 | 0.0007 | 4.86 | 0.0017 |
| COMPR1*DS | 0.0115 | 4 | 0.0028 | 18.03 | 3.916e-10 |
| COMPR2*DS | 0.0081 | 4 | 0.0020 | 12.68 | 9.152e-08 |
| Error | 0.0109 | 68 | 0.0001 | | |
| Total | 0.7294 | 107 | | | |

Table 2. Result table of multi-band ANOVA

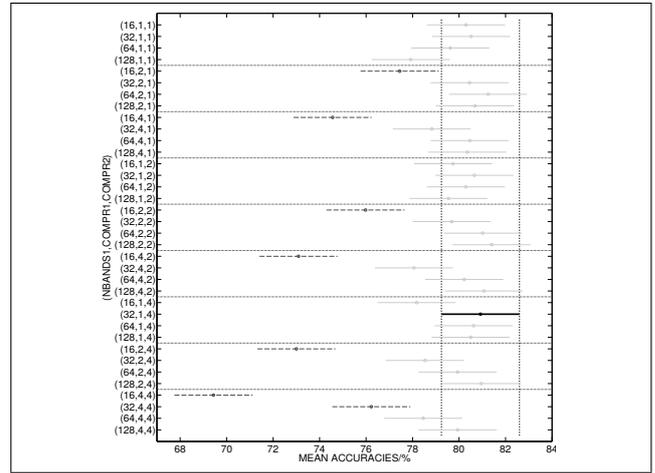


Figure 5. Influence of the number of bands and the compression factors. Chosen parameter is shown as a bold line, the significantly different means are depicted by dashed lines. Additional horizontal lines improve legibility.

Therefore there are significant effects on the classification accuracy caused by the number of frequency bands, the first and second compression rates as well as the data set. The fact that the type of data set used has an influence on the accuracies achieved is clear since the three data sets have different levels of difficulty (mean accuracies for the datasets are 87.1% (DBall), 80.1% (DLat) and 69.3% (DPop)). Since all two-factor interactions are also significant, we have to investigate all factors together to find out which combinations of factors are optimal in terms of achieved accuracy. It is important to point out that the two-factor interactions with the datasets (DS) in this as well as in the second ANOVA only influence the degree but not the direction a factor has on the dependent variable. Therefore we can analyze aggregate results across all three datasets. In Figure 5 we plotted mean accuracies and 95% confidence intervals for all combinations of factors “Number of bands” (NBANDS1), “Compression 1” (COMPR1) and “Compression 2” (COMPR2). The mean accuracies are based on the results from all three data sets. A considerable number of combinations of factors is able to achieve similar levels of mean accuracy of around 80% and more. We concentrate on one combination that achieves good accuracy and compact representation at the same time: NBANDS = 32, COMPR1 = 1, COMPR2 = 4, *i.e.* this combination uses a log-filterbank with 32 filters after the first STFT, and reduces the resulting number of bands to eight in the second reduction in Figure 2. The periodicities in each of these eight bands are described using 25 coefficients (5 NBINS \times 5 octaves). This specific combination is shown using a bold line in Figure 5. Based on the results from the ANOVA, we compare the mean accuracy for this one combination with all other combinations with a series of t-tests (level of significance $\alpha = .05$). Tukey’s HSD adjustment was used to account for the effect of multiple comparisons. All combinations significantly different from the chosen combination are shown as dashed lines. These combinations start

at a low number of bands (NBANDS1=16), and then further reduce this representation, except one case in which starting with 32 bands and reducing them to 2 bands leads to significant decrease. Compared to the chosen scheme, no other higher dimensional combination can significantly improve the results. This shows that a number of bands much smaller than the number of semitone bands (85) is sufficient. This number can be further compressed to obtain a more compact descriptor; A lower bound for the number of bands to start with is at about 32, and a lower bound of bands to keep at the end is 4.

At this point it should be pointed out that instead of the second reduction in Figure 2, also usage of a two dimensional DCT was considered, which resulted in the Onset Coefficients proposed in [1]. The first DCT reduces the number of bands, while the second DCT reduces dimensionality of the periodicity content description in every band. However, it was found that by using DCT the dimensionality of periodicity content description cannot be further reduced, and application of a DCT to the dimension of the bands leads to no performance gain compared to our simple reduction based on linear combination of neighboring bands. Moreover, when applying a DCT results are no longer nicely interpretable as log-periodicity spectra, and for those reasons it appears to be preferable to refrain from using Onset Coefficients.

4.2 Processing parameter ANOVA

We performed a four-way analysis of variance (ANOVA) with the following factors: “Unsharp mask” (MASK, 2 levels: 0, 1), “Window length” (WINL, 4 levels: 6, 8, 10, 12), “Number of bins” (NBINS, 4 levels: 3, 4, 5, 6), “Data set” (DS, 3 levels: DBall, DLat, DPop). We also looked into possible two-factor interactions. The dependent variable is again the accuracy resulting from 1-Nearest-Neighbor classification. As can be seen in Table 3, all main effects as well as the two-factor interactions “MASK*DS” and “NBINS*DS” are significant at the .05 error level. There

| Source | Sum Sq. | d.f. | Mean Sq. | F | Prob>F |
|------------|---------|------|----------|---------|------------|
| MASK | 8224.3 | 1 | 8224.28 | 5495 | 2.2382e-58 |
| WINL | 171.5 | 3 | 57.15 | 38.19 | 1.1558e-13 |
| NBINS | 239.7 | 3 | 79.89 | 53.38 | 1.4701e-16 |
| DS | 6772.5 | 2 | 3386.23 | 2262.48 | 5.0402e-55 |
| MASK*WINL | 10.9 | 3 | 3.63 | 2.43 | 0.0747 |
| MAKS*NBINS | 10.6 | 3 | 3.53 | 2.36 | 0.0811 |
| MASK*DS | 671.2 | 2 | 335.61 | 224.23 | 9.7103e-28 |
| WINL*NBINS | 18.5 | 9 | 2.05 | 1.37 | 0.2229 |
| WINL*DS | 29.5 | 6 | 4.92 | 3.29 | 0.0075 |
| NBINS*DS | 50.9 | 6 | 8.49 | 5.67 | 0.0001 |
| Error | 85.3 | 57 | 1.5 | | |
| Total | 16284.8 | 95 | | | |

Table 3. Result table of processing parameter ANOVA

is a strong positive effect of using the “Unsharp mask” on the accuracy in all tested combinations of parameters. To be precise, mean accuracies improved by 22.4% for DBall, 10.9% for DLat, and 22.1% for DPop. In Figure 6 we plotted mean accuracies and 95% confidence intervals for all levels of factor “Window length” (WINL). The mean accuracies are based on the results from all three data sets. The result for using a window length of WINL = 8 is shown

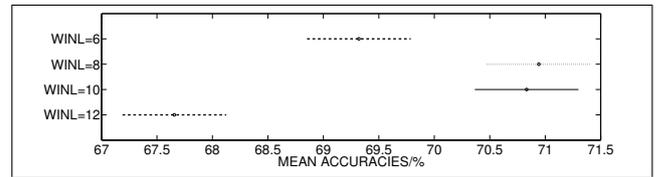


Figure 6. Influence of WINL in the LOG-STFT 2. Chosen parameter shown as a dotted line, the significantly different means are depicted by dashed lines.

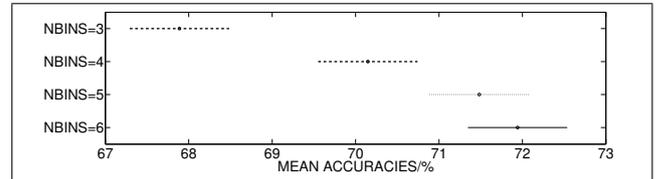


Figure 7. Influence of NBINS in the LOG-STFT 2. Chosen parameter shown as a dotted line, the significantly different means are depicted by dashed lines.

as a dotted line Figure 6. Using WINL = 8 is significantly better than using WINL = 6 or 12 and equally good as using WINL = 10 (based on t-tests, $\alpha = .05$, Tukey’s HSD adjustment). In Figure 7 we plotted mean accuracies and 95% confidence intervals for all levels of factor “Number of bins” (NBINS). The mean accuracies are based on the results from all three data sets. The result for using NBINS = 5 number of bins is shown as a dotted line Figure 7. Using NBINS = 5 is significantly better than using NBINS = 3 or 4 and equally good as using NBINS = 6 (based on t-tests, $\alpha = .05$, Tukey’s HSD adjustment).

The most important conclusion from the processing parameter ANOVA concerns the effect of the window length. By using STFT, we are bound to the stationarity constraint. In this paper, the finding is that increasing window lengths to values of more than 8s leads to problems. This phenomenon was described in [8] as well, but on partly different datasets and using different descriptors. However, the common aspect is that the processed signal had to be stationary within their analysis window as well. This leads to the conclusion that rhythmic aspects of music performances tend to be non-stationary beyond this temporal limit.

4.3 Summary and comparison

In order to quantify the performance gain that is achieved when using the optimized parameters, we will contrast the accuracies of the improved features (OP_{opt}) with the original setting from [1], denoted as OP_{org} . The exact parameters of the improved and the original setups are listed in Table 4.

In Table 5.(a), accuracies on all five datasets using the OP features are depicted, while Table 5.(b) shows the accuracies for the STM_{org} features computed as described in Section 2 together with the STM_{opt} , which were obtained by integrating the STM computation into the improved multi-band processing scheme: Instead of the logarithmic filterbank applied in the LOG-STFT 2 step in Figure 2, we

| Parameter | Improved | Original |
|-----------|----------|----------|
| NBANDS1 | 32 | 85 |
| COMPR1 | 1 | 2.2 |
| COMPR2 | 4 | 1 |
| MASK | 1 | 1 |
| WINL | 8 | 6 |
| NBINS | 5 | 5 |

Table 4. Comparison of system parameters

input the linear axis periodicity spectral magnitudes into a Discrete Scale Transform and keep the magnitude. All the other processing steps are the same as for OP_{opt} (except of NBINS, which is specific to the OP computation). Bold numbers in Table 5 indicate significant changes, at a .05 error level, for either OP (Table 5.(b)) or STM (Table 5.(b)). Underlined numbers indicate significant differences between the different features, thus comparing either OP_{opt} with STM_{opt} or OP_{org} with STM_{org} . Thus, bold numbers indicate changes caused by parameter improvement, while underlined numbers indicate differences between OP and STM features.

| Dataset | (a) | | (b) | |
|---------|-------------|-------------|-------------|-------------|
| | OP_{opt} | OP_{org} | STM_{opt} | STM_{org} |
| DBall | 88.4 | 86.1 | 84.1 | 85.1 |
| DLat | 81.0 | 81.8 | 79.7 | 65.2 |
| DPop | 74.4 | <u>68.6</u> | 70.3 | 60.5 |
| DCrete | 70.4 | 64.0 | 68.2 | 61.5 |
| DTurk | 46.2 | 45.2 | <u>56.3</u> | <u>58.3</u> |

Table 5. Classification Accuracies

It can be seen that both for OP and STM descriptors, introducing the improved multiband processing leads to significant improvement in three cases (bold numbers in Table 5). Not surprisingly, for the synthesized melodies in DTurk changed multiband processing shows no effect. Observing the underlined accuracies in Table 5, it can be seen that only for DTurk there is a significant advantage of the STM over the OP features, whereas OP appear to represent a more accurate similarity measure on the three popular music datasets. The superior performance on the popular music datasets related to the small variance in pulse-tempo in the classes. The small set of Cretan dances has similar standard deviations as the DBall (for exact values refer to [8]), but larger overlaps between distributions. The accuracies on this set are not significantly different for OP and STM descriptors (70.4% and 68.2%, respectively). However, in a musical context where we have to face huge variance of tempo for one and the same rhythmic class, such as in Turkish art music, the tempo robustness of STM lead to a significant improvement over OP (56.3% compared to 46.2%).

5. CONCLUSIONS

In this paper, a crucial problem for rhythmic similarity estimation in music was addressed: Depending on the tempo variances inherent in classes of a musical style it is either of advantage to encode larger tempo changes in the descriptors, or to use descriptors that are robust for even large tempo changes. The former case was addressed by descriptors based on Onset Patterns, while for the latter Scale Transform based descriptors were shown to be more

adequate. An advantage of both descriptor types is that no tempo estimation has to be performed on the audio signal, which is an error-prone step in almost all styles of music. Another important contribution of this paper is the improvement of system parameters using an analysis of variance (ANOVA). It is shown that the obtained parameters lead to improvements even for different approaches (STM). The conclusions drawn from the ANOVA are related to the numbers of bands to be used for rhythm description, and the limitation of a STFT analysis window length to 8 seconds. This limitation also limits the possible resolution of the OP descriptors, because for a higher resolution (NBINS) longer windows would be necessary. Thus, the stationarity requirement for the STFT limits the possible parameter space of the OP description. A possible approach to explore the effects of going beyond this border is the usage of transforms that can deal with non-stationary signals.

Acknowledgements

This research was supported by the Austrian Research Fund (FWF), project no. Z159 (Wittgenstein Award), and the Vienna Science and Technology Fund (WWTF), project MA09-024 (Audiominer).

6. REFERENCES

- [1] T. Pohle, D. Schnitzer, M. Schedl, P. Knees, and G. Widmer, "On rhythm and general music similarity," in *Proc. of ISMIR*, 2009.
- [2] G. Cooper and L. Meyer, *The Rhythmic Structure of Music*. University of Chicago Press, 1960.
- [3] G. Peeters, "Spectral and temporal periodicity representations of rhythm for the automatic classification of music audio signal," *IEEE Trans. Audio, Speech, and Language Proc.*, vol. 19, no. 5, pp. 1242–1252, 2011.
- [4] J. Jensen, M. Christensen, and S. Jensen, "A tempo-insensitive representation of rhythmic patterns," in *Eusipco*, Glasgow, Scotland, 2009.
- [5] A. Holzapfel and Y. Stylianou, "Rhythmic similarity in traditional turkish music," in *Proc. of ISMIR*, 2009.
- [6] S. Dixon, F. Gouyon, and G. Widmer, "Towards characterisation of music via rhythmic patterns," in *Proc. of ISMIR*, 2004.
- [7] A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Trans. Acoustics Speech and Signal Processing*, vol. 14, no. 1, pp. 342–355, 2006.
- [8] A. Holzapfel and Y. Stylianou, "Scale transform in rhythmic similarity of music," *IEEE Trans. Speech and Audio Processing*, vol. 19, no. 1, pp. 176–185, 2010.
- [9] R.A. Bailey, *Design of Comparative Experiments*. Cambridge, UK: Cambridge University Press, 2008.
- [10] C.N. Silla Jr., A.L. Koerich, and C.A.A. Kaestner, "The Latin Music Database," in *Proc. of ISMIR*, 2008.

GESTURAL CONTROL OF REAL-TIME SPEECH SYNTHESIS IN LUNA PARK

Grégory Beller
IRCAM, Paris
beller@ircam.fr

ABSTRACT

This paper presented the researches and the developments realized for an artistic project called *Luna Park*. This work is widely connected, at various levels, in the paradigm of the concatenative synthesis, both to its shape and in the processes which it employs. Thanks to a real-time programming environment, synthesis engines and prosodic transformations are manipulated, controlled and activated by the gesture, via accelerometers realized for the piece. This paper explains the sensors, the real time audio engines and the mapping that connects this two parts. The world premiere of *Luna Park* takes place in Paris, in the space of projection of the IRCAM, on June 10th, 2011, during the festival AGORA.

1. INTRODUCTION

Luna Park is a piece of musical theater, of duration about one hour, written by Georges Aperghis, staged by Daniel Levy, and whose computer music is designed by Grégory Beller. The general subject of the piece approaches the way the electronic surveillance and the massive collection of personal digital data make of our current world, a gigantic park of attraction. Four performers are Eva Furrer, octobass flute and voice, Johanne Saunier, dance and voice, Mike Schmidt, bass flute and voice, and Richard Dubelsky, air percussions and voice. The fact that they have all four vocal parts to perform, as well as the scenography of the set (including video), move *Luna Park* closer to a previous work of G. Aperghis, called *Machinations* (2002). However, this new work distinguishes itself from the previous one, notably by gesture sensors' use and from the speech synthesis. Indeed, various sensors, (accelerometers, tactile ribbons and piezoelectric sensors) were developed and realized to allow the performers to control various audio engines, by the gesture. The mapping between the data stemming from these sensors and the various audio processings, realized within the real-time programming environment Max/MSP, is different from a sequence in the other one and can evolve in the time.

That is why this article is presented according to the following plan. In a first part, this article lists the various sensors realized for this creation and gives details of their

developments, as well as the data which they produce. In a second part, the realized audio engines are described under the shape of real-time processes. Besides the concatenative synthesis engine, is presented an innovative engine of prosodic transformation allowing the real time modification of the speech rate. The third part proposes some examples of mapping between the sensors data and the audio engines parameters, notably used for the piece. Finally, the fourth part allows to conclude and to propose some perspectives.

2. GESTURE CAPTURE

2.1 Background

In the case of the acoustic instruments, a gesture is necessary for the sound production. This is not any more the case of the electronic instruments where the control of electronic sound processes is separated from the process of sound production. Several projects of research and creation in the IRCAM use the gesture capture. Whether it is to augment acoustic instruments, (Bogen Lied [1], Augmented Violin Project [2], augmented percussions of Fedele) or to connect different worlds together such as music, video, gesture and dance for the creation (Glossopoeia) or for the pedagogy [3], certain number of sensors was realized in the IRCAM. However, the gesture capture was neither used yet to control prosodic modifications nor vocal synthesizers in the spectrum of the contemporary creation in the IRCAM. It is thus a new way to use the gesture capture systems that we propose for this project of research and creation. The gestural control of the speech synthesis constitutes henceforth a complete field of research. Controllers of various types were elaborated for various types of synthesizers of spoken voice, or sung voice [4]. Among these mappings, we find the "Speech Conductor" [5, 6], the "Glove-Talk" [7], the "Squeeze Vox" [8], the "SPASM" [9], the "OUISPER" project [10, 11] and some others [12]. We chose to use the movements of the hand for several reasons, besides crossing the scenographic reasons. First of all, the spontaneous speech can be naturally accompanied with a movement of hands. The idea to accompany the movements of hands by the speech, by the reversibility, seems thus natural. The percussive aspect of the movements fits the concatenative synthesis in which segments are activated in a discrete way in the time, so managing the segmental aspect of the speech. On the contrary, the continuous aspect of the movements of hands allows a control of the prosody, the suprasegmental aspect of

the speech. If we consider the classic asymmetry right-left such as know it the conductors (for the right-handers, the left hand is rather connected with the expression, whereas the right hand is rather connected with the important temporal markers of the music), we can then create a gestural control of both hands of the synthesis, with for a right-hander, a right hand managing the segmental aspect and a left hand managing the suprasegmental aspect. It is one of the possible scenarios that we exploited for the creation of *Luna Park* (see section 4).

2.2 Accelerometers gloves

The technology of gloves wireless accelerometers / gyroscopes used [13], presented on figure 1 allows to measure the accelerations of both hands according to 6 axes (3 in translation and 3 in rotation with gyroscopes). The raw data delivered by gloves are not necessarily easy to interpret. So a first stage of preprocessing allows to return more interpretable data.



Figure 1. Images of the sensors (to the right) and of Richard Dubelsky who wears them in hands thanks to gloves (to the left).

2.2.1 Preprocessing

The data resulting from the wifi receiver are transmitted via UDP every 1 ms. To synchronize them to the internal clock of Max/MSP, they are first median filtered (order 5) and sub-sampled by a factor 5. Thus we obtain a stable stream of data every 5 ms. Then various descriptors of the gesture arise from these preprocessed raw data.

2.2.2 Variation of the momentum

The estimate of the immediate acceleration allows to know, at any time, the variation of momentum relative to the gesture. This momentum, according to the laws of the classical mechanics, is directly proportional in the speed. The raw data coming from the sensor are at first “denoised” thanks to the average on the last 4 samples. The root of the

sum of the square of these six filtered values allows to obtain a proportional quantity in the variation of momentum of the gesture.

2.2.3 Hit energy estimation

The hit energy estimation allows the immediate release from the observation of the variation of the momentum of the gesture. Three values, delivered by the sensors of acceleration in translation, are stored in a circular buffer including all the time, 20 samples. Three standard deviation corresponding to these values are added, all the time, (norm I corresponding to the sum of the absolute values). This sum also allows to represent the variation of momentum of the gesture. To detect variation of this value, corresponding to abrupt variations of the gesture, it is compared all the time with its median value (order 5). When the difference between these two values exceed certain arbitrary threshold, a discrete value appears to mean the presence of a fast change of the gesture. It allows, for example, to emit a regular click, when we beat a measure with the hand, every time the hand changes direction. The hit energy estimation is a process allowing to generate discrete data from a gesture, by definition continuous. Indeed, of a continuous physical signal, it allows by thresholding, to define moments corresponding to the peaks of variation of the momentum, which coincide, from a perceptive point of view for the user, in peaks of efforts (of acceleration). By this process, it becomes then possible to create precise air percussions either sounds activation at the moment when the hand of the user changes direction or accelerates surreptitiously.

2.2.4 Absolute position of the hand

The Earth’s gravitational field introduces an offset into the answer of the sensors which can be exploited to deduct the absolute position of the hands, as well as the presence of slow movements. This quasi-static measure brings a continuous controller to the performer. A playful example of the use of this type of data is the air rotary potentiometer in which the rotation of the hand can control the volume (or other) of a sound.

2.3 Piezoelectric sensors

Besides the accelerometers gloves, the percussionist plays physical percussions. He emits sounds by striking certain zones of his body and/or the structure surrounding him. The integration of local and sensitive zones to the touch, on a suit, is not easy. The traditional pads of an electronic drum kit are too stiff and non-adapted to hand striking. We chose to use smaller and more flexible, piezoelectric microphones. Two microphones (one near the left hip and the other one near the right shoulder) placed on the percussionist allow him to play with two different zones of his body. Six microphones of the same type are also arranged in its surrounding space. The audio signals delivered by these various microphones are processed by a classical attack detection system allowing the percussionist to activate various types of sounds according to zones. Likely if the

pads of an electronic drum kit was arranged on and around the percussionist.

3. AUDIO ENGINES

3.1 Real time concatenative synthesis

The concatenative synthesis is realized in real time thanks to the object *Mubu.concat* developed in association with the team Real-Time Musical Interaction of the IRCAM [14, 15]. This object contains many functionalities of the *cataRT* patch [16]. The object takes, as input, a sound file (buffer) and an associated markers file. He allows the reading of segments by the choice of their indexes. It also includes other options affecting the reading, such as the transposition, the windowing, or still the used output. Segments can succeed one another automatically or be launched by a metronome or another discrete signal emitted by a sensor (stemming from the hit energy estimation, for example). The order of segments is arbitrary and all the sequences of index are possible. Among them, an incremental series of step 1 will restore the original audio file without audible presence of the segmentation, whereas a random series will generate a variation of the starting material and will make audible the beforehand chosen segmentation. Various methods to generate interesting sequences of index within the framework of the speech synthesis are presented below.

3.2 Speech synthesis

If the audio engine of concatenative synthesis does not need to be sophisticated, compared with the other paradigms of synthesis, such as the HMM-based speech synthesis or still the articulatory synthesis, it is because the intelligence of the concatenative speech synthesis rests on the selection of segments, that is, on the definition of the sequence of the indexes. Indeed, the concatenative speech synthesizer bases on two distances allowing to define simultaneously the closeness of the result with regard to a target (distance of the target) and the quality of this result (distance of concatenation).

3.2.1 Distance of the target

The first distance, as its name indicates it, requires the definition of a target. In the Text-To-Speech synthesis, this target is defined in a symbolic way by the text and by the various analyses which derive from it (grammar, phonetics...). In the hybrid synthesis speech/music [17], a target can be defined in an acoustic way as a sequence of audio descriptors. Any targets may be used as long as it shares with segments, a common descriptor. The synthesizer *IrcamTTS* [18, 19], presents a peculiarity face to face the other TTS (Text-To-Speech) synthesizers, because he allows the user to define his target in a symbolic way, by the text, but also in an acoustic way, by some prosodic symbols. So the user can write on the same support, in a joint way, the wished text and the way he would like this text is pronounced. This tool is very appreciated, consequently, by composers who can write not only the text, but also the

prosody they wish, like a score. The target can be also defined in real time.

3.2.2 Distance of concatenation

The distance of concatenation allows to estimate the perceptive weight of the concatenation of two segments. Naturally consecutive segments cause a zero weight. Segments from which spectrum in edges are very different, will produce, a higher weight, supposed to mean the introduction of an synthesis artifact.

3.2.3 Speech synthesis in batch mode

As the definition of a real time target is not a common matter, most of the TTS synthesizers work in batch mode. The user writes a sentence, then chooses generally a voice, to pronounce it. The most spread selection algorithm of segments then appeals a Viterbi decoding allowing the joint minimization of the target distance and the distance of concatenation on the whole sentence to synthesize. The “backward” phase of this algorithm allowing the definition of the optimal solution requires to know the end of the sentence to select the beginning, what makes the synthesis engine profoundly not real-time.

3.3 Real time speech synthesis

A real time speech synthesizer has sense only if the text is generated in real time. The case appears (as shown in section 4) when the text is generated by statistical models (HMM, N-gram, K-NN) which transforms or generates one text on the fly. The real-time constraint does not allow us any more to use the phase “backward” of the Viterbi algorithm guaranteeing the optimality of the path on a whole sentence, because we do not know early, the end of the current sentence. The joint minimization of the distance in the target and the distance of concatenation can be made then only between every segment, in a local way and not in a global one. The advantage of this method lies in its ability to react, while its inconvenience is that it produces a sound result poorer than the classic batch TTS synthesis. In *Luna Park*, several paradigms are used to generate, in real time, targets which guide the concatenative synthesis.

3.3.1 Predefined sequences

First of all, certain texts are fixed, a priori, and modeled under the shape of a sequence of words, syllables, phones or semi-phones. Segments constituting these sequences are then launched by the performers and can be chosen according to data stemming from sensors or from vocal analyses (see section 4). It allows for expected syllables series, and to produce a clear semantic sense.

3.3.2 Predefined sequences and N-gram

It happens that several segments possess the same symbol (several phones corresponding to the same phoneme, for example). We can then estimate the probability of transition from a symbol to the other one and make random series respecting more or less the initial sequence. Thanks to N-gram of order N variable, we can control in real time the closeness of the generated sequence with regard to the

predefined text (the bigger N, the closer to the initial sequence; the smaller N, the further to the initial sequence). It notably allows the performers to control the semantic aspect of the output.

3.3.3 Predefined sets and HMM

Like a text, some segment sets (syllables, phones, words) were also predefined. In the same way, segments belonging to these sets are activated by the performers according to descriptors or in a random way. It notably allows to create textures (of phones or syllables) or still rhythms from a single syllable. An interface was create to allow to choose in real time, the probability of appearance of such symbols. It appears under the shape of a histogram of the available symbols. The modification of this histogram allows to modify the probability of transition from a symbol to the other one (HMM of order 1). Once the symbol is chosen, a corresponding segment can be activated according to various controllers / descriptors values. The figure 2 presents the interface that permits to generate in real time targets. At the top, a recording segmented in syllables thanks to the program IrcamAlign [20] which allows the speech segmentation in various units in batch mode), is the input. In the middle, the histogram presenting the relative rate of appearance of their symbols. Below, the same modified histogram allows to modify the probability of appearance of the activated syllables. For example, the generated output is composed of an equiprobable set of syllables “ni”, “naR” and “noR”.

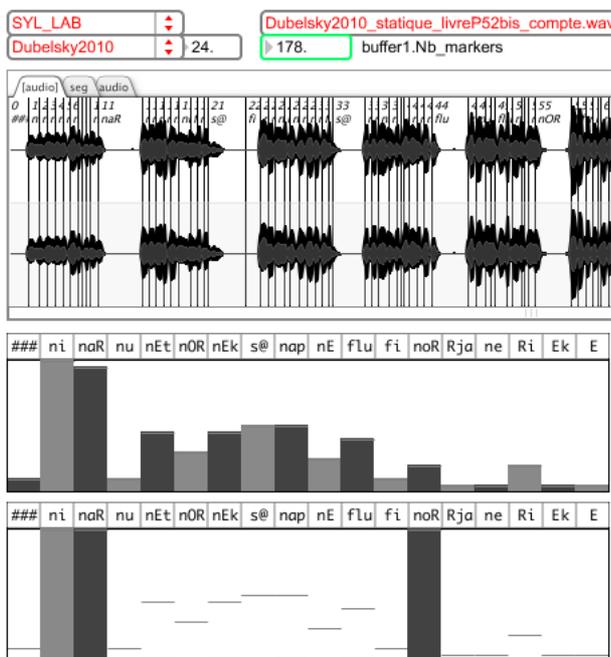


Figure 2. Interface for the definition of the real time target by editing histogram.

3.4 Real time prosodic transformation

Real time prosodic transformation allows many applications [21]. It is possible thanks to speech analysis/synthesis paradigm allowing to estimate and to modify the prosodic dimensions of the speech. The prosodic transformation can apply to the voice of the performers or to the synthesis output in a equivalent way. The prosody or the way to speak can be described by a space in five dimensions: the intonation, the intensity, the speech rate, the vocal quality and the degree articulation [22]. If most of these dimensions are today measurable in batch mode [23], some of them are also measurable in real time. It is the case of the intonation (yin [24]) and of the intensity (loudness). We added a speech rate estimator (syllex). These three real-time speech descriptors allow to inform us about the way the performers utter. They can be used, in the same way as the sensors, to control the various audio engines. If the modification of the intonation by transposition and that of the intensity by variable gain are henceforth known and well enough mastered in real time, it does not also go away for the speech rate. Now we expose in this section, one new paradigm allowing the transformation of the speech rate in real time. All the prosodic transformations used are available in the SuperVP [25] audio engine, the IRCAMs high quality phase vocoder in real time. In fact, the SuperVP library already implements a quality transposition, as well as some other modifications such as the spectral envelope transformation. One of the objects of this library, SuperVP.ring, allows to make these modifications on a circular buffer the size of which can be arbitrarily defined (3 seconds in our case). The advantage of the circular buffer is to keep the instantaneousness of the transformation, while enabling, at any time, to be able to move in short-term past (term equivalent to the size of the buffer). Thanks to it, we can locally stretch out certain portions of the signal as the vowels (using a real time voicing detection) and provoke at the listener’s the perception of a slowing down of the speech rate. If we cannot move to the future, the return in the immediate position can be made in a accelerated way, provoking, this time, the perception of an acceleration of the speech rate. As if the read head of the buffer behaved as an elastic which we stretch out and relax. Extremely, it is possible to freeze the read head of the buffer in a place that provokes a “stop on sound” who can give interesting effects (extremely long vowels for example that makes speech sounds like sing).

4. EXAMPLES OF MAPPING

In this part, we give some examples of mapping between the control data and the parameters of the audio engines. The figure 3 lists the various available controllers (to the left), as well as the various parameters of the audio engines (to the right). The mapping consists in connecting the controllers with the parameters (by some linear or non-linear scales) and it can vary in the time, as it is the case in *Luna Park*. Two types of connections are possible: the discrete connections and the continuous connections. Indeed, the discrete controllers (underlined on the figure 3), giving only a value from time to time, as the hit energy

estimator, correspond to the control of type percussive and are going to serve for controlling the discrete parameters of the audio engines, as the activation of a segment for the concatenative synthesis (highest arrow). On the contrary, a continuous connection connects a continuous controller, as the linear absolute position on a tactile ribbon in a continuous parameter of audio engines such as the transposition, for instance (lowest arrow).

Some scenarios chosen for the piece are described be-

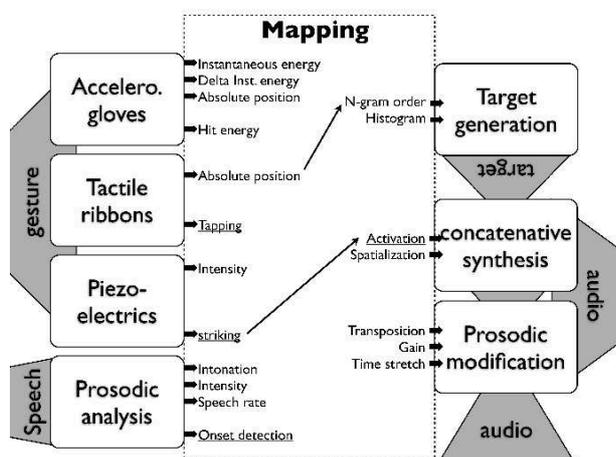


Figure 3. The mapping is at the interface between the data stemming from controllers (to the left) and the parameters of the audio engines (to the right). As example, two arrows were drawn. The highest allows to make vary the order of N-gram using a tactile ribbon. The lowest allows to activate a segment of the concatenative synthesis by a striking on the body.

low. By connecting in a direct way the hit energy estimator of the right glove of the percussionist (who is right-handed) with the activation of the synthesis, he can manage the speech rate and segmental aspects by percussive movements of the right hand. If the rotation of the left hand is connected with the transposition and with the intensity of the synthesis, he can then control the prosody of this one with both hands. In a scene of the piece, a dancer caresses a tactile ribbon situated over a screen transverse. The speed of her caress, deduced from the detected absolute linear position of her finger, is then connected with the speech rate modification of another performer's speech, which is speaking at the same moment. Thanks to a mapping favoring the catch of the vowels, she can manage to make "a stop on sound" in a vowel which gives the effect that she hold his tong. The accumulation of the energy of the hands of the percussionist is used, in a scenario, to control the N-gram order of the generation of targets. The activation of the segments is here automatic (continuous stream where every end of segment activates the next one) and controller changes only the order in which segments are chosen. The more the percussionist is energetic and the more the order of N-gram decreases, going to the random for large-scale movements.

5. FUTURE PERSPECTIVES

Of this period of research can be deduced several perspectives concerning the gesture capture, the real time speech synthesis, as well as their connections. Concerning the gesture capture, the used sensors possess the advantage to be rather light to be integrated into gloves, as well as good sensitivity in the movement (capture of the variation of the momentum is accurate). On the other hand, they present a rather low energy autonomy, and do not offer the measure of the absolute static position. It would be beneficial to add to accelerometers, another technology allowing to access the absolute position. As regard the audio engine, it would be interesting to bend over the other speech synthesizers based on parametric (articulatory), semi-parametric (HMM) or hybrid models (concatenative/HMM). Indeed, the concatenative synthesis in batch mode has for advantage its degree of realism, which becomes difficult to maintain in real time. Finally the mapping between the gesture and the speech synthesis is a rich subject of research. As a research track, we can imagine more complex mappings where become interleaved the temporal and the semantic aspects of both the hand gesture and the vocal gesture.

6. ACKNOWLEDGMENTS

Author would like to thank Fred Bevilacqua, Bruno Zamborlin, Norbert Schnell, Diemo Schwarz, Riccardo Borghesi, Emmanuel Fléty, Maxime Le Saux, Pascal Bondu, Xavier Rodet, Christophe Veaux, Pierre Lanchantin and Jonathan Chronic, for their helps.

7. REFERENCES

- [1] S. Lemouton, "Utilisation musicale de dispositifs de captation du mouvement de l'archet dans quelques oeuvres récentes," in *JIM*, 2009.
- [2] F. Bevilacqua, N. H. Rasamimanana, E. Fléty, S. Lemouton, and F. Baschet, "The augmented violin project: research, composition and performance report." in *NIME*, 2006.
- [3] F. Bevilacqua, F. Guédy, N. Schnell, E. Fléty, and N. Leroy, "Wireless sensor interface and gesture-follower for music pedagogy," in *NIME*, 2007, pp. 124–129.
- [4] P. Cook, "Real-time performance controllers for synthesized singing," in *NIME*, 2000.
- [5] C. d'Alessandro, N. D'Alessandro, S. L. Beux, J. Simko, F. Cetin, and H. Pirker, "The speech conductor: gestural control of speech synthesis," in *eNTERFACE*, 2005.
- [6] N. D'Alessandro, C. d'Alessandro, S. L. Beux, and B. Doval, "Real-time calm synthesizer: New approaches in hands-controlled voice synthesis," in *NIME*, 2006, pp. 266–271.

- [7] S. Fels and G. Hinton, "Glove-talk 2: A neural network interface which maps gestures to parallel formant speech synthesizer controls," *IEEE Transactions on Neural Networks*, vol. 9, no. 1, pp. 205–212, 2004.
- [8] P. Cook and C. Leider, "Squeeze vox: A new controller for vocal synthesis models," in *ICMC*, 2000.
- [9] P. Cook, "Spasm: a real-time vocal tract physical model editor/controller and singer: The companion software synthesis system," *Computer Music Journal*, vol. 17, no. 1, pp. 30–34, 1992.
- [10] T. Hueber, G. Chollet, B. Denby, G. Dreyfus, and M. Stone, "Towards a segmental vocoder driven by ultrasound and optical images of the tongue and lips," in *Interspeech*, 2008, pp. 2028–2031.
- [11] B. Denby and M. Stone, "Speech synthesis from real time ultrasound images of the tongue," in *ICASSP*, 2004, pp. 685–688.
- [12] A. Esposito, M. Faundez-Zanuy, E. Keller, M. Marinaro, B. Kröger, and P. Birkholz, "A gesture-based concept for speech movement control in articulatory speech synthesis," *Verbal and Nonverbal Communication Behaviours*, no. 4775, pp. 174–189, 2007.
- [13] E. Fléty and C. Maestracchi, "Latency improvement in sensor wireless transmission using ieee 802.15.4," in *NIME*, 2011.
- [14] N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Müller, "FTM-Complex Data Structures for Max," in *ICMC*, Barcelona, Spain, Sep. 2005.
- [15] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, and R. Borghesi, "Mubu and friends - assembling tools for content based real-time interactive audio processing in max/msp," in *ICMC*, 2009.
- [16] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-time corpus-based concatenative synthesis with catart," in *DAFx*, 2006.
- [17] G. Beller, D. Schwarz, T. Hueber, and X. Rodet, "Hybrid concatenative synthesis in the intersection of speech and music," in *JIM*, A. Sedes and H. Vaggione, Eds., vol. 12, 2005, pp. 41–45.
- [18] C. Veaux, G. Beller, and X. Rodet, "Ircamcorpustools: an extensible platform for spoken corpora exploitation," in *LREC*, Marrakech, Morocco, may 2008.
- [19] G. Beller, C. Veaux, G. Degottex, N. Obin, P. Lanchantin, and X. Rodet, "Ircam corpus tools: Système de gestion de corpus de parole," *TAL*, 2009.
- [20] P. Lanchantin, A. C. Morris, X. Rodet, and C. Veaux, "Automatic phoneme segmentation with relaxed textual constraints," in *LREC2008*, Marrakech, Morocco, 2008.
- [21] G. Beller, "Transformation of expressivity in speech," *Linguistic Insights*, vol. 97, pp. 259–284, 2009.
- [22] ———, "Analyse et modèle génératif de l'expressivité : Application à la parole et à l'interprétation musicale," Ph.D. dissertation, Université Paris XI, IRCAM, June 2009.
- [23] G. Beller, D. Schwarz, T. Hueber, and X. Rodet, "Speech rates in french expressive speech," in *Speech Prosody 2006*, SproSig. Dresden: ISCA, 2006, pp. 672–675.
- [24] A. D. Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *JASA*, vol. 111, pp. 1917–1930, 2002.
- [25] A. Roebel, F. Villavicencio, and X. Rodet, "On cepstral and all-pole based spectral envelope modeling with unknown model order," in *PRL*, 2006.

AN INTERACTIVE SURFACE REALISATION OF HENRI POUSSEUR'S 'SCAMBI'

Robin Fencott

Queen Mary University of London
Interaction, Media and Communication Group
RobinFencott@eecs.qmul.ac.uk

John Dack

Middlesex University
Lansdown Centre for Electronic Arts
J.Dack@mdx.ac.uk

ABSTRACT

We have constructed an interactive touch surface exhibit to re-appropriate a historic electroacoustic composition for the digital age. The electroacoustic work in question is Henri Pousseur's seminal composition 'Scambi', originally created in 1957 at the RAI Studios, Milan. The status of Scambi as a key example of an electroacoustic 'open' form makes it ideal for re-appropriation as an interactive public exhibit, while an existing musicological analysis of Pousseur's compositional instructions for Scambi provide insight for the user interface design and translation of written textual composition process into interactive software. The project is on-going, and this paper presents our current work-in progress. We address the musicological, practical and aesthetic implications of this work, discuss informal observation of users engaging with our tabletop system, and comment on the nature of touchscreen interfaces for musical interaction. This work is therefore relevant to the electroacoustic community, fields of human computer interaction, and those developing new interfaces for musical expression. This work contributes to the European Commission 'DREAM' project.

1. INTRODUCTION

DREAM is a European Commission funded cross institutional and multidisciplinary project exploring the Digital Reworking and re-appropriation of ElectroAcoustic Music. The overarching goal of DREAM is a permanent exhibition housed in the Milan Museum of Musical Instruments, to celebrate and document the highly influential Studio di Fonologia Musicale (RAI, Milan, Italy). One of our contributions to the DREAM project is a interactive tabletop implementation of 'Scambi', an electroacoustic work created by Henri Pousseur in 1957 at The Studio di Fonologia Musicale della RAI Milano (see figure 1). This exhibit aims to demonstrate to visitors, through active engagement, the process of creating an electroacoustic composition, and demonstrate to visitors some of the ways in which music technology and compositional practice has changed since the 1950s. In addition to the technical contribution our work makes to the DREAM project, it is also

Copyright: ©2011 Robin Fencott et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.



Figure 1. People creating a realisation of Scambi on the interactive surface.

within our remit to assess the feasibility and implications of re-appropriating historic, analog electroacoustic composition for public exhibition using modern technologies.

1.1 Scambi and the 'Open' form

Scambi, created in 1957, is an electroacoustic work which exemplifies the notion of an 'open' form. An open form is a work left to some degree underspecified by the author, so as to create a situation in which multiple distinct instantiations can be realised by other people. Although it can be argued that scored music is always 'open' to a certain degree (for instance dynamics are often imprecisely described) [1], composers in post-war Europe explored many and varied forms of openness in musical composition [1, 2]. A discussion of the philosophical and musicological intricacies of open form in art or music is clearly beyond the scope of this paper, and we suggest [3] as a starting point for the

curious reader. The rest of this section provides a more concrete discussion of Pousseur's Scambi.

The sonic materials of Scambi are a collection of 32 sound segments, each approximately 36 or 42 seconds in length. Pousseur sculpted these pieces of audio by running white noise through processes such as amplitude filtering, reverberation and tape-speed modulation. Originally these sections were stored on lengths of magnetic tape.

Pousseur identified four parameters within his sonic materials; *Relative Pitch*, *Speed*, *Homogeneity* and *Continuity*, and these were used to describe the starting and ending conditions for each sound segment. By joining together sound segments with matching start and end conditions, multiple sonic compositions could be assembled, with the connecting rule (matching start and end conditions) ensuring seamless joins between segments. Pascal Decroupet's musicological analysis of Scambi [4, 5] uses a notation system of 1s and 0s to describe the start and end of each audio segment; relative pitch (low 0 to high 1), the statistical speed (slow 0 to fast 1), the homogeneity of sound material (dry 0 to reverberated 1) and continuity (inclusion of pauses 0 to continuous sound 1). In Decroupet's system, a segment starting 1111 would begin high, fast, reverberated and continuous, while a sound ending 0100 would be low, fast, dry and include pauses.

Although Pousseur was interested in the idea of total continuity between sections [6] he also noted that the connecting rules are 'but a guide to the making of a unified whole, it being left open to assemble a meaningful event without their help' [6]. Composers are not obliged to use all sections, there are no constraints on the length of the composition, and polyphonic structures are permitted, whereby multiple segments are played simultaneously. Clearly with this small set of guidelines and sonic materials, a vast array of potential configurations are made available to composers. Alongside Pousseur, composers Luciano Berio and Marc Wilkinson created realisations of Scambi at the RAI studio, using magnetic tape and analog equipment. Recently a number of composers have created realisations using digital audio software as part of the UK Arts and Humanities Research Council funded 'Scambi Project' [3]. Both of these approaches require a certain degree of technical expertise. Our work here aims to further simplify the realisation of Scambi, and in fact takes as inspiration Pousseur's imagining (in 1959) of an environment in which people can create realisations of Scambi in a social context [6]. More detailed notes on the realisation of Scambi are found in [5].

2. RELATED WORK

2.1 Analog Emulation

The music software industry has marketed the concept of virtual-analogue technologies to emulate classic pieces of music technology. Software such as Propellerhead's 'Re-Birth' and 'Reason' present users with interfaces that visually resemble sought-after synthesisers and audio effects. As well as attempting to provide a faithful reproduction of the sounds created by these devices, these applications

(and many others like them) use virtual buttons, LED displays and on-screen dials to visually resemble and recreate the user interaction experience of their physical counterparts. These forms of physical controls and user feedback were a necessity for the original physical hardware devices. Within the software domain choices about the interaction design are more open to negotiation as the sound production is not constrained by the physical characteristics of an electronic circuit, while the interaction metaphor [7] is constrained only by the imagination of the designers, and the specification of the machine the software is intended to run on.

This discussion is included to highlight the importance of considering interface design and metaphor when emulating or attempting to recreate a physical or tactile musical interface in an on-screen software environment. A literal, visually faithful recreation of the user interface may be more immediately recognisable, yet may not take full advantage of the affordances or capabilities of the new medium. We believe this debate is as relevant for the re-appropriation of electroacoustic work as it is for the design of analog emulation technologies, and we return to this discussion in section 4.

2.2 Interactive Surfaces for Music and Exhibition

Touch surfaces for musical interaction, performance and composition are a rapid field of expansion, with touchscreen mobile telephones and tablet computers becoming commonplace tools for musicians [8]. At a larger physical scale, the *reactTable* [9] has captured a great deal of public attention. The *reactTable* allows musicians to collaboratively patch together sound generators and processors by manipulating and arranging small physical objects on a rear-projected tabletop interface. These physical objects represent different sound generators and processors; with position, rotation, and proximity to one-another mapped to various synthesis parameters. Similar physical object based interfaces for music-making include *BlockJam* [10] and *Audiopad* [11], while [12] presents a multi-touch music environment based entirely on direct touch instead of tangible objects. Regardless of whether these systems use direct touch or tangible object based interaction, a key feature of surface interfaces is the provision for multiple points of interaction by *one or more people simultaneously*. Additionally such large-scale interfaces can be used within musical performances as a spectacle that bridges the gap between a performer's physical gestures and the music being created.

Aside from the interactive surface interfaces based on the paradigms of computer music software (on-screen oscillators, musical keyboards, sliders and so on), surfaces are ideal for placement in public contexts [13] where accessibility and immediacy are central concerns, and furthermore it has been noted that members of the public do not usually associate interactive surfaces with conventional forms of computer interaction [14]. Examples of interactive surfaces designed for playful engagement in public exhibitions include Fencott's interactive cellular automata [15] and Iwai's 'Composition on the Table' [16], both of which

leverage the potential of interactive surfaces to support direct intuitive engagement with sonic and visual materials in a manner which is distinct from conventional music-making techniques or tools.

2.3 Preservation of Electroacoustic Music

Electroacoustic works are often tied intrinsically to the technologies employed in their realisation. The preservation of electroacoustic works is therefore a problematic issue for musicologists, historians and composers alike. [17] observes that due to obsolescence, many compositions and performances are impossible to repeat without major reconstructive work to rebuild systems and port code to new platforms, for instance Arfib's implementation of Music V synthesis algorithms [20] for gestural control. However for the majority of composers, written, audio and visual documentation are the only methods of preserving their work for future generations. There are already many projects dedicated to the task of documentation, for instance [18] [19]. However for an open form to *remain open*, documentation alone may not be satisfactory. Rather, it is crucial that the means of producing new instantiations is preserved. It is testament to Pousseur's own written documentation, the extensive research surrounding his methods, and the conservation of his original audio materials that our work is made possible.

3. IMPLEMENTATION

This section discusses our implementation of the interactive surface Scambi interface. We first give a brief overview of the physical table interface constructed for the project, which serves as a prototype for software development purposes and will be used in several public exhibitions in the UK. We then move on to discuss the software design decisions and the influence of existing musicological research on Scambi. Video and additional documentation is available on the first author's website [21].

3.1 Hardware

We constructed a computer vision based multi-touch table. In this approach, a camera views the underside of the touch surface, and computer vision techniques are used to identify the location of fingers and objects on the surface. A predefined 'Fiducial marker' needs to be attached to the underside of any object which is to be tracked. In our system tangible objects were constructed by glueing the fiducial markers to acrylic tiles (see figure 3).

Our touch surface is a 5mm clear acrylic sheet. A short throw data projector is used to back-project onto the touch surface, with a sheet of 1mm matt translucent plastic affixed to the underside of the acrylic as a projection surface. For finger and object tracking the touch surface is illuminated with six infrared (IR) emitter arrays; a technique often referred to as Direct Illumination (DI). Each IR array comprise of up to 32 Osram SFH485P infrared emitters (see figure 2). A Firewire Unibrain Firei camera with a wide angle lens and daylight blocking filter is mounted next to the projector to view the touch surface.

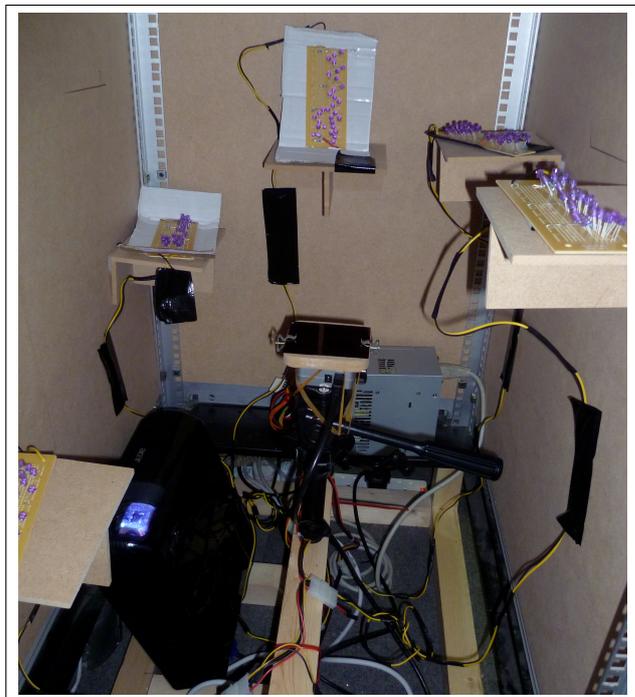


Figure 2. Internal components, including camera with daylight filter, data projector (bottom left) and IR emitters.

The matt finish of the projection surface also helps eliminates 'hot-spots' of reflected infrared light being detected by the camera. Computer vision and fiducial tracking is handled with reactIVision [22]. This application transmits position information about fiducial markers and direct finger touch using the TUIO Open Sound Control protocol to the Scambi Sequencer application (see 3.2). The whole system runs on an Apple G5 Power Mac.

Several publications document the construction of interactive surfaces [23, 24], so rather than re-iterate these details we move straight to the implementation of the Scambi Sequencer software. Further reflections, lessons and documentation about our construction process are given on the first author's website [21].

3.2 Scambi Surface Sequencer

The Scambi Sequencer allows multiple participants to create realisations of Scambi by arranging tangible objects on the interactive table surface. The Scambi sequencer was written in C++ OpenFrameworks [25] using the ofxTUIO addon library [26] to receive TUIO messages from reactIVision.

As described in 1.1, Scambi comprises of 32 different audio segments which can be arranged to form many different compositions. In the Scambi Sequencer, sections are represented as on-screen waveforms (see figure 3), and associated to a fiducial marker. Placing and removing fiducial markers on the table surface enables the dynamic creation and deletion of Scambi sections within the composition. Sections can be arranged spatially on the table, although their coordinate position is not mapped to a parameter. Duplicate sections are permitted, and as many can be added to a composition as will fit on the table surface.

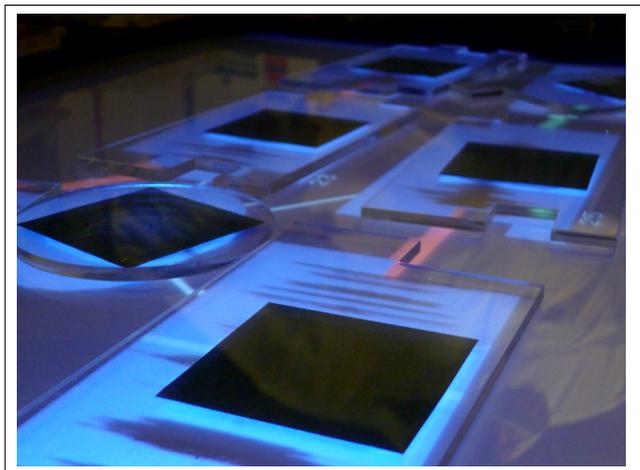


Figure 3. The Scambi interactive surface.

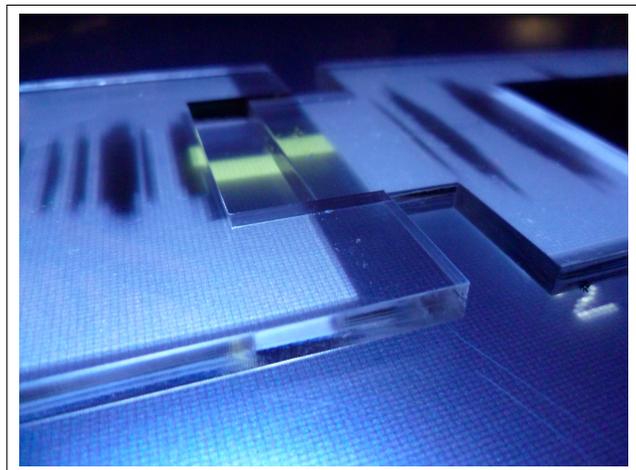


Figure 4. Tessellating acrylic tiles.

It is important to stress that working with the sound segments in this way is entirely different from the approach facilitated by the RAI studios. For instance, the visual waveform representation of the audio was not available to composers, who would have instead relied upon their listening skills to become familiar the sounds. Also, while our design allows for multiple instances of the same sound segment by adding two markers to the table, composers in the 1950s would need to manually duplicate lengths of analogue tape to achieve the same effect.

The musicological notation developed to represent the start and end conditions (unintentionally) resembles the binary number system. Taking advantage of this, the start and ending conditions for each sound segment were copied from [5] and stored as unsigned integer values (e.g., a section starting ‘0100’ was represented as integer value 4). Equality testing and bit-masking operations could then be used as convenient mechanisms to determine the degree of match between sound segments.

Given Pousseur’s liberal attitude towards adherence of his matching system, we felt it appropriate to indicate, rather than enforce the connecting rules within our software interface. This gives participants the opportunity to fully explore different combinations of sounds within the Scambi composition, while still drawing attention to Pousseur’s original intentions. We use a jigsaw or puzzle metaphor to visually imply the start and ending conditions of each Scambi section. The jigsaw shapes are laser cut into the acrylic tiles, and are echoed in the graphical projections on the table. Sections matching on all four of Pousseur’s parameters (pitch, speed, homogeneity and continuity) visually and physically tessellate (see figure 4). Sections placed within a pre-defined proximity to one another on the table surface are automatically joined via connecting lines. Matching sections are connected by a single thick green line, while sections that are not fully matched are joined with a faint red line. These design decisions were made for several reasons. Firstly, we wanted to guide users towards Pousseur’s ideal of complete continuity between sections, while in no way restricting people from exploring more discontinuous configurations. Secondly, we wanted to im-

ply the activity of joining sections through the physical affordances of the physical tiles and their projected graphical representations.

Playback of sound segments is started or stopped by touching the projected waveform. Visually, playing runs from the left to right and the current position is indicated a vertical bar. When a segment finishes playing it automatically triggers the playback of any segments connected to its right-hand edge. Sound sections can be added and removed at any point in the interaction, and there is no limit to the number of sounds concurrently playing.

3.3 Real-time Manipulation

The Scambi sequencer allows users to manipulate the audio playback using additional fiducial objects. The stereo position and volume can be altered using the Pan and Volume objects. These objects maps their rotation value to all Scambi sections within a pre-defined proximity. The playback speed and pitch of Scambi sections can be controlled using the Pitch object (see figure 5). Up to a half-speed decrease is permitted by the object, in line with Pousseur’s suggestion that the original material can be lowered by an octave without losing interest [6]. With these controls, we remained sensitive to the historic and technological context of the original Scambi, by mimicking the operations available to composers working in the RAI Studios. For instance the pitch control object mimics the behaviour of tape-speed manipulation by altering both speed and pitch.

To avoid discontinuous jumps between extreme values the manipulation objects use a circular mapping scheme. In the case of the Pan, both 0 and 180 degree positions represent the stereo centre, while 90 degrees indicates hard right panning and 270 degrees is hard left. (see figure 6).

4. OBSERVATIONS OF USE

During development we invited people to use the surface interface and give feedback on their experiences of it. Many of these people were undergraduate Sonic Arts and Fine Arts students from Middlesex University. As observers, we took written notes and conducted informal conversations.



Figure 5. Real-time pitch manipulation.

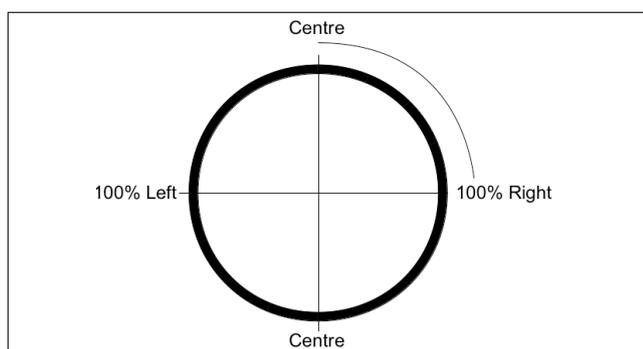


Figure 6. Circular mapping for rotational Pan object.

We answered any questions they had during the course of the interaction and explained the purpose of the project, but gave minimal instructions for how to use the table, as we were interested in witnessing participant's initial encounters with the exhibit, and observing their intuitions about the interface functionality. Reactions were generally very positive, although our attention was drawn to a number of problematic issues, which relate specifically to our interface design, yet may be applicable more generally to interaction with tabletop and surface interfaces.

Our interface uses a mixture of direct touch (fingers touching the surface) and interaction via tangible objects with fiducial markers. Although touch screen interfaces are becoming more common (mobile phones, kiosks, etc), often, our participants did not immediately realise the surface supported both forms of interaction. In our implementation finger tracking is much less reliable than fiducial tracking, although this might be addressed with additional IR lighting. Where touch was unresponsive, participants would often try pressing harder on the screen, although this has little effect, while in overly sensitive areas, touch points were often detected when fingers were 'hovering' over the surface, rather than in contact with the screen. As the direct illumination approach to touch sensing (see section 3) does not require contact with the screen this issue is difficult to eliminate entirely.

Initially, participants appeared more focused on the properties of the surface interface than the Scambi composi-

tion itself. They often experimented with adding sounds to the surface and moving objects around. Some participants were already aware of Scambi, and usually progressed to creating compositions or asking about the process of matching sound segments together. Those who were not familiar with Scambi required more information about the composition process before they understood the purpose of the exhibit. Clearly for a stand-alone exhibit, additional information would be useful to aid visitors in appreciating the work. This information could be provided in written form, or could be included in the Scambi Sequencer itself, as a structured 'training' mode that guides participants through the process of adding sounds, combining them and manipulating them.

A potential pitfall of re-contextualising historic electroacoustic using modern technology is that the new technologies may overshadow or distract from the original work. In our case, some participants using an early version of the system were confused by the fiducial symbols, and misunderstood the relationship between sound sections, on-screen waveforms and the printed markers. Some participants assumed the fiducials were created by Pousseur to represent the the Scambi sounds, while others thought the fiducial symbols stored the audio in some way. This confusion highlights an important factor to consider when designing systems using marker based object tracking. The symbols are visually striking and it is easy to see how they could be interpreted as a significant or salient aspect of the work, however they are but a component of the sensing apparatus, and irrelevant from a user's perspective. In our case this confusion was resolved by simply concealing the markers as much as possible, although we make this point more broadly to stress the sensitivity one must have when re-appropriating historic works using new technologies, especially where the infrastructure of the presentation platform can become confused with the aesthetic experience of the work itself.

5. CONCLUSIONS

This paper has described the development of an interactive surface exhibit which revisits Henri Pousseur's electroacoustic open form 'Scambi'. The open nature of Scambi and the flexibility implied by Pousseur's documentation of the work make it ideal for re-appropriation as an exhibit, while an abundance of work concerning the construction of interactive surfaces allowed us to focus on interaction design concerns, which were informed by existing musical research on Scambi. We have reflected on observations of users interacting with the exhibit, and highlighted consequential design issues for both touch-screen interaction and the re-appropriation of historic electroacoustic works more generally. Our Scambi sequencer was presented at a public exhibition in early May 2011 at Middlesex University, alongside live performances and fixed-format realisations of Scambi and other open forms by Henri Pousseur.

6. ACKNOWLEDGEMENTS

The DREAM project is part of the Culture Programme 2007-2013, and funded by the European Commission Education, Audiovisual and Culture Executive Agency. Table construction assisted by staff of the Middlesex University 3D Workshop and Peter Williams of the Digital Media Workshop.

7. REFERENCES

- [1] J. Dack, "The 'open' form – literature and music," 2005, presented at 'Scambi Symposium'. <http://www.scambi.mdx.ac.uk/Documents/Symposium Paper.pdf>.
- [2] R. Maconie, *Stockhausen on Music*. Marion Boyars, 1989.
- [3] (2011). [Online]. Available: <http://www.scambi.mdx.ac.uk/>
- [4] P. Decroupet, "Vers une théorie generale," *MusikTexte*, vol. 98, pp. 31–43, 2003.
- [5] J. Dack, "Notes on potential realizations of scambi," 2004, <http://www.scambi.mdx.ac.uk/documents.html>.
- [6] H. Pousseur, "Scambi," *Gravesaner Blätter*, vol. IV, pp. 36–54, 1959.
- [7] M. Duignan, J. Noble, P. Barr, and R. Biddle, "Metaphors for electronic music production in reason and live," in *Computer Human Interaction*, ser. Lecture Notes in Computer Science, M. Masoodian, S. Jones, and B. Rogers, Eds. Springer Berlin // Heidelberg, 2004, vol. 3101, pp. 111–120.
- [8] J. Oh, J. Herrera, N. J. Bryan, L. Dahl, and G. Wang, "Evolving the mobile phone orchestra," in *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*, Sydney, Australia, 2010.
- [9] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina, "The reactable," in *Proceedings of the International Computer Music Conference*, 2005.
- [10] H. Newton-Dunn, H. Nakano, and J. Gibson, "Block jam: a tangible interface for interactive music," in *NIME '03: Proceedings of the 2003 conference on New interfaces for musical expression*. Singapore, Singapore: National University of Singapore, 2003, pp. 170–177.
- [11] J. Patten, B. Recht, and H. Ishii, "Audiopad: a tag-based interface for musical performance," in *NIME '02: Proceedings of the 2002 conference on New interfaces for musical expression*. Singapore, Singapore: National University of Singapore, 2002, pp. 1–6.
- [12] P. L. Davidson and J. Y. Han, "Synthesis and control on large scale multi-touch sensing displays," in *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06)*, Paris, France, 2005.
- [13] H. Benko, A. D. Wilson, and P. Baudisch, "Precise selection techniques for multi-touch screens," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. Montréal, Québec, Canada, 2006, pp. 1263 – 1272.
- [14] K. Ryall, C. Forlines, C. Shen, M. R. Morris, and K. Everitt, "Experiences with and observations of direct-touch tabletops," in *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 89–96.
- [15] R. Fencott, "Interactive music using multi-touch cellular automata," in *(re)Actor3 the third international conference on digital live art*, Liverpool, September 2008.
- [16] T. Iwai, "Composition on the table," in *ACM SIGGRAPH 99 Electronic art and animation catalog*, ser. SIGGRAPH '99. New York, NY, USA: ACM, 1999, pp. 10–.
- [17] B. Pennycook, "Who will turn the knobs when i die?" *Organised Sound*, vol. 13, no. 3, pp. 199–208, 2008.
- [18] M. M. Johannes Goebel and P. Wood. Ideama. [Online]. Available: <http://on1.zkm.de/zkm/e/institute/mediathek/ideama/>
- [19] (2011). [Online]. Available: <http://polaris.gseis.ucla.edu/blanchette/MUSTICA.html>
- [20] D. Arfib and L. Kessous, "From "music v" to "creative gestures in computer music," in *VII Simpósio Brasileiro de Computação Musical*, 2000.
- [21] (2011). [Online]. Available: <http://www.robinfencott.com/ScambiSurface.php>
- [22] M. Kaltenbrunner and R. Bencina, "reactivision: a computer-vision framework for table-based tangible interaction," in *Proceedings of the 1st international conference on Tangible and embedded interaction*, ser. TEI '07. New York, NY, USA: ACM, 2007, pp. 69–74.
- [23] J. G. Sheridan, J. Tompkin, A. Maciel, and G. Roussos, "Diy design process for interactive surfaces," in *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, ser. BCS-HCI '09. Swinton, UK, UK: British Computer Society, 2009, pp. 485–493.
- [24] J. Y. Han, "Low-cost multi-touch sensing through frustrated total internal reflection," in *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ser. UIST '05. New York, NY, USA: ACM, 2005, pp. 115–118.
- [25] (2011). [Online]. Available: openframeworks.cc/
- [26] M. Dörfelt. Open frameworks contributed addons. [Online]. Available: <http://www.openframeworks.cc/addons/contributed>

SPATIO-TEMPORAL UNFOLDING OF SOUND SEQUENCES

Davide Rocchesso

IUAV - University of Venice
roc@iuav.it

Stefano Delle Monache

IUAV - University of Venice
stefano.dellemonache@gmail.com

ABSTRACT

Distributing short sequences of sounds in space as well as in time is important for many applications, including the signaling of hot spots. In a first experiment, we show that the accuracy in the localization of one such spot is not improved by the apparent motion induced by spatial sequencing. In a second experiment, we show that increasing the number of emission points does improve the smoothness of spatio-temporal trajectories, even for those rapidly-repeating pulses that may induce an auditory-saltation illusion. Other indications for auditory-display designers can also be drawn from the experiments.

1. INTRODUCTION

Everyday environments are populated of organisms and artefacts that constantly signal their presence and their state. Often, sound is exploited as the preferred channel to display the presence and the location of objects. For example, the Sonic Keyfinder¹ is a small key-ring that can be attached to any object, like bags, canes, remote controls, and reacts with bleeps to whistle or any loud noise, like shouts. Ordinary objects become sonically augmented and may exploit the human ability to locate a sound source in the physical space to communicate their location, and call attention. This scenario becomes intriguing for those objects that are provided with embedded computational affordances. For example, there are companies producing systems capable of charging mobile devices by proximity, without using any plugs. These systems rely on a charging hotspot that is usually embedded into furniture or dashboards and signaled by visual cues. In many circumstances, for aesthetic reasons or to avoid visual distraction, it would be preferable to use non-visual cues to signal a hotspot. However, auditory spatial resolution is poor [1] and, therefore, some degree of exploration is necessary. The apparent motion of a sound source may affect the instantaneous localization of sound events [2]. In this work, we are interested in measuring the quantity and quality of such apparent-motion effects in ecological conditions.

¹ <http://www.youtube.com/watch?v=7FJVNOW7aOo&NR=1&feature=fvwp>

Copyright: ©2011 Davide Rocchesso et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The paper has the following structure: In Section 2 the literature on non-visual mis-localization of stimuli in space is reviewed. In Section 3 two experiments, the first on localization accuracy, and the second on spatio-temporal sonic gestures, are described and discussed, in terms of implications for design. In Section 4 we draw our conclusion.

2. THE TACTILE RABBIT AND AUDITORY SALTATION

There are some non-visual illusions that show how humans consistently mis-localize stimuli in space when these are presented under certain temporal constraints. In particular, the cutaneous rabbit effect occurs when stimulating the skin at different points in a temporal sequence, if the temporal interval between two stimuli is small and their actual displacement is large. In such case the perceptual system consistently underestimates inter-stimulus distance and over-estimates inter-stimulus time. This illusion is correctly predicted by a Bayesian model that incorporates prior expectation for speed [3]. These effects have been recently exploited in product design, to actuate a jacket with vibration motors that are sparsely located on a large area of the body [4]. Thanks to the “rabbit”, a small number of “actuators can create the sensation that the arm is being tapped in several spots between the motors” [5]. It has also been shown how the duration of vibration bursts and the inter-onset-interval affect the experienced continuity and pleasantness of tactile stimuli [6].

In the auditory domain, an illusion similar to the cutaneous rabbit was reported in the seventies and called the auditory saltation [7]. A sequence of clicks was emitted by means of three loudspeakers only. However, for a certain range of inter-stimulus intervals, the subjects consistently reported sound events occurring between the actual emission points, with a phenomenal experience described as “a stick being run along a picket fence”. This particular even distribution of apparent locations was reported for very short inter-stimulus intervals (less than 50 ms), but a spread of apparent locations was reported up to about 200 ms of inter-stimulus interval. Experiments that measured the strength of auditory saltation with presentation of clicks via headphones were performed twenty years later [8]. For monoaural stimuli the effect never occur, and localization is discrete. For localization to be continuous in space, the stimuli must be dichotic with Interaural Time Difference (ITD) in a certain range (less than 1 ms) and inter-stimulus interval shorter than 100 ms. This kind of stimuli are perceived similarly as a variable ITD click train, representing a source that is actually hopping between the two ex-

treme positions. Further experiments with dichotic clicks measured the strength of the saltation effect under different degrees of lateralization [9]. A temporal window for stationarity was also measured to be about 350 ms long, thus meaning that if the initial stationary clicks (before the actual change in ITD occurs) stay within this window, the whole progression through space is reported to begin immediately. A procedure for the psychophysical assessment of individual auditory saltation, useful for the diagnosis of dyslexia, was also proposed [10]. With the method of constant stimuli, subjects had to discriminate between “actual” motion and saltation, with sequences played via headphones. The mean saltation threshold was found to be around 100 ms. In another experiment, subjects had to adjust eight sliders to report on the apparent position of individual clicks. It was found that some individual responses can be non monotonic. The reduced rabbit paradigm (three clicks) was used to check the effect of spectral content on saltation [11]. When the second click has a different content from the third click the effect is much weakened. It is argued that a form of perceptual masking occurs, where the localization of the target is impaired by the subsequent click. Displacements associated with saltation are stronger when the temporal, spatial, and spectral proximity of the stimuli is higher.

3. EXPERIMENTS ON SPATIO-TEMPORAL SONIC GESTURES

When sound is associated with movement we can talk about sonic gestures, with or without human agency [12]. Schemata of action-sound types can be summarized in: (i) *iterative*, when quick successions of small movements, and therefore corresponding sounds, are fused in a single gesture, or sound event, such as a drum roll; (ii) *impulsive*, namely gestures that imply discontinuous effort and are aimed at discrete events, such as hitting or knocking; (iii) *sustained*, when the action type requires a prolonged and continuous effort, such as bowing a string [13]. In music, a gesture is a coherent unit that develops in time, a trajectory in a space where a musical parameter (typically pitch) unfolds. Music theory, and especially music rhetoric, is in a large extent about how to design, concatenate, and overlap gestures [14]. Only occasionally the musical gestures inhabit a physical space, when there is an explicit displacement of a sound source, or when a gesture spans a spatial arrangement of sound sources (as in an orchestra). As long as energetic coherence unfolding in time and space is preserved, the action-sound types may be applied to new sounds, and be physically distributed in space to afford some gestural configurations.

3.1 Experiment 1: Is localization accuracy gesture-dependent?

The first experiment is aimed at studying if accuracy on spatial localization of a sound may be affected by sound motion, namely if arranging a point-like sound in a sequence of pulses distributed in space and time leads to a localization improvement.

In our experimental environment four piezo speakers are taped on a line along the middle line of a cardboard panel (1000 × 700mm), and arranged at equal distance from each other. The panel is hanging on a wall, with the speakers hidden from view, and the long side parallel to the floor. A projector beams a horizontal strip on the middle line of the panel. The strip is the clicking area for the user who will be using a mouse to manually input the estimated location of the sound event.

Rapid sequences of one to four impact sounds are played in various positions and direction, each impact assigned to a speaker. Various basic gestures are performed, in the classes: (i) point-like, (ii) linear monotonic sequence, (iii) linear sequence with one inversion of direction (at second or third impact). Subjects are asked to point to the position of the last impact in the sequence. Dispersion of answers gives an indication of accuracy in localization. The hypothesis is that accuracy increases with apparent sound motion, and with expectation on the final point in the sequence. We expect that accuracy on static localization integrates with other information coming from motion (essentially temporal information and expectation), thus increasing the final accuracy. The psychoacoustic literature has previously faced the problem of measuring accuracy in localization tasks [15].

3.1.1 Setup

Let δ be the distance between two contiguous piezo speakers, and d be the distance between the panel and the listener. If the listener has the head facing the panel and symmetrically located in the middle of the two piezo speakers, the angle between the listener and such two speakers is

$$\alpha = 2 \arctan \frac{\delta}{2d}. \quad (1)$$

If $d = 1700\text{mm}$ and $\delta = 300\text{mm}$, we get $\alpha = 10.085^\circ$. This is well beyond the minimum audible angle, which amounts to a couple of degrees for frontal sources [1]. Let T be the inter-onset interval between the sounds being emitted by two adjacent speakers. To be sure that no offset displacement due to auditory saltation occurs, T should be larger than about 300ms [9]. Although we are interested in the final position, a displacement in the intermediate positions may affect the regularity of the pattern and, therefore, the expected final position.

If a continuous sound source would move continuously between two adjacent points, the velocity limen would be $9.1^\circ/\text{s}$ for a source moving at $30^\circ/\text{s}$ [16]. Given the above constraint on the time to jump from a location to the adjacent one, we would have velocities lower than this, thus ensuring that α is larger than the minimum audible movement angle.

The subject seats in front of the cardboard panel, approximately at a distance of 1.7m. The test is run in ecological conditions, that is in an ordinary, everyday life environment, and in our case in a small room of $3 \times 5\text{m}$ with a wooden floor and a glass door in a glass wall, equipped with regular office furniture (a large bookshelf, a desk and chairs). The background noise includes the fan of the video

projector, the hum of the heating system, various noises coming from the corridor, church bells in the distance from time to time. The measured average background noise under experimental conditions was 40.7dbA (rms value with frequency weighting A and fast exponential averaging of 125ms).

3.1.2 Stimuli

The sound stimuli are synthetic impact sounds, generated with the Sound Design Toolkit [17], and designed to convey the impression of short impacts on a wooden surface. Figure 1 shows the four pulses, each emitted by a piezo speaker, as captured at the position of the listener’s head, with a Zoom H2 Handy Recorder with X/Y internal microphones configuration at 120°. To have a clearer view, the waveforms of the impact sound have been juxtaposed with an inter-onset interval much larger than the 300ms used in the experiment. The two channels have been displaced 200ms for a better comparison. The synthetic sound for the experiment has been prepared by interactive listening through the actual setup, with the whole transmission chain, from real-time synthesis to pressure waves at the ear. It has to be noted that differences in the piezo speakers and in their mechanical coupling with the board give rise to different waveforms at the listening position. As seen from figure 1, the stimuli peaked about 20dB higher than background rms noise level.

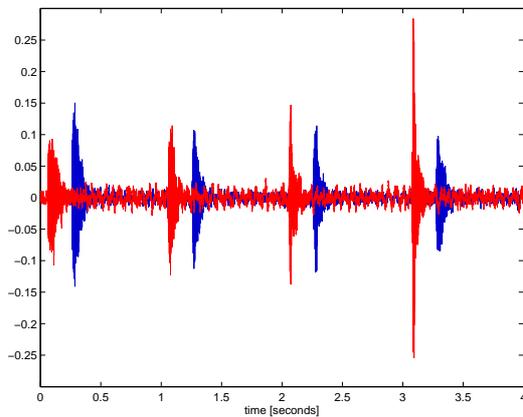


Figure 1. Waveforms of the synthetic, wooden impact sound stimulus as coming from the four emission points and reaching the listener’s head. Left (blue colour) and right (red colour) channels have been chopped and displaced for better visibility.

3.1.3 Procedure

The trial is automated and formed of 5 cycles of 26 randomly played sound stimuli, for a total amount of 130 stimuli. Each stimulus is either a single impact sound or a sequence of two to four impacts emitted from different piezo loudspeakers. The list of stimuli is described in table 1. The subject is asked to locate the last heard sound along a red line projected on the panel. At the end of each stimulus the colour of the line switches to green, and the

| n. | sequence of actuated speakers | | | |
|----|-------------------------------|---|---|---|
| 1 | 1 | | | |
| 2 | 2 | | | |
| 3 | 3 | | | |
| 4 | 4 | | | |
| 5 | 1 | 2 | | |
| 6 | 2 | 3 | | |
| 7 | 3 | 4 | | |
| 8 | 1 | 2 | 3 | |
| 9 | 2 | 3 | 4 | |
| 10 | 1 | 2 | 3 | 4 |
| 11 | 4 | 3 | | |
| 12 | 3 | 2 | | |
| 13 | 2 | 1 | | |
| 14 | 4 | 3 | 2 | |
| 15 | 3 | 2 | 1 | |
| 16 | 4 | 3 | 2 | 1 |
| 17 | 1 | 2 | 1 | |
| 18 | 2 | 3 | 2 | |
| 19 | 3 | 4 | 3 | |
| 20 | 4 | 3 | 4 | |
| 21 | 3 | 2 | 3 | |
| 22 | 2 | 1 | 2 | |
| 23 | 1 | 2 | 3 | 2 |
| 24 | 2 | 3 | 4 | 3 |
| 25 | 4 | 3 | 2 | 3 |
| 26 | 3 | 2 | 1 | 2 |

Table 1. The 26 stimuli used in the experiment

subject can indicate the final landing point, by pointing and clicking with a mouse. Between the subject selection and the following stimulus there is a pause of 3s. Collected responses are saved in a text file. For each stimulus the text file includes: the corresponding sequence, the inter onset interval, the final location along the line in a range between 0 and 1, the subject’s response time in ms.

The subjects are briefly informed on the objective of the experiment, namely measuring the accuracy in the spatial localization of moving sound sources, and asked to give their informed consent. Afterwards the task is explained through a metaphor: the subject will hear a pet pattering behind the red strip. When the pet stops, the color of the strip turns green and the position of the pet on the strip has to be located by pointing and clicking with the mouse. After the experiment, each subject is debriefed and his or her comments recorded for further analysis.

3.1.4 Results

Eleven subjects, seven males and four females, ranged in age between 27 and 43, performed the experiment. Only subject n. 5 reported a partial hearing loss at one ear. A posteriori we verified that he could easily discriminate between left and right stimuli, and we decided to keep him in the pool of subjects. The boxplots of figure 2 show how the individual responses to the five cycles of the single stimulus are distributed. It is clear that some subjects (5, 7) preferred to collapse their responses toward the extremes, while the others used most of the available space.

To test the hypothesis that the sequencing of impact sounds affects the localization accuracy for the last impact, the results from all 11 subjects and all 26 stimuli have been aggregated in the boxplots of figure 3. The nine boxplots correspond to the categories of impact sound sequences listed in table 2.

The visual inspection of the boxplots induces some observations: (i) emission points are quite well localized; (ii)

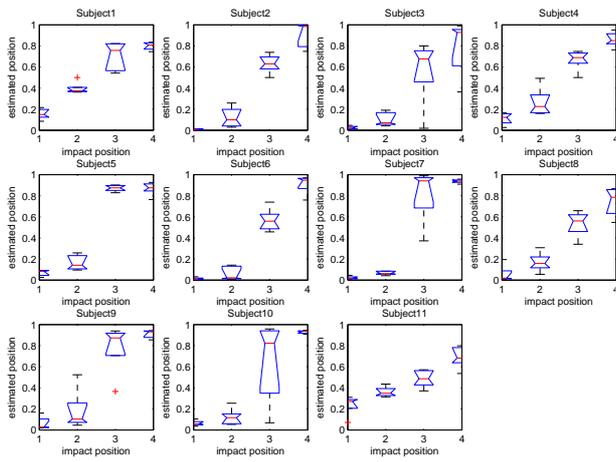


Figure 2. Boxplots of individual responses sorted by emission point (1 to 4)

| sequence description | sequence n. (see table 1) |
|---------------------------------------|---------------------------|
| single impact | 1, 2, 3, 4 |
| double impact - left to right | 5, 6, 7 |
| double impact - right to left | 11, 12, 13 |
| triple impact - LR (3,4) and RL (1,2) | 8, 9, 14, 15 |
| quadruple impact - LR (4) and RL (1) | 10, 16 |
| triple impact back and forth - LRL | 17, 18, 19 |
| triple impact back and forth - RLR | 20, 21, 22 |
| quadruple impact back and forth - LLR | 23, 24 |
| quadruple impact back and forth - RRL | 25, 26 |

Table 2. Categories of impact sequences

localization is more accurate near the board rim; (iii) accuracy is not affected by the sequence. Overall, the subjects localize the final impact in each sequence around the actual emission point with a standard deviation that is always less than 20% of the whole strip length, which is significantly larger than the minimum audible angle.

From visual inspection of the single-impact boxplot of figure 3 it seems that standard deviation is smaller at the rim. That there is a significant variation in variance of localization among the different emission points is confirmed by a Levene’s test on equality of variances ($F(3, 216) = 8.4994, p = 2.3E-5$). Standard deviation at positions 1 and 4 is around 10% of the whole strip length. That localization is more accurate at the rim is confirmed by the results of localization of arrival points for the other sequences. In order to eliminate the variability among repeated measures of the same subject in the same condition, we took the median of each set of five cycles and checked if such median estimate would change its variance with the different categories described in table 2. A Levene’s test did not allow to confute the null hypothesis of equality of variances. For example, for final emitting point at position 1 (five possible sequences in table 1): $F(4, 50) = 0.484$, and $p = 0.747$. Similarly, for final emitting point at position 2 (seven possible sequences in table 1): $F(6, 70) = 0.709$, and $p = 0.6444$.

3.1.5 Discussion

It seems that the hypothesis of better accuracy for sequences of three or more impacts, as induced by some expectation

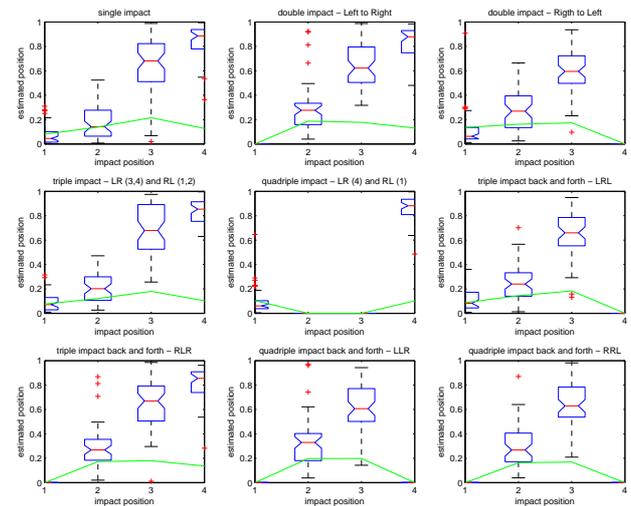


Figure 3. Boxplots of all collected responses sorted by emission point (1 to 4). Each boxplot represents a subset of stimulation sequences (see table 2). The green line connects the values of standard deviation, for each emission point.

on the space-time localization of the final impact, is not confirmed by the experiment. In fact, auditory saltation effects typically occur for intermediate emissions in a sequence. Stretching and compression of time and space do not occur for the extreme stimulation points, also in the cutaneous rabbit illusion. However, the experiment shows that the points close to the boundary are located quite accurately.

In the comments, all the subjects pointed out a major perceived immediacy in locating point-like, stationary sounds. The majority exploited the sounds occurring on the rims of the panel as anchors and reference points to construct the reach of the designed auditory space. In addition, several subjects showed a preference to an eyes-free interaction approach in executing the task, due to a relative uselessness of sight (subject 1), a clearer identification of the landing point at the extreme left, right or the center of the strip (subject 6), and a certain misleading and interfering effect of the cursor on screen with the localization task (subject 9).

Indeed, the primary aim of audition with respect to space is to orient the gaze toward the source [18, 19]. In this respect, using a sequence that traverses the available surface space, as in stimuli 10 and 16 of table 1 gives some advantages, as compared to a in-place emission. First, the early impacts of the sequence work as *attensons*, to drive the users’ attention toward the emission point, since “they emphasize the sound motion” (subject 1). In our experimental conditions the subjects were attending the audio-visual display, but in ecological and practical settings, the attention of potential users will need to be driven toward the interaction point. Second, sequencing acoustic emissions in time and space opens a wide design space: While point-like stimuli were reported simply as “sound”, sound

motion in sequences was described in terms of pairs of opposites near/far, back and forth, and through metaphors, therefore stressing a major, inner expressiveness. For instance, subject 6 visualized the perceived jumps of the virtual pet, while subject 9 figured the task as a sort of Duck Hunt game². Ultimately, subject 7 imagined something similar to the conjuring trick of the cups and the balls.

Several subjects reported to be misled by pitch during the experiment. Although the stimulus is a synthetic sound, perceived timbral differences at the various emission points are due to the non-linear characteristics of the four piezo-speakers, and to the different excitation of the normal modes of the board. Several subjects reported the metaphor of a piano keyboard as early strategy in executing the task: “Following the sounds as if they were moving on a musical scale”, “with lower pitches on my left and higher ones on my right” (subjects 2, 3). In fact, pitch height (frequency) is known to have an associative spatial stereotype effect, with the apparent movement of a sound source in the orthogonal plane. Higher-frequency pitches tend to be associated to right/up locations, while lower-frequency pitches to left/down locations [20].

In summary, spatio-temporally distributed pulses do not affect significantly the accuracy in the localization of the final landing point. Nonetheless, the experiment shows some implications for the design of auditory interfaces: Even simple arrangements of point-like sounds in basic, linear, monotonic sequences allow to construct expressive gestures and give rise to meaningful, interpretive processes.

3.2 Experiment 2: Auditory saltation and perceived gesture

In the second experiment we explored the gestural dimension of spatio-temporally distributed pulses. In particular, we investigated how short sequences of point-like sounds that originate on the rim and traverse the available surface space, are perceived and represented in terms of trajectories and gestures. Indeed, short sequences of pulses displaced in space and rapidly repeated in time (inter-onset-interval – IOI below 100ms) are perceived as continual. The hypothesis is that actual perception of the time-space distribution of events may be affected by illusory saltation (see section 2).

It is interesting to see how subjects represent the perceived gesture, and how this relates to the physical spatio-temporal distribution of pulses. For this purpose, participants were asked to reproduce, by tapping or tracing on a graphic tablet, the sequence of pulses. The precise timing, position and displacement of the pen tip on the tablet was acquired.

3.2.1 Setup and stimuli

This second experiment was run with the same setup and impact sounds, as in the first experiment, except that no visuals are projected on the panel. As an input device, a Wacom Intuous 2 USB tablet is used. The trial consists of 3 groups of 8 sound stimuli, for a total of 24 randomly played stimuli. Each stimulus is a sequence of twelve impact

| n. | sequence of actuated speakers | | | | | | IOI (ms) |
|----|-------------------------------|--------|-----|--------|----|----|----------|
| 1 | 111111 | | | 444444 | | | 300 ms |
| 2 | 444444 | | | 111111 | | | ” |
| 3 | 111 | 222 | 333 | 444 | | | ” |
| 4 | 444 | 333 | 222 | 111 | | | ” |
| 5 | 111111 | 222222 | | | | | ” |
| 6 | 444444 | 333333 | | | | | ” |
| 7 | 11 | 22 | 33 | 44 | 33 | 22 | ” |
| 8 | 44 | 33 | 22 | 11 | 22 | 33 | ” |
| 9 | 111111 | | | 444444 | | | 150 ms |
| 10 | 444444 | | | 111111 | | | ” |
| 11 | 111 | 222 | 333 | 444 | | | ” |
| 12 | 444 | 333 | 222 | 111 | | | ” |
| 13 | 111111 | 222222 | | | | | ” |
| 14 | 444444 | 333333 | | | | | ” |
| 15 | 11 | 22 | 33 | 44 | 33 | 22 | ” |
| 16 | 44 | 33 | 22 | 11 | 22 | 33 | ” |
| 17 | 111111 | | | 444444 | | | 75 ms |
| 18 | 444444 | | | 111111 | | | ” |
| 19 | 111 | 222 | 333 | 444 | | | ” |
| 20 | 444 | 333 | 222 | 111 | | | ” |
| 21 | 111111 | 222222 | | | | | ” |
| 22 | 444444 | 333333 | | | | | ” |
| 23 | 11 | 22 | 33 | 44 | 33 | 22 | ” |
| 24 | 44 | 33 | 22 | 11 | 22 | 33 | ” |

Table 3. The 24 stimuli used in the experiment. Numbers 1 to 4 represent the active speaker, each number repeated according to the number of pulses per actuated speaker. Reading a line left to right gives the spatial and temporal unfolding of the sequence.

sounds evenly distributed in time and spatially arranged as (i) a traversing sequence from one side to the other, (ii) or a traversing sequence with one inversion of direction (at the anchor point), (iii) or a sequence presenting two blocks of six impacts at the opposite rims, (iv) or a sequence of two blocks of six impacts at adjacent positions on the left or right half of the panel. The first group of sequences is played with an IOI of 300ms, the second group with an IOI of 150ms, and the third group with an IOI of 75ms. The complete list of stimuli is given in table 3.

The Wacom tablet is used as scaled analogue of the cardboard panel, where the subjects can represent the stimuli. The collected data for the subject’s response to each stimulus includes an indexical flag of the randomly played sequence, the corresponding IOI, the points marked with the relative temporal distance in ms between each pair, the XY coordinates of the pen strokes per instant of time.

The subject is instructed about the objective of the experiment, namely observing if and how sequences of short pulses, spatio-temporally distributed along the horizontal axis located in the middle part of a surface (the cardboard panel) may be perceived as gestural strokes. Hence, the subject is asked to reproduce on the tablet the last heard sequence, with freedom to use both pointillistic and/or continuous strokes, according to his/her ease and confidence. The subjects are explicitly acquainted of the unidimensional nature of the experiment that takes in account only the perceived movement of the sound sequences in the horizontal plane, and does not consider the perceived displacement in the vertical plane. A short training session is dedicated to listening to a couple of sequences per group of stimuli, in order to raise any misunderstanding about the task. Two sequences with long IOI, one traversing the panel and one presenting the blocks of impacts at the opposite rims, are always played as initial training elements,

² http://en.wikipedia.org/wiki/Duck_Hunt.

in order to highlight the difference between the stimuli that are coming from left and right extremities of the panel and the stimuli that traverse the board. The long IOI acts a control condition, being the auditory saltation effect typically occurring for shorter IOI. The sequences are manually triggered by the experimenter, after the experiment each subject is debriefed and his or her comments recorded for further analysis.

3.2.2 Precision of the input device

Graphic tablets have been extensively used as input device in sound and music computing [21]. Some measurement of total latency have been performed [22] by capturing the contact sound of the pen on the tablet with a microphone and by measuring the time lag between the detected sound impulse and the contact information received via the tablet. We did a similar measurement using the Alesis soundcard iO26 firewire, a MacBook Pro 2.33 GHz Intel Core Duo running Max/MSP 5 on Mac OS X 10.6.6. The external wacom by Jean-Michel Couturier³ was used to capture the tablet data. The two stimuli (audio and tablet data) were displaced in time of a few (positive or negative) milliseconds. On a sequence of 128 stimuli, the measured mean temporal displacement was -1.44ms , with a standard deviation of 13.77ms . The time interval between detected XY events for continuous strokes is on average between 8ms and 12ms .

3.2.3 Results

Twelve volunteer subjects, four males and eight females in an age between 29 and 70, participated to the experiment.

For each subject and each stimulus, the X coordinates of the Wacom events were plotted versus time and the plots for all stimuli were arranged in a table, as reported in figure 4 for one subject. For ease of comparison, time was normalized to overall gesture duration (represented by number 1000). In addition, we measured the density of pen events sent by the tablet in dots/ms. It is clear from the zero overlap between boxes in figure 5, that the density for 300ms of IOI (1 in the figure) is significantly smaller than the density for 75ms of IOI (3 in the figure). The overall duration of gestures has also been measured and is represented in the boxplot of figure 6. It shows that gestures corresponding to 300ms of IOI take significantly longer than gestures corresponding to 75ms of IOI.

3.2.4 Discussion

On the single subject of figure 4, several observations may be made: (i) plots in the leftmost column are more step-like than plots in the rightmost column; (ii) plots in the leftmost column are more dot-like, while plots in the rightmost column are more continuous; (iii) local inconsistencies are found, as for example in the central plot of the bottom row. The first observation seems to support the emergence of a saltation effect. To see how this generalizes across subjects, the gestures of all subjects were made continuous by resampling with a zero-order hold, and the mean gestures plus/minus standard variance were plotted (see figure 7).

³ <http://www.jmc.blueyeti.fr/download.html>

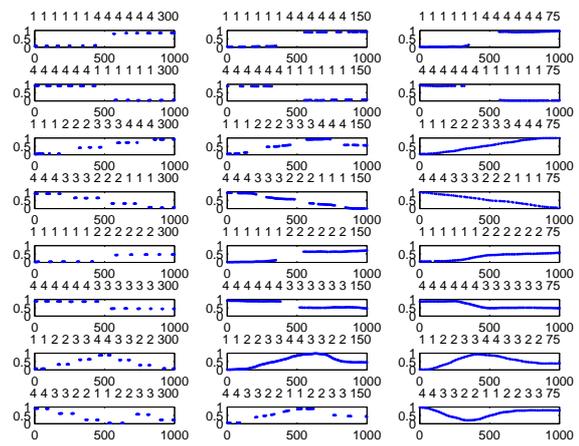


Figure 4. Reproduced gestures for one subject. The three columns, left to right, correspond to IOI of 300ms, 150ms, and 75ms, respectively. In each plot, vertical axis is the normalized horizontal position in the tablet, and horizontal axis is time, normalized to the whole gesture duration.

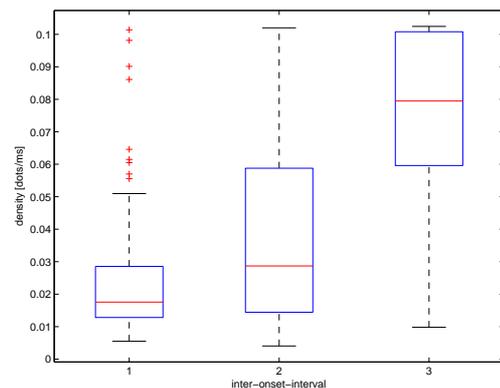


Figure 5. Boxplots of density of pen events for all subjects and all sequences, for the three different IOIs.

Here, the difference between the leftmost and rightmost plots is much harder to appreciate. On the contrary, it is clear that the sequences that traverse the board by using all the four speakers (second and third row) are smoother than the sequences that use only the extremal speakers. This shows that the internal speakers are not useless in defining the movement and that illusory saltation between the extremal position is not very relevant. We should be careful before saying that there is no or little saltation when going from 300ms to 75ms, just because the leftmost and rightmost plots in rows 3 and 4 of figure 7 look similar. Indeed, the steps that are clearly visible in the corresponding leftmost plots of figure 4 may disappear from the mean trajectory just because different subjects locate them at different positions in time.

The second observation made for the single subject is supported by the fact that density of pen events for 75ms of IOI (3 in figure 5) is significantly larger than density

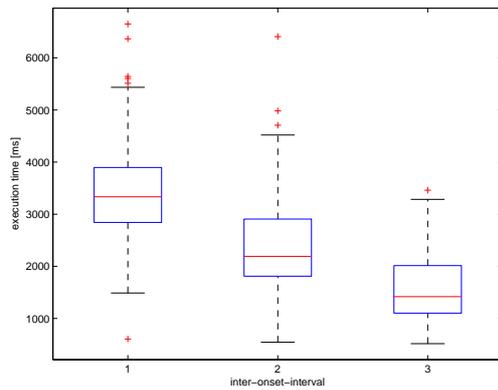


Figure 6. Boxplots of duration of reproduced gestures for all subjects and all sequences, for the three different IOIs.

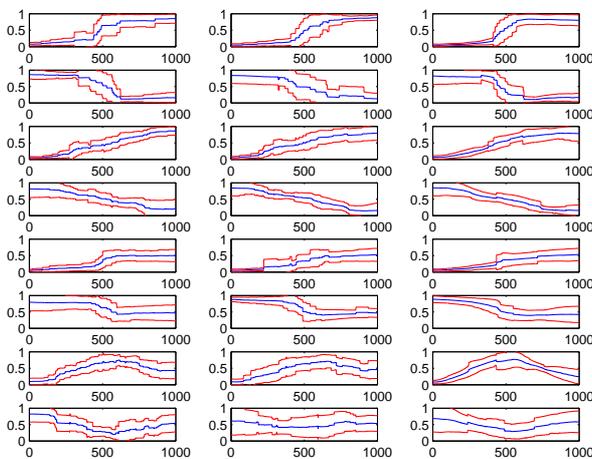


Figure 7. Mean and standard deviation of the collected responses of all the subjects. Duration of responses has been normalized to 1000ms. The three columns are sorted by IOI (300ms, 150ms, and 75ms), from left to right.

for 300ms of IOI (1 in the figure), thus meaning that gestures were definitely denser, or more continuous, in the latter case. The perceived major smoothness and continuity of sequences with shorter IOI (75ms) is corroborated by participants' comments. This fact is only partially emerging from figure 7. Finally, with respect to gesture duration (figure 6), it should be noted that gestures corresponding to stimuli with 75ms of IOI take one quarter of the time taken by the stimuli using a 300ms. Subjects were left free to use their own time scale in the reproducing gesture and, indeed, they did not scale with the duration of stimuli, as the ratio between median execution times in the first and third columns of figure 6 is about 2.35. Conversely, the medians of densities reported in figure 5 scale almost perfectly with the density of stimuli.

Several subjects reported to perceive the sequences with shorter IOI and presenting pulses at the extremities (n. 17 and 18 in table 3), as two *blocks* of events very close to

each other and almost tied. Nonetheless only three subjects represented them with a tied and continuous stroke, which highlights the limited incidence of auditory saltation effect in ecological conditions. Subjects 1 and 9 stressed a sort of wake effect, subject 7 recalled the sound of rolling on a snare, while subject 9 reported the metaphor of two separate blocks of *domino* pieces falling. In traversing sequences, the *domino* effect sensation was in fact complete, running from one edge to the other, the movement more fluent and pleasurable (subject 7). Traversing sequences, in particular those with shorter IOI, give rise to cognitive representations emanating from the sound characteristics [23], and the spatio-temporal distribution. The depicted gestures represent a sort of signature, or perceived morphology of the sound contours in space and time. An additional consideration concerns how timing of pulses, and total duration of sequences affect the adopted strategy. In the comments, the control condition sequences (IOI 300ms) are described with terms, often inappropriate or naïve, like *rhythm*, *beats*, *jumps*, *hits*, therefore implying blocks of discrete events. This group of sequences acted as reference point for the execution of the task, even when they were randomly presented later in the trial. It is a natural and immediate strategy to try to count the pulses and tap them accordingly, when possible. On the contrary, the comparison against the other two groups of sequences is done in terms of a scale of speed. Pulses with short IOI (75ms) are almost perceived as tied and induce a change of strategy, showing the difficulty of tapping at the same tempo in the attempt of reproducing the sequence. Pulses with intermediate IOI (150ms) seem to require some preparatory gestures in order to reproduce the taps. It was observed that several subjects that preferred to tap these sequences prepared them by tapping in the air in order to keep the tempo. These sequences reveal the subjects' attitude toward a pointillistic or stroke-like approach to the task.

4. CONCLUSION

Sound is increasingly being used as intentional design element in artefacts and environments, as specifier of brands and privileged channel of interaction. Space is an obviously ineluctable dimension of the experience of the world and the moderate accuracy of the human ear in sound localization is a matter of fact. Research in sound design has to look closely at the element of space. For instance, the effectiveness of a well designed sound logo may be reduced if badly presented at the touch points between companies and their customers. For this purpose, we investigated the quantity and quality of apparent sound motion effects, such as auditory saltation, in ecological conditions. In the first experiment we found that arranging a point-like sound in spatio-temporally distributed sequences does not improve noticeably the localization of the final, landing point. Yet, design can exploit the emerging anchor effects of the extremal elements to display interaction reaches, while taking advantage of the initial elements to call attention toward the emission point. The second experiment showed that auditory saltation effect in ecological conditions is reduced compared to headphones listening. Nonetheless we

could observe how it affects the perceived representations of the sound motion, in terms of gestural strokes. This opens a wide design space, since acoustic brand units, in the form of sound logos, jingles, display or product sound, can be developed in space and time, thus introducing a shape aspect that is normally not explicit.

5. REFERENCES

- [1] A. W. Mills, "On the minimum audible angle," *The Journal of the Acoustical Society of America*, vol. 30, no. 4, pp. 237–246, 1958.
- [2] J. Neuhoff, "Auditory motion and localization," in *Ecological psychoacoustics*, J. Neuhoff, Ed. New York: Academic Press, 2004, pp. 87–111.
- [3] D. Goldreich, "A Bayesian Perceptual Model Replicates the Cutaneous Rabbit and Other Tactile Spatiotemporal Illusions," *PLoS ONE*, vol. 2, no. 3, pp. e333+, March 2007.
- [4] P. Lemmens, F. Crompvoets, D. Brokken, J. van den Eerenbeemd, and G.-J. de Vries, "A body-conforming tactile jacket to enrich movie viewing," in *Proceedings of the World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 7–12.
- [5] W. D. Jones, "Jacket lets you feel the movies," *IEEE Spectrum*, 2009, <http://spectrum.ieee.org/biomedical/devices/jacket-lets-you-feel-the-movies>.
- [6] J. Raisamo, R. Raisamo, and V. Surakka, "Evaluating the effect of temporal parameters for vibrotactile saltatory patterns," in *Proceedings of the 2009 international conference on Multimodal interfaces*, ser. ICMI-MLMI '09. New York, NY, USA: ACM, 2009, pp. 319–326.
- [7] C. D. Bremer, J. B. Pittenger, R. Warren, and J. J. Jenkins, "An Illusion of Auditory Saltation Similar to the Cutaneous "Rabbit"," *The American Journal of Psychology*, vol. 90, no. 4, 1977.
- [8] D. I. Shore, S. E. Hall, and R. M. Klein, "Auditory saltation: A new measure for an old illusion," *The Journal of the Acoustical Society of America*, vol. 103, no. 6, pp. 3730–3733, 1998.
- [9] D. P. Phillips and S. E. Hall, "Spatial and temporal factors in auditory saltation," *The Journal of the Acoustical Society of America*, vol. 110, no. 3, pp. 1539–1547, 2001.
- [10] J. C. Kidd and J. H. Hogben, "Quantifying the auditory saltation illusion: An objective psychophysical methodology," *The Journal of the Acoustical Society of America*, vol. 116, no. 2, pp. 1116–1125, 2004.
- [11] S. Getzmann, "The Effect of Spectral Difference on Auditory Saltation," *Experimental Psychology*, vol. 55, no. 1, pp. 64–71, 2008.
- [12] A. R. Jensenius, M. M. Wanderley, R. I. Godøy, and M. Leman, "Musical gestures: concepts and methods in research," in *Musical Gestures - Sound, Movement, and Meaning*, R. I. Godøy and M. Leman, Eds. New York: Routledge, 2010, pp. 12–35.
- [13] R. I. Godøy, "Gestural affordances of musical sound," in *Musical Gestures - Sound, Movement, and Meaning*, R. I. Godøy and M. Leman, Eds. New York: Routledge, 2010, pp. 103–125.
- [14] A. Schneider, "Music and Gestures: A Historical Introduction and Survey of Earlier Research," in *Musical Gestures: sound, movement, and meaning*, R. I. Godøy and M. Leman, Eds. New York, NY, USA: Routledge, 2010, pp. 69–100.
- [15] S. Carlile, P. Leong, and S. Hyams, "The nature and distribution of errors in sound localization by human listeners," *Hearing Research*, vol. 114, no. 1-2, pp. 179 – 196, 1997.
- [16] S. Carlile and V. Best, "Discrimination of sound source velocity in human listeners," *The Journal of the Acoustical Society of America*, vol. 111, no. 2, pp. 1026–1035, 2002.
- [17] S. Delle Monache, P. Polotti, and D. Rocchesso, "A toolkit for explorations in sonic interaction design," in *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound*, ser. AM '10. New York, NY, USA: ACM, 2010, pp. 1:1–1:7.
- [18] D. V. Valkenburg and M. Kubovy, "In defense of the theory of indispensable attributes," *Cognition*, vol. 87, pp. 225–233, 2003.
- [19] M. Kubovy and M. Schutz, "Audio-visual objects," *Review of Philosophy and Psychology*, vol. 1, pp. 41–61, 2010.
- [20] E. Rusconi, B. Kwan, B. L. Giordano, C. Umilt, and B. Butterworth, "Spatial representation of pitch height: the SMARC effect," *Cognition*, vol. 99, no. 2, pp. 113 – 129, 2006.
- [21] M. Zbyszynski, M. Wright, A. Momeni, and D. Cullen, "Ten years of tablet musical interfaces at CNMAT," in *Proceedings of the 7th international conference on New interfaces for Musical Expression*. ACM, 2007, pp. 100–105.
- [22] M. Wright, R. J. R. Cassidy, and M. F. Zbyszynski, "Audio and gesture latency measurements on Linux and OSX," in *Proceedings of the International Computer Music Conference*, 2004, pp. 423–429.
- [23] B. Caramiaux, P. Susini, T. Bianco, F. Bevilacqua, O. Houix, N. Schnell, and N. Misdariis, "Gestural embodiment of environmental sounds: an experimental study," in *Accepted for publication in Proc. NIME 2011 - New Interface for Musical Expression*, Oslo, Norway, 2011.

AN EXPLORATION ON THE INFLUENCE OF VIBROTACTILE CUES DURING DIGITAL PIANO PLAYING

Federico Fontana

Marco Civolani

Università di Udine

Department of Mathematics
and Computer Science
via delle Scienze, 206
33100 Udine, Italy

federico.fontana@uniud.it

Stefano Papetti

Valentina del Bello

Università di Verona

Department of Computer Science
strada Le Grazie, 15
37134 Verona, Italy

stefano.papetti@univr.it

Balázs Bank

Budapest University of Technology
and Economics

Department of Measurement
and Information Systems
H-1521 Budapest, Hungary

bank@mit.bme.hu

ABSTRACT

An exploratory experiment was carried out in which subjects with different musical skills were asked to play a digital piano keyboard, first by following a specific key sequence and style of execution, and then performing freely. Judgments of perceived sound quality were recorded in three different settings, including standard use of the digital piano with its own internal loudspeakers, and conversely use of the same keyboard for controlling a physics-based piano sound synthesis model running on a laptop in real time. Through its audio card, the laptop drove a couple of external loudspeakers, and occasionally a couple of shakers screwed to the bottom of the keyboard. The experiment showed that subjects prefer the combination of sonic and vibrotactile feedback provided by the synthesis model when playing the key sequences, whereas they promote the quality of the original instrument when performing free. Even if springing out of a preliminary evaluation, these results were in good accordance with the development stage of the synthesis software at the time of the experiment. They suggest that vibrotactile feedback modifies, and potentially improves the performer's experience when playing on a digital piano keyboard.

1. INTRODUCTION

For its versatility and diffusion in diverse musical styles, with the advent of electro-mechanics, electronics, and finally digital technology, the piano has been progressively re-designed and engineered in different forms mainly to make its portability easier. Although sounding quite different, siblings such as the Clavinet and Rhodes electric pianos were initially able to keep a certain flavor of the original instrument, and then to conquer their own niche in contemporary music. In the meantime the early digital piano keyboards had begun to revolutionize the musical instrument market, by making piano performances possible

on small stages and at home, where it would be otherwise unpractical or impossible to set up the original instrument.

With the advance of technology, digital pianos have progressively increased in sound accuracy and fidelity of their keyboard's response. Current flagship products exhibit sounds and key mechanics that satisfy the performer completely, once taken into account the relatively minor cost, size and weight of the digital instrument compared to its mechanical counterpart. On the other hand, the issue of vibrotactile feedback has been still left largely unexplored, despite playing a fundamental role in the performance on a real piano.

In spite of the current psychological and applied research trend toward a more systematic inclusion of vibrotactile devices in musical interfaces [1, 2, 3], also thanks to the notable decrease in costs of the related actuation technologies, we found only few studies on the topic of vibrations in the piano. Among these studies [4] there are a computational solution for improving the vibrotactile feedback provided by the upright piano through adaptation of the keybed impedance toward the characteristic values of the grand piano [5], and signs of research activity advertised in the CCRMA web pages, on tactile feedback design applied to the keyboards based on previous research made by Chafe [6].

Performers traditionally experience vibrotactile feedback in digital pianos only as a by-product of the resident loudspeaker system. By transmitting vibrations across the instrument body during the sound reproduction, the loudspeakers are in fact responsible of providing some related cues to the player. Arguably, such cues cannot achieve the intensity nor resemble the quality of those originating from a real piano keyboard, when the soundboard resonates under the action of the strings.

At least one flagship product, the Yamaha AvantGrand digital piano (see www.avant-grand.com), augments sounds by means of an active vibration system. By enabling transducers located under the keyboard and behind the music stand, it promises to engage users in a full-body sensory experience during playing [7, 8]. Indeed, the multimodal perception of harmonic components across the whole human body has been recognized to increase the engagement and sense of presence in users [9].



Figure 1. Clavinova YDP-113.

In the case of the Yamaha product, all its design solutions concur to form an extremely faithful reproduction of the original instrument, including its visual appearance. Due to unavoidable contingencies, in our experiment we rather made use of a less sophisticated musical interface, in both visual and non visual sense. More precisely:

- we used a Yamaha Clavinova YDP-113, an inexpensive digital piano (see Figure 1) that provides internal loudspeakers, but also offers MIDI master keyboard functionalities;
- in alternative to the Clavinova sounds, we synthesized sonic and vibrotactile feedback in real time by means of a physics-based piano sound model running on a laptop.

Vision did not play an integrative role in the multimodal scene. In fact, due to the appearance of the Clavinova-based setup, subjects were constantly aware that they were *not* playing a real piano. Using this setup we were able to provide the subjects with different sounds, with and without vibrotactile feedback.

The results of our tests overall suggest that the subjective judgments on sound quality were influenced by the vibrotactile feedback. Furthermore, as shown in the following of this paper, an analysis of the individual judgments indicates that the tactile modality can improve the auditory perception of digital piano sounds. However promising, such results call for a more robust and systematic validation that is expected to become object of future research.

2. EXPERIMENT

The experimental hypothesis was that the vibrotactile feedback coming from the instrument keyboard had influence on the perceived quality of piano sounds.

2.1 Subjects

Nine subjects voluntarily participated in the experiment. Three of them were pianists, four of them were other instrument players, and two of them were non-musicians.



Figure 2. Monacor Carpower BR 25 shaker.

2.2 Setup and configurations

Instead of explicitly being involved in a vibrotactile evaluation task, subjects were asked to rate the sound quality associated to two different digital piano settings: in the former the Clavinova worked with its internal loudspeakers; in the latter, the Clavinova controlled a synthesis software running in real time on a Core 2 Duo Dell Latitude E6400 laptop, in its turn driving a pair of Genelec 2029BR external loudspeakers along with a pair of Monacor Carpower BR 25 shakers.

The shakers (one is shown in Figure 2) were screwed to the bottom of the Clavinova. They are able to transmit mechanical power to the body they are in contact with. As a side effect they also generate some sound, amounting to few dB of intensity level that adds to the loudspeaker emission.

In spite of a claimed active band in the range 30-300 Hz, we measured that the Monacor shakers in practice work up to a few kHz, hence covering sufficiently well the entire vibrotactile perceptual band in correspondence of the finger, in particular including its higher sensitivity region centered around 250 Hz [10].

As noted by Bank [11], the active range of the shaker is sufficient to excite all the components that can be perceived by the palm [12] during normal piano playing. Figure 3 shows examples of such thresholds in dashed lines for the notes C_2 , C_4 , C_6 , and the C chord $\sharp 2$.

The software running on the laptop implemented a recently developed physics-based model [13] for the synthesis of piano sounds. The model was configured to compute two sound signals in correspondence of the left and right part of the soundboard. These signals formed the output for the Genelec loudspeakers and, equivalently, for the shakers.

In their own admission, the professional “golden ears” working on its fine-tuning, at the time of the experiment the model was not yet well balanced in the higher octave range. Furthermore we did not apply any amplitude, nor spectral equalization to the signals feeding the shakers: such manipulations are needed to simulate the vibrotactile response of specific piano keyboards such as those investigated by Bank [11]. For our purpose, we just tuned the intensity level of the shakers based on the subjective impression of two expert piano players who helped realize

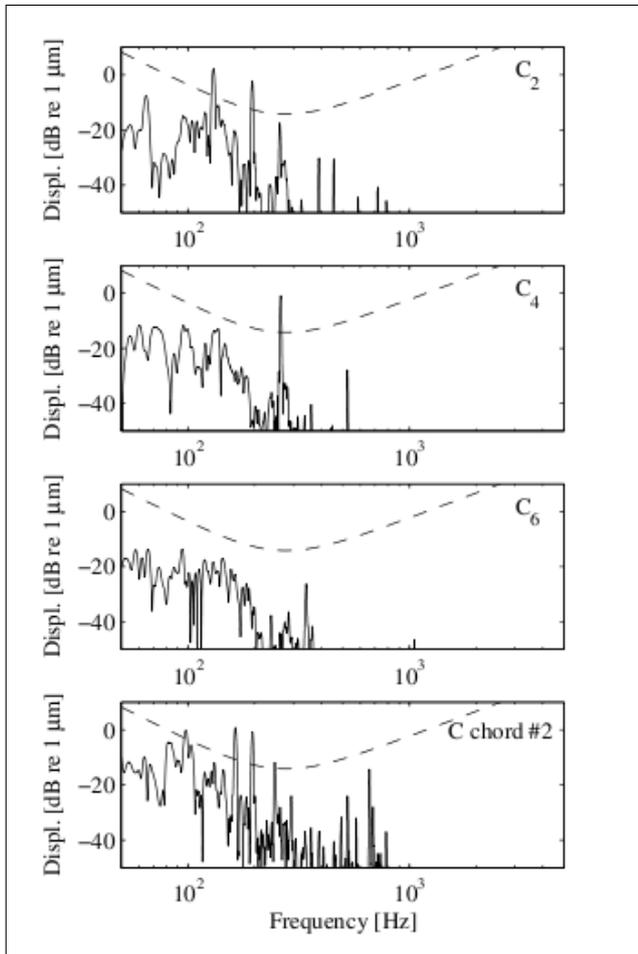


Figure 3. Spectra of the tactile response of a Bösendorfer grand piano measured at the keyboard (solid line), along with perceptual thresholds for the palm from Verrillo [12] (dashed line). Examples given for notes C_2 , C_4 , C_6 , and the C chord #2, all played at *forte* level.

the experiment.

Figure 4 illustrates the experimental setup. Powered by a Pioneer A-225 stereo amplifier, the two shakers were respectively positioned under the leftmost and mid part of the keyboard. In this way they emitted energy in correspondence of the lower octaves, whose keys are mostly responsible of producing vibrations falling within the tactile perceptual band (see Figure 3).

Subjects were exposed to three possible experimental configurations—refer also to the positions of the switches S1 and S2 in Figure 4:

C: Clavinova only (Clavinova speakers on; S1 off);

M: physical model with Genelec (Clavinova speakers off; S1 on; S2 off);

MS: physical model with Genelec and shakers (Clavinova speakers off; S1 on; S2 on).

The intensity levels were equalized so as to minimize the overall loudness changes across the three configurations, i.e., by setting the level of C midway M and MS. As a result, we measured Sound Pressure Levels (SPL) between

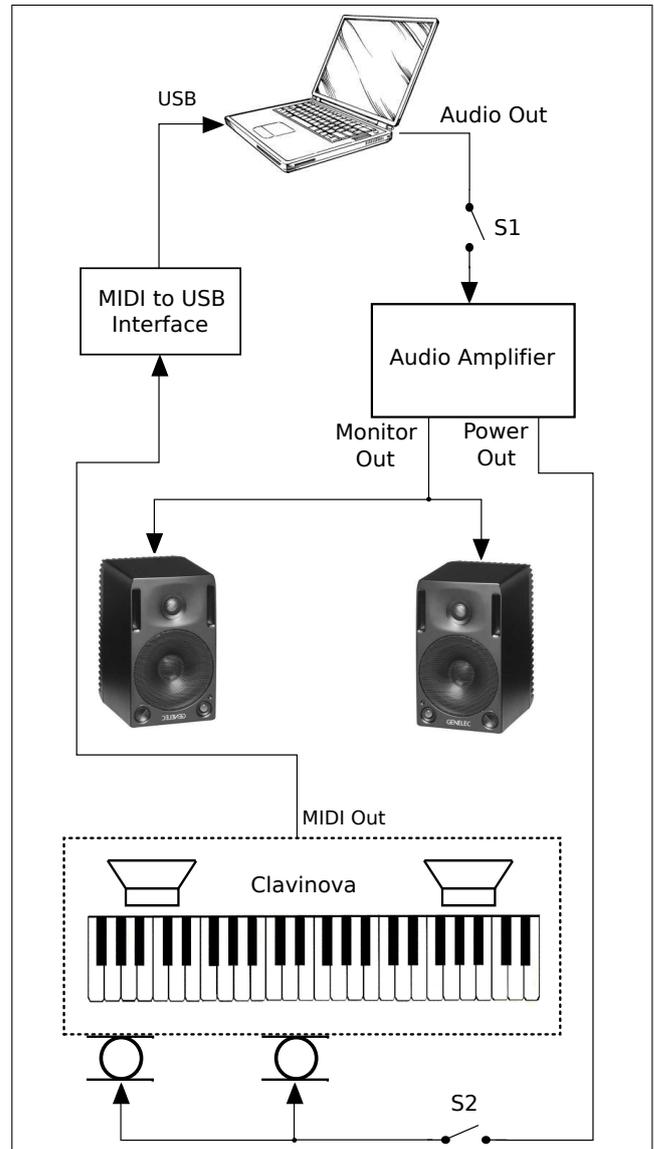


Figure 4. Experimental setup.

69 dB(A) (configuration M) and 71.5 dB(A) (configuration MS), with configuration C lying between the two. At any moment, by operating on the amplifier and by turning up or down the main volume of the Clavinova to predefined levels, the experimenter could easily switch between the different configurations. Subjects were not informed about the presence of the shakers below the keyboard, nor obviously about the fact that these devices could be switched on and off during the experiment.

2.3 Task

Subjects were initially asked to play the keys F , G , A , B of the lower four octaves (numbered 1, 2, 3 and 4 starting from the left side) in both directions. In more detail, they had to perform four tasks: playing i) an ascending *staccato* using only their right forefinger, ii) an ascending *legato* using at each octave the first four fingers of their right hand, iii) a descending *staccato* using only their right forefinger, and iv) a descending *legato* using at each octave the first four fingers of their right hand. The ascending pat-

terns were performed by moving upward across the keyboard, conversely the descending patterns were performed starting from the fourth octave down to the first one. The sixteen hot keys were marked with a red pencil to help unpracticed piano players accomplish the task without effort.

In addition to the tasks explained above, subjects with self-reported sufficient ability to play the piano were invited to perform freely on the instrument, for instance by playing one of their preferred songs.

2.4 Method

The four tasks were repeated three times across the different configurations C, M and MS, for a total of $4 \cdot 3 \cdot 3 = 36$ randomized *short* trials for each subject. This part of the experiment took about 35 minutes. Additional 15-20 minutes were required by the pianists to accomplish the free performance in the three configurations C, M and MS, summing to three additional *long* trials for these subjects.

At the end of every short trial, subjects marked the perceived sound quality on a scale ranging from 1 (very low) to 7 (very high). Long trials were instead judged qualitatively by the pianists. At the end of the experiment, the subjective skill in playing the piano was rated (1 to 7) along with the difficulty in performing the task (low, medium, high, very high). Only one subject reported a medium difficulty in performing the task, all the rest of the group otherwise rated the task difficulty to be low.

3. RESULTS

The aggregation of the judgments given by all the subjects during the short trials provides three sets of $4 \cdot 3 \cdot 9 = 108$ ratings, each set corresponding to a respective configuration. The mean values respectively amount to 3.741 for C, 3.981 for M, and 4.185 for MS (last row of Table 1).

A repeated-measures ANOVA conducted on the subjects' average ratings under the three configurations states that the mean values are significantly different at a 2% significance level: $F(2, 8) = 3.44$, $p = 0.017$. A similar analysis, made by restricting the attention to couples of such sets, shows that the difference is significant for both M and MS ($F(1, 8) = 15.46$, $p < 0.001$) and C and MS ($F(1, 8) = 3.44$, $p < 0.017$), whereas it is not significant between C and M ($F(1, 8) = 1.89$, $p = 0.19$). Furthermore, t-tests pairing the subjective judgments across the different conditions show p -values respectively equal to $p_{C \leftrightarrow M} = 0.176$, $p_{C \leftrightarrow MS} = 0.015$, and $p_{M \leftrightarrow MS} = 0.256$, with obvious meaning of the subscripts of p .

Concerning the free performance, the three pianists respectively opted for playing a pop song by the Beatles, an improvised jazz tune, and a *preludio* by Bach. All of them had a strong preference for the C configuration.

4. DISCUSSION

The different judgments existing between the single key patterns and free performance were almost certainly affected by the limited accuracy of the physical model in

| Mean Values | | | |
|-----------------|--------------|--------------|--------------|
| Subject | C | M | MS |
| Pianists | | | |
| 2 | 4.000 | 5.000 | 5.167 |
| 5 | 3.833 | 5.000 | 5.250 |
| 8 | 4.084 | 2.333 | 2.083 |
| Other Musicians | | | |
| 1 | 4.083 | 5.250 | 5.083 |
| 4 | 2.833 | 3.500 | 2.917 |
| 7 | 5.667 | 4.167 | 4.333 |
| 9 | 3.750 | 3.750 | 4.250 |
| Non Musicians | | | |
| 3 | 3.000 | 3.500 | 4.250 |
| 6 | 2.417 | 3.333 | 4.333 |
| Aggregate | | | |
| All | 3.741 | 3.981 | 4.185 |

Table 1. Mean values for the different configurations, by subject plus aggregate.

the higher octaves. It seems clear that as soon as the pianists heard a degradation of the sound quality, the vibrotactile modality lost any significance in their subjective judgment, and consequently the physics-based model was downgraded in their judgments.

Conversely, the results obtained by judgments on single notes ranging in the lower four octaves reveal that the vibrotactile feedback adds discrimination in the otherwise not significantly different judgment of the physics-based sound against the Clavinova samples.

However encouraging, the result on the aggregate data becomes less dramatic if reported on an individual basis. In fact, if we analyze the significance of the different configurations subject-by-subject then we discover that, once taken individually, subjects tend to grade the configurations mainly by their sound, and rarely the difference between M and MS gains significance.

Figure 5 shows, using double-sided arrows, subject-by-subject significance of the differences between configurations, obtained by computing t-values of the corresponding paired data at 5% significance level. The aggregate result presented in Section 3 is shown as well, at the bottom of the same figure.

From this figure, along with a look at the subjective mean values listed in Table 1, it can be observed that pianists seem to fairly weigh the vibrotactile modality, and rather base their judgments on robust decisions informed by the auditory modality—indeed, the question to the subjects was exactly that of rating the quality of the sound. As we move toward less specialized listeners, i.e. other instrument players, the corresponding mean values and related significances become more variate, including those of two subjects who do not appreciate any significant difference among the experimental configurations. Finally, non-experienced subjects exhibit less common judgments, including one subject who finds significant differences between M and MS.

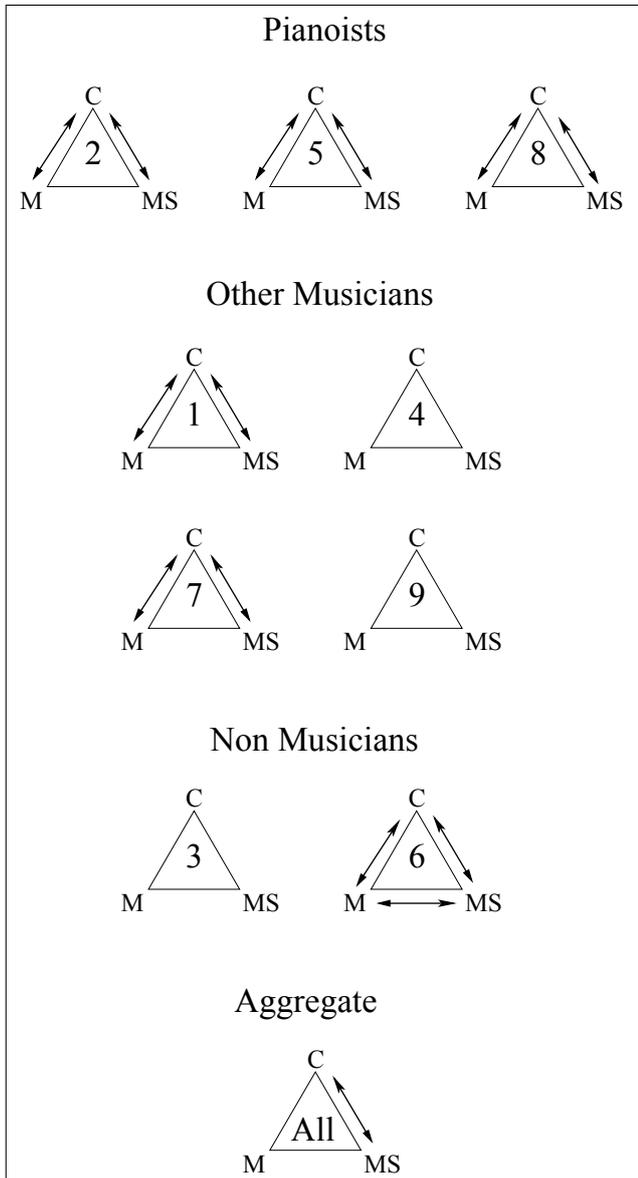


Figure 5. Significance of judgments, by subject plus aggregate. Double-sided arrows connect configurations whose different judgment is significant at 5% level, according to t-tests pairing the corresponding data. Subject number written at the center of the corresponding triangle.

Even if the number of subjects populating the different categories is too scarce to signify anything, a possibly relevant case record can be inferred from the subject-by-subject analysis. Specifically there may be a trend, potentially indicating a decreasing importance of the vibrotactile feedback for subjects who have more familiarity with musical sound evaluations, with minimal significance of the tactile modality for pianists. On the other hand, the results of non-experienced subjects may suggest that physically-consistent vibrotactile feedback can help unpracticed users enter into contact with musical interfaces such as keyboards, whose complexity of use is well known by practitioners.

In summary, the proposed experiment represents a first, far from being exhaustive attempt to understand the importance of vibrotactile cues in digital piano playing. Fur-

thermore, it shows limits in setup and methodology. Contingent difficulties for the experimenter in recruiting a sufficient number of subjects with different musical skill levels, together with the impossibility (due to limits in manpower) to prepare two acoustically different physics-based piano models having comparable sound quality, prevented to include a control session in which subjects could compare two models without vibrotactile feedback. Were musicians biased by the assumption that the vibration of a digital piano keyboard could not be changed during the tasks, or was the vibrotactile feedback equalized too roughly to elicit a definite sensation of quality improvement in pianists, are just two among the many questions that cannot be answered using our data.

5. CONCLUSIONS

In an experiment where subjects, after playing a digital piano, had to rate two different instrument models, we investigated the salience of vibrotactile cues as potential enhancers of the sound quality. Specifically, the test was made by switching on and off two shakers while subjects were performing a task with the latter model. Conversely, the former was played without any modification of its vibrotactile feedback.

Results say that, overall, the inclusion of the vibrotactile modality adds a significant improvement to the quality of the sound of the latter model. An analysis conducted subject-by-subject suggests that differences exist among the individual judgments on the same models, without a specific preference for the configuration enabling the vibrotactile feedback.

A classification of the subjects based on their knowledge of musical instruments, specifically the piano, was postulated to investigate musical skill as a possible predictor of preference. In the limits of the low number of subjects forming the three resulting classes, the subjective analyses indicate that pianists may be only weakly (albeit not significantly) influenced by the vibrotactile augmentation when making judgments on sound quality, whereas other musicians and non-musicians may be influenced more. Nevertheless, this conclusion is purely tentative at this stage of the experimentation.

Encouraged by this experience, we are planning to follow up with a more robust setup and overall experimental design. Concerning the setup, we will operate an equalization of the signals from the shakers meanwhile providing all the conditions for comparing configurations, in which the control of the auditory and vibrotactile feedback will be completely independent and orthogonal. Concerning the experimental methodology, new subjective tasks will be designed allowing for the extraction of more reliable figures of sound quality and realism of the overall experience.

Acknowledgments

The research leading to these results has received partial funding from the Joint Project E-PHASE, participated by the University of Verona and Viscount SpA, and from the

European Community's Seventh Framework Programme under FET-Open grant agreement 222107 NIW - Natural Interactive Walking.

6. REFERENCES

- [1] A. Askenfelt and E. V. Jansson, "On vibration and finger touch in stringed instrument playing," *Music Perception*, vol. 9, no. 3, pp. 311–350, 1992.
- [2] A. Galembo and A. Askenfelt, "Quality assessment of musical instruments - effects of multimodality," in *5th Triennial Conference of the European Society for the Cognitive Sciences of Music (ESCOM5)*, Hannover, Germany, 2003.
- [3] E. R. Miranda and M. M. Wanderley, *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Middleton, WI: A-R Editions, 2006.
- [4] W. Goebel and C. Palmer, "Tactile feedback and timing accuracy in piano performance," *Experimental Brain Research*, vol. 186, no. 3, pp. 471–479, 2008.
- [5] M. Keane, "Improving the upright piano," *Acoustics Australia*, vol. 35, no. 1, pp. 11–15, Apr. 2007.
- [6] C. Chafe, "Tactile audio feedback," in *Proc. Int. Computer Music Conf.*, Tokio, Japan, Sep. 10-15 1993, pp. 76–79.
- [7] E. Guizzo, "Keyboard maestro," *IEEE Spectrum*, vol. 47, no. 2, pp. 32–33, Feb. 2010.
- [8] Yamaha Corp., "Electronic Musical Instrument," 1988, US patent 5,189,242.
- [9] M. Altinsoy and S. Merchel, "Cross-modal frequency matching: Sound and whole-body vibration," in *Haptic and Audio Interaction Design*, ser. Lecture Notes in Computer Science, R. Nordahl, S. Serafin, F. Fontana, and S. Brewster, Eds. Springer Berlin / Heidelberg, 2010, vol. 6306, pp. 37–45. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15841-4_5
- [10] R. T. Verrillo, "Psychophysics of vibrotactile stimulation," *J. of the Acoustical Society of America*, vol. 77, no. 1, pp. 225–232, Jan. 1985.
- [11] B. Bank, "Vibration of piano keys – measurement results and implementation possibilities," Dept. Comp. Sci., University of Verona, Italy, Tech. Rep., 2008, internal project report.
- [12] R. T. Verrillo, "Effect of contactor area on the vibrotactile threshold," *J. of the Acoustical Society of America*, vol. 35, no. 12, pp. 1962–1966, 1963.
- [13] B. Bank, S. Zambon, and F. Fontana, "A modal-based real-time piano synthesizer," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 18, no. 4, pp. 809–821, 2010, special Issue on Virtual Analog Audio Effects and Musical Instruments.

ON COMPUTING MORPHOLOGICAL SIMILARITY OF AUDIO SIGNALS

Martin Gasser

Austrian Research Institute
for Artificial Intelligence
martin.gasser@ofai.at

Arthur Flexer

Austrian Research Institute
for Artificial Intelligence
arthur.flexer@ofai.at

Thomas Grill

Austrian Research Institute
for Artificial Intelligence
thomas.grill@ofai.at

ABSTRACT

Most methods to compute content-based similarity between audio samples are based on descriptors representing the spectral envelope or the texture of the audio signal only. This paper describes an approach based on (i) the extraction of spectro-temporal profiles from audio and (ii) non-linear alignment of the profiles to calculate a distance measure.

1. INTRODUCTION

Many real-world applications in the field of audio similarity would greatly benefit from an approach that explicitly models the temporal evolution of certain aspects of the signal and derives similarity values from this high-level description of the signal. Apart from being able to calculate similarities between signals that would be totally indistinguishable under a *bag-of-frames*-type [1, 2, 3] approach, such a method would also facilitate the classification of audio signals according to *morphological* descriptions.

Schaeffer [4, 5] proposed the description of *sound objects* according to a set of criteria, which include descriptions of the sound matter, the sound shape, and variation criteria. The notion of such sound-shapes appears in everyday applications e.g. with so-called *up-* and *downlifting* sounds as used in fields of sound-design and jingle production. Recently, some of those criteria have been implemented [6, 7, 8]. These works focus on classification of audio signals based on simple categories like “ascending” or “descending” based on standard audio features (e.g. loudness, spectral centroid, pitch).

The contribution of this paper is two-fold: First, we show some deficiencies of a standard descriptor (the spectral centroid [9]) for modeling spectral evolution over time. In particular, the spectral centroid is very sensitive to background noise if the noise level exceeds the level of the signal, whereas humans can easily spot spectral movement, even in the presence of strong (stationary) noise. We present a noise-robust and efficient approach to model spectro-temporal evolution based on tracking window-to-window cross-correlations of Constant-Q magnitude spectra over time.

Copyright: ©2011 Martin Gasser et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

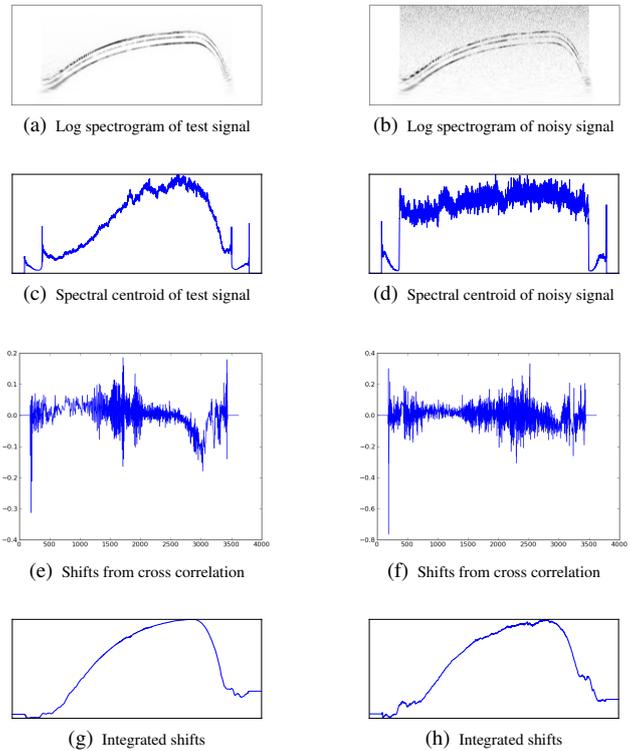


Figure 1. Profiles calculated from a clean test signal and a noisy signal ($SNR_{db} = 1$)

We also propose to calculate similarities between spectral evolution trajectories reconstructed from the aforementioned descriptor by using *Dynamic Time Warping* [16] and briefly evaluate our approach on a set of synthetic audio samples.

2. FEATURE EXTRACTION

A natural candidate for capturing the spectral evolution of a signal is the *spectral centroid* [9]. The spectral centroid is calculated as a weighted mean of the frequencies present in the signal. Thus, the lower the signal-to-noise ratio of a signal is, the less meaningful is the spectral centroid value. Figure 1(d) shows a typical result of a spectral centroid calculation on a noisy signal.

Another approach to capturing the spectral movement of signals is F0-tracking [10]. Since we want to be able to capture spectral evolutions of highly inharmonic sounds as well, F0-tracking approaches have been rejected in the first place.

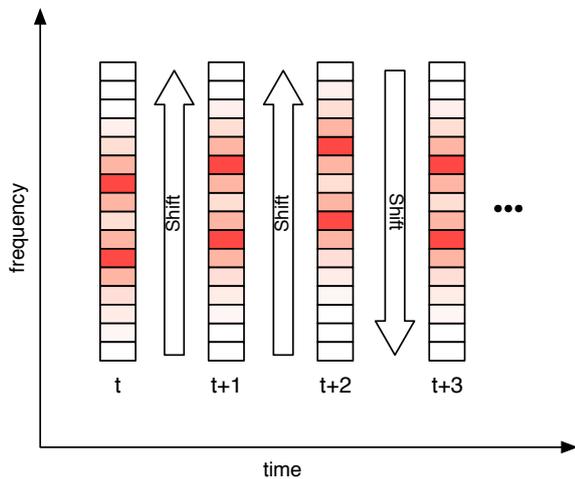


Figure 2. Shifted log-scaled magnitude spectra

Instead of trying to find instantaneous descriptors like F0 or the spectral centroid, we take a different route: Based on the observation that successive logarithmically scaled magnitude spectra calculated from a coherent spectral movement are cross-correlated (see figure 2), we cross-correlated Constant-Q magnitude spectra, from the cross-correlations we derive the hypothetical shift values to optimally align two magnitude spectra, and finally we calculate the final trajectory by cumulative summing of the shift values. Figure 1 demonstrates the weakness of spectral centroid under noisy conditions (Fig. 1(c) and Fig. 1(d)) and how we derive the final spectral profile from the shift values (Fig. 1(e)–Fig. 1(h)).

To calculate the logarithmically scaled short-time spectra of the input signal, we apply a Constant-Q transform [11] (CQT) and use the absolute value of the result.

For the Constant-Q analysis, we use a frequency resolution of 32 bins per octave, with a minimum frequency of 50 Hz and a maximum frequency corresponding to the Nyquist frequency (the sampling rate of the signals being 44.1 kHz). The hop size for the analysis is set to 1.451 ms and the time domain basis functions of the CQT are tapered with Hann windows. The window size resulting from the aforementioned parameters is 371.5 ms. We use the CQT algorithm as described by Brown and Puckette [12] and discard 98% of the coefficients in the spectral kernel SK by thresholding it with $0.01 * \max(SK)$, taking advantage of sparse matrix multiplications.

2.1 Mathematical background

According to the correlation theorem [13], the cross-correlation r of two signals can be calculated efficiently in the Fourier domain as:

$$\mathbf{G}_a = \mathcal{F}\{g_a\} \quad , \quad \mathbf{G}_b = \mathcal{F}\{g_b\}$$

$$R = \frac{\mathbf{G}_a \mathbf{G}_b^*}{|\mathbf{G}_a \mathbf{G}_b^*|} \quad , \quad r = \mathcal{F}^{-1}\{R\}$$

where \mathbf{G}_a and \mathbf{G}_b are the Fourier transforms of the sig-

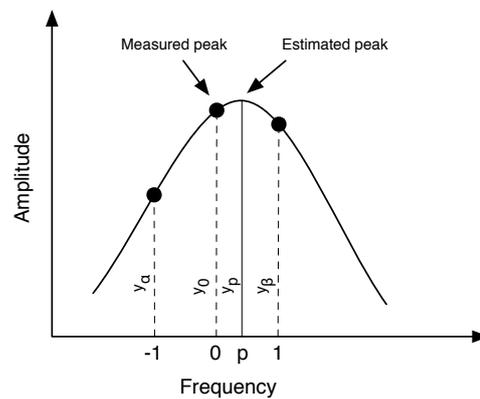


Figure 3. Fitting a parabola to the correlation function

nals g_a and g_b , respectively, and \mathbf{G}_b^* denotes the complex conjugate of \mathbf{G}_b .

By peak-picking the cross-correlation function, the shift factors can be recovered.

One problem that we encountered was that the frame-to-frame shift factors were potentially smaller than the resolution of the spectral analysis. Therefore, the resulting shift factors were much too coarse and in many cases, the correlation function always peaked at lag 0. Our solution to the problem was to use correlation interpolation [14] by fitting a parabola to the peak and the two surrounding points in the cross-correlation function (see figure 3).

$$p = \frac{1}{2} \frac{y_\alpha - y_\beta}{y_\alpha - 2y_0 + y_\beta} \quad (1)$$

By adding the value p to the index of the peak index of the cross correlation function, the final shifting factor for a pair of magnitude spectra is computed.

3. SIMILARITY COMPUTATION

We assume that similar audio signals show a similar spectral evolution in time. In order to measure the similarity between two signals, we (i) align a low dimensional representation of the spectral evolution of the signals and we (ii) derive a similarity measure from the quality of the optimal alignment.

A well-researched method that solves the problem of non-linear alignment of time series is Dynamic Time Warping (DTW) [15, 16]. DTW is a dynamic programming algorithm, that is, it calculates a matrix of partial optimal solutions to sub-problems and finds the optimal solution of the problem in a back-tracking manner. The result of DTW is a list of matchings $(a_i, b_j)_k$, where a match in time step k relates the elements a_i and b_j from the time series a and b (a_i and b_j are the elements at positions i and j in the time series a and b , respectively). By summing and normalizing the distances between matched elements, a distance measure for the time series is derived.

Figure 4 shows a simple example of DTW (the first column shows the distance matrix and the optimal path, the second column shows the resulting point-to-point align-

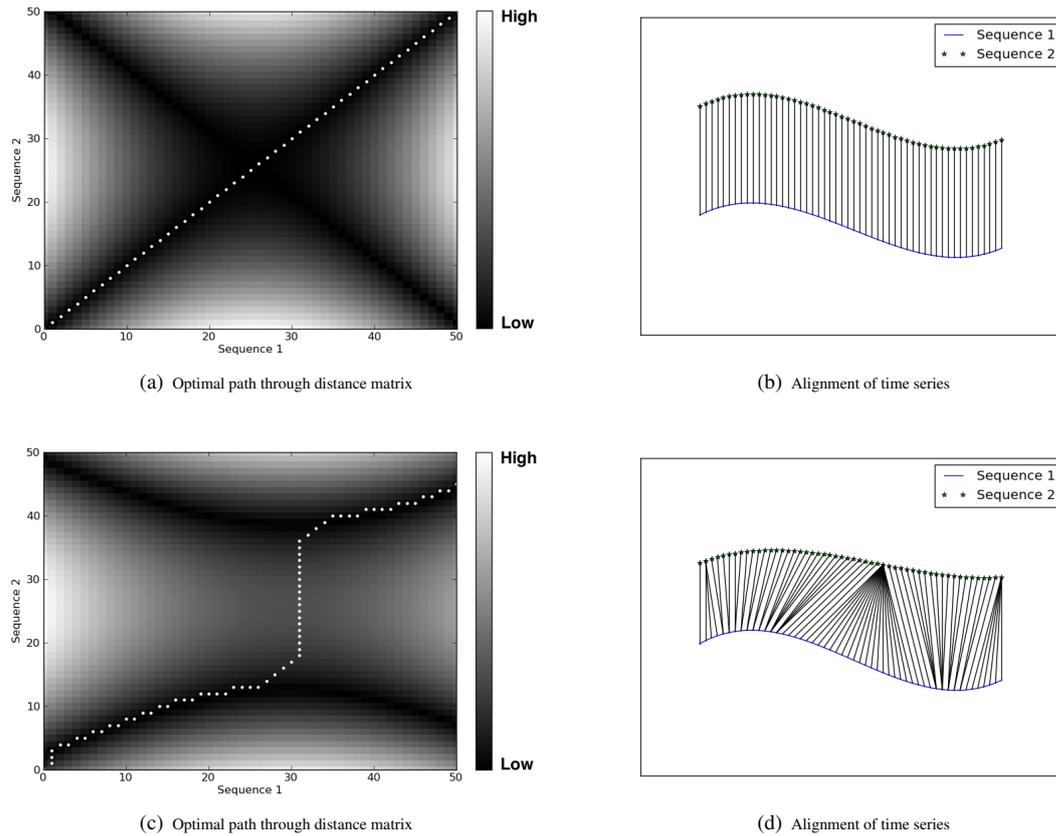


Figure 4. DTW example

ment). Whereas in figure 4(a)/figure 4(b) the two time series are identical, figure 4(c)/figure 4(d) shows two slightly different series, yielding a non-zero distance.

In order to be independent of translations along the y axis, we used a variant of DTW called Derivative Dynamic Time Warping [17], that is, instead of directly matching values of time series, we matched their first order differences.

Our algorithm for calculating morphological similarities can be summed up as follows:

1. Calculate the Constant-Q magnitude spectrum of the signals
2. For each pair of spectral frames, calculate the cross-correlation, pick the peaks and derive the shift value
3. Integrate shifts (compute cumulative sums)
4. To account for different signal lengths and to reduce computation time, resample the integrals of the spectral evolution trajectories to length n (we used $n = 100$)
5. Align the trajectories with DTW and compute the distances

4. EVALUATION

To evaluate our approach, we generated short audio samples consisting of a sinusoidal oscillator following a set of

pitch envelopes $\{e_i(x) | i \in [0, 17]\}$ modeling up-down and down-up movements.

$$t_\alpha(x) = \begin{cases} \sin(\frac{x}{\alpha} \cdot \frac{\pi}{2}), & x \leq \alpha \\ \sin(\frac{\pi}{2} + \frac{x-\alpha}{1-\alpha} \cdot \frac{\pi}{2}), & x > \alpha \end{cases} \quad (2)$$

$$e_i(x) = \begin{cases} t_{(i+1)/10}, & 0 \leq i < 9 \\ 1 - t_{(i-8)/10}, & 9 \leq i \leq 17 \end{cases} \quad (3)$$

See figure 5 for a plot of the resulting 18 functions.

As a proof of concept, we calculated DTW distances directly on the pitch envelopes without synthesizing audio and extracting the trajectories in the first place. Figure 6 demonstrates the theoretical applicability of the DTW approach to our problem: For each pair of pitch envelopes, the DTW distance is mapped to a color (dark corresponds to low, light to high values). As can be seen, the distances vary correspondingly to the choice of the α parameter in equation 2.

From the profiles in figure 5, we generated short audio samples by applying a bank of resonators with time-varying filter frequencies to an excitation signal consisting of white noise. The resonator frequencies were set to $(env * 1.1, env * 1.6, env * 2.1, env * 2.2)$, where env is the value of a prototype envelope. We also generated two sets of signals with added white noise ($SNR_{db} = 6$ and 1, respectively) and calculated their distances to the clean signals. Figures 7(a)- 7(c) show that the distance measure is

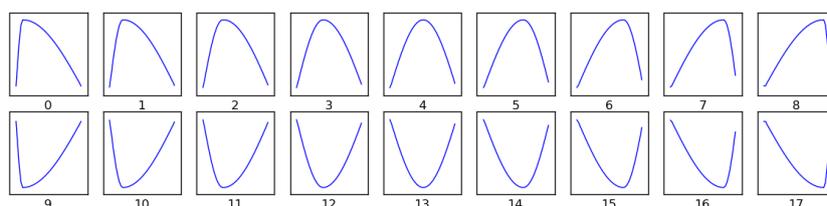


Figure 5. Pitch envelopes used for the evaluation

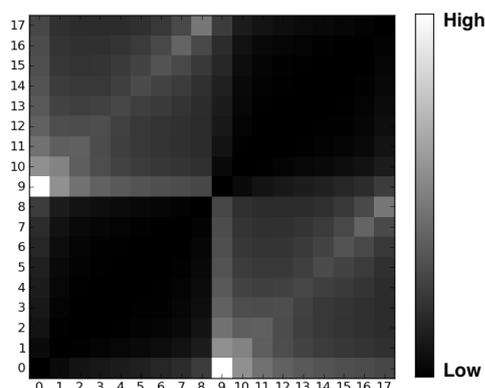


Figure 6. DTW distance matrix calculated from the 18 prototypes

very robust, even under very noisy conditions.

5. CONCLUSIONS AND FUTURE WORK

We have presented an approach to calculating similarity between audio samples based on morphological criteria. We model the spectro-temporal evolution of signals by calculating the pairwise interpolated cross-correlation between successive log-scaled short-time magnitude spectra and integrating the resulting shift values to a trajectory. To calculate similarities, trajectories are aligned with Derivative Dynamic Time Warping and a distance value is derived from the resulting DTW path.

In this paper, we have only considered spectral evolution of signals; we plan on using the same methodology to calculate similarities from loudness functions as well.

We also think that the method has potential for interactive retrieval of audio samples, since it should be straightforward for users to directly draw the desired spectral evolution trajectory. We will investigate this possibility in future research.

6. ACKNOWLEDGMENTS

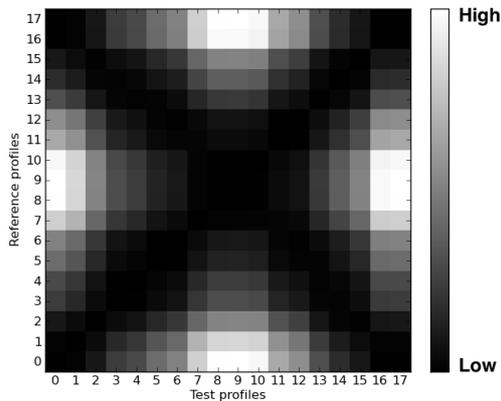
This research is supported by the Austrian Science Fund (FWF, P21247, and Z159) and the Vienna Science and Technology Fund (WWTF, project “Audiominer”)

7. REFERENCES

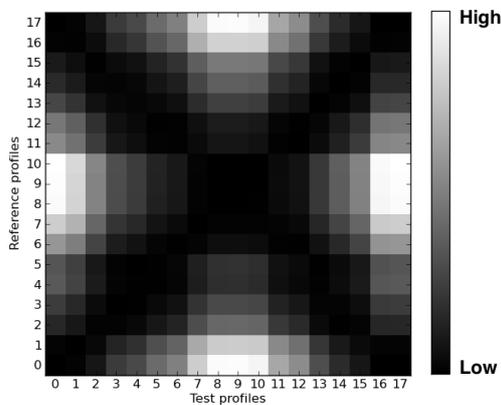
- [1] M. Mandel and D. Ellis, “Song-level features and support vector machines for music classification,” in *Proceedings of the 6th International Conference on Music Information Retrieval*, London, United Kingdom, 2005.
- [2] E. Pampalk, “Computational models of music similarity and their application in music information retrieval,” Ph.D. dissertation, Vienna University of Technology, Vienna, Austria, March 2006. [Online]. Available: <http://www.ofai.at/~elias.pampalk/publications/pampalk06thesis.pdf>
- [3] J.-J. Aucouturier, B. Defreville, and F. Pachet, “The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music,” *Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.
- [4] P. Schaeffer, *Traité des objets musicaux*. Editions du Seuil, Paris, 1966.
- [5] M. Chion, *Guide des objets sonores, Pierre Schaeffer et la recherche musicale*. Ina-GRM/Buchet-Chastel, Paris, 1983.
- [6] J. Ricard and P. Herrera, “Morphological sound description computational model and usability evaluation,” in *AES 116th Convention*, 2004. [Online]. Available: <files/publications/AES116-jricard.pdf>
- [7] J. Bloit, N. Rasamimanana, and F. Bevilacqua, “Modeling and segmentation of audio descriptor profiles with segmental models,” *Pattern Recogn. Lett.*, vol. 31, pp. 1507–1513, September 2010. [Online]. Available: <http://dx.doi.org/10.1016/j.patrec.2009.11.003>
- [8] G. Peeters and E. Deruty, “Sound indexing using morphological description,” *IEEE Transactions on Audio, Speech & Language Processing*, vol. 18, no. 3, pp. 675–687, 2010.
- [9] G. Peeters, “A large set of audio features for sound description (similarity and classification) in the CUIDADO project,” IRCAM, Tech. Rep., 2004.
- [10] A. de Cheveigne and H. Kawahara, “Yin, a fundamental frequency estimator for speech and music,” *The Journal of the Acoustical Society of America*, vol.

111, no. 4, pp. 1917–1930, 2002. [Online]. Available: <http://link.aip.org/link/?JAS/111/1917/1>

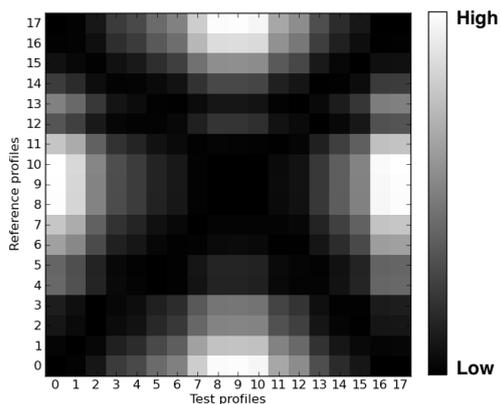
- [11] J. C. Brown, “Calculation of a constant q spectral transform,” *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.
- [12] J. Brown and M. Puckette, “An efficient algorithm for the calculation of a constant q transform,” *Journal of the Acoustical Society of America*, vol. 92, no. 5, pp. 2698–2701, 1992.
- [13] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.
- [14] Q. Tian and M. N. Huhns, “Algorithms for subpixel registration,” *Computer Vision, Graphics, and Image Processing*, vol. 35, no. 2, pp. 220–233, 1986.
- [15] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.
- [16] D. J. Berndt and J. Clifford, “Finding patterns in time series: a dynamic programming approach,” pp. 229–248, 1996. [Online]. Available: <http://portal.acm.org/citation.cfm?id=257938.257961>
- [17] E. J. Keogh and M. J. Pazzani, “Derivative Dynamic Time Warping,” in *In First SIAM International Conference on Data Mining (SDM'2001)*, 2001. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.6686>



(a) Reference profiles to themselves



(b) Reference profiles to noisy version ($SNR_{db} = 6$)



(c) Reference profiles to noisy version ($SNR_{db} = 1$)

Figure 7. Distance matrices of reference profiles to noisy versions

SOUND SPATIALIZATION CONTROL BY MEANS OF ACOUSTIC SOURCE LOCALIZATION SYSTEM

Daniele Salvati

AVIRES Lab.

Dep. of Math. and Computer Science

University of Udine, Italy

daniele.salvati@uniud.it

Sergio Canazza

Sound and Music Computing Group

Dep. of Information Engineering

University of Padova, Italy

canazza@dei.unipd.it

Antonio Rodà

AVIRES Lab.

Dep. of Math. and Computer Science

University of Udine, Italy

antonio.roda@uniud.it

ABSTRACT

This paper presents a system for controlling the sound spatialization of a live performance by means of the acoustic localization of the performer. Our proposal is to allow a performer to directly control the position of a sound played back through a spatialization system, by moving the sound produced by its own musical instrument. The proposed system is able to locate and track the position of a sounding object (e.g., voice, instrument, sounding mobile device) in a two-dimensional space with accuracy, by means of a microphone array. We consider an approach based on Generalized Cross-Correlation (GCC) and Phase Transform (PHAT) weighting for the Time Difference Of Arrival (TDOA) estimation between the microphones. Besides, a Kalman filter is applied to smooth the time series of observed TDOAs, in order to obtain a more robust and accurate estimate of the position. To test the system control in real-world and to validate its usability, we developed a hardware/software prototype, composed by an array of three microphones and a Max/MSP external object for the sound localization task. We have got some preliminary successful results with a human voice in real moderately reverberant and noisy environment and a binaural spatialization system for headphone listening.

1. INTRODUCTION

The spatialization of sound plays an increasingly important role in electroacoustic music performance from the twentieth century. A first widely studied aspect concerns techniques and algorithms for the placement of sounds in a virtual space. In 1971, John Chowning proposed a pioneering system that simulated the movement of sound sources in the space [1]. Afterwards, Moore [2] developed a general model that drew on basic psychophysics of spatial perception and on work in room acoustics, relying on the precedence effect. To date, many techniques are used for spatialization, such as: holographic approach [3] like 3D panning (Vector Base Amplitude Panning [4]) and Ambisonics [5], Wavefield Synthesis [6], and transaural techniques based on an idea by Schroeder [7]. Besides the methods based on

virtual environments using loudspeakers, we mention the theory and practice of 3D sound reproduction using headphones, that requires the filtering of sound streams with Head Related Transfer Functions (HRTFs) [8].

Another important aspect of sound spatialization is related to the control task. Recently, research has begun to investigate control issues, especially related to gesture controlled spatialization of sound in live performance [9]. Most systems of control make use of a separate interface and a specific performer (usually not on stage) to control the movement of sounds. In that sense, the evolution of control systems was mainly related to the design of different equipments, such as multichannel devices with faders, control software with mouse and joystick for two-dimensional movement, sophisticated software with 3D virtual reality display [10], sensors interfaces such as data gloves based system, head trackers and camera-based tracking systems [11].

In [12], the authors propose a system to allow real-time gesture control of spatialization in a live performance setup, by the performers themselves. This gives to the performers the control over the spatialization of the sound produced by their own instrument, during the performance of a musical piece. In the same way, our system provides the capability to control the spatialization of sound by the performer himself, using the potentiality offered by microphone array signal processing. Recently, microphone array signal processing is increasingly being used in human computer interaction systems, for example the new popular interface Microsoft Kinect incorporates a microphone array to conduct acoustic source localization and noise suppression to improve voice recognition. The microphone array approach has the advantage that the performer does not have to wear any sensor or device which can be a hindrance to his/her movements; moreover, it can replace or integrate camera-based tracking systems that can have problems with the low lighting of the concert hall.

This paper presents a system for controlling the sound spatialization of a live performance by means of the acoustic localization of the performer. Our proposal is to allow a performer to directly control the position of a sound played back through a spatialization system, by moving the sound produced by its own musical instrument. The proposed system is able to locate and track the position of a sounding object (e.g., voice, instrument, sounding mobile device) in a two-dimensional space with accuracy, by means of a mi-

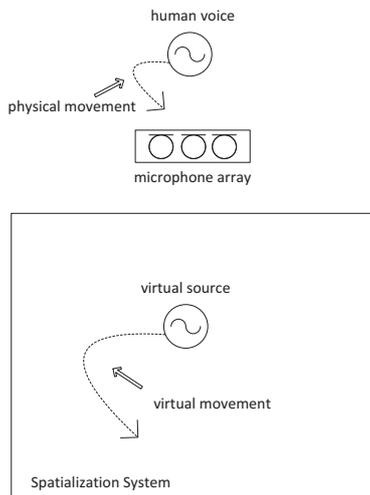


Figure 1. Sound spatialization control setup.

crophone array (see Figure 1).

The paper is organized as follows: after presenting the system architecture in Section 2, we summarize the algorithms for the time delay estimation in Section 3. Section 4 describes the Kalman filter to smooth the observed TDOAs. In Section 5, we illustrate the two-dimensional position estimation. Finally, Section 6 shows the developed prototype and some experimental results with human voice.

2. SYSTEM ARCHITECTURE

The system consists of three main components: i) a microphone array for signal acquisition; ii) signal processing techniques for sound localization; iii) a two-dimensional mapping function for controlling the sound spatialization parameters.

The array is composed by three microphones arranged in an uniform linear placement (in near-field environment, three microphones are the bare minimum to locate source in a plane). Signal processing algorithms estimate the sound source position in a horizontal plane by providing its Cartesian coordinates. Last component regards how to transform the x-y coordinates of the real source into parameters for the virtual source movement, depending on the spatialization setup. To this purpose, we mention the Spatial Sound Description Interchange Format (SpatDIF) [13], a format to describe, store and share spatial audio scenes across 2D/3D audio applications and concert venues. However, this paper is mainly focused on the localization task.

Figure 2 summarizes the block diagram of system. A widely used approach to estimate the source position consists in two steps: in the first step, a set of TDOAs are estimated using measurements across various combinations of microphones; in the second step, knowing the position of sensors and the velocity of sound, the source positions is calculated by means of geometric constraints and using approximation methods such as least-square techniques [14].

The traditional technique to estimate the time delay be-

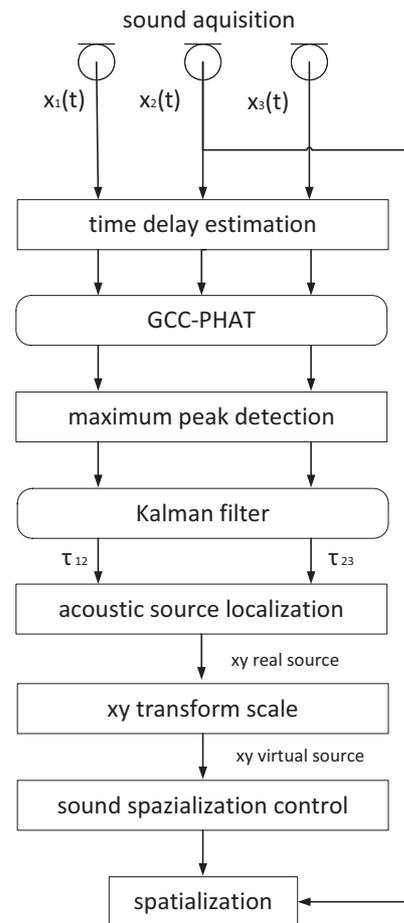


Figure 2. Block diagram of system.

tween a pair of microphones is the GCC-PHAT [15]. Following this approach, the maximum peak detection of the GCC functions provides the estimation of the TDOAs between microphones 1-2 and 2-3. Then, a Kalman filter is applied in order to smooth in time [16] the two estimated TDOAs. The Kalman filter provides a robust and accurate estimation of τ_{12} and τ_{23} , moreover it is able to provide a source position estimation, also if the TDOA estimation task misses the target in some frame of analysis.

3. TIME DELAY ESTIMATION

GCC [15] is the classic method to estimate the relative time delay associated with the acoustic signals received by a pair of microphones in a moderately reverberant and noisy environment [17, 18]. It basically consist in a cross-correlation followed by a filter that aims at reducing the performance degradation due to additive noise and multipath channel effects. The signals received at the two microphones $x_1(t)$ and $x_2(t)$ may be modeled as

$$\begin{aligned} x_1(t) &= h_1(t) * s(t) + n_1(t) \\ x_2(t) &= h_2(t) * s(t - \tau) + n_2(t) \end{aligned} \quad (1)$$

where τ is the relative signal delay of interest, $h_1(t)$ and $h_2(t)$ represent the impulse responses of the reverberant

channels, $s(t)$ is the sound signal, $n_1(t)$ and $n_2(t)$ correspond to uncorrelated noise, and $*$ denotes linear convolution. The GCC in the frequency domain is

$$R_{x_1x_2}(t) = \sum_{w=0}^{L-1} \Psi(w) S_{x_1x_2}(w) e^{\frac{jw\tau}{L}} \quad (2)$$

where w is the frequency index, L is the number of samples of the observation time, $\Psi(w)$ is the frequency domain weighting function, and the cross-spectrum of the two signals is defined as

$$S_{x_1x_2}(w) = E\{X_1(w)X_2^*(w)\} \quad (3)$$

where $X_1(w)$ and $X_2(w)$ are the Discrete Fourier Transform (DFT) of the signals and $*$ denotes the complex conjugate. GCC is used for minimizing the influence of moderate uncorrelated noise and moderate multi-path interference, maximizing the peak in correspondence of the time delay.

The relative time delay τ is obtained by an estimation of the maximum peak detection in the filter cross-correlation function

$$\hat{\tau} = \underset{t}{\operatorname{argmax}} R_{x_1x_2}(t). \quad (4)$$

PHAT [15] weighting is the traditional and most used function. It places equal importance on each frequency by dividing the spectrum by its magnitude. It was later shown that it is more robust and reliable in realistic reverberant conditions than other weighting functions designed to be statistically optimal under specific non-reverberant noise conditions [19]. The PHAT weighting function normalizes the amplitude of the spectral density of the two signals and uses only the phase information to compute the GCC

$$\Psi_{\text{PHAT}}(w) = \frac{1}{|S_{x_1x_2}(w)|}. \quad (5)$$

GCC works very well with human voice, and it is traditional used with human speech. Instead, it is widely acknowledged that GCC performance is dramatically reduced in case of harmonic sound, or generally pseudo-periodic sounds. In fact, segments of pseudo-periodic sound, when filtered by GCC, have less influence on the deleterious effects of noise and reverberation. Thus, sound objects in which the harmonic component greatly prevails on the noisy part (for example musical instruments like flute and clarinet) require new considerations for the localization task that have to be investigated.

4. TIME DELAY FILTERING USING KALMAN THEORY

The Kalman filter [20] is the optimal recursive Bayesian filter for linear systems observed in the presence of Gaussian noise. We consider that the state of the TDOA estimation could be summarized by two variables: the position τ and velocity v_τ . These two variables are the elements of the state vector \mathbf{x}_t

$$\mathbf{x}_t = [\tau, v_\tau]^T. \quad (6)$$

The process model relates the state at a previous time $t-1$ with the current state at time t , so we can write

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_{t-1} \quad (7)$$

where \mathbf{F} is the transfer matrix and \mathbf{w}_{t-1} is the process noise associated with random events or forces that directly affect the actual state of the system. We assume that the components of \mathbf{w}_{t-1} have Gaussian distribution with zero mean normal distribution with covariance matrix \mathbf{Q}_t , $\mathbf{w}_{t-1} \sim N(0, \mathbf{Q}_t)$. Considering the dynamical motion, if we measured the system to be at position x with some velocity v at time t , then at time $t + dt$ we would expect the system to be located at position $x + v \cdot dt$, thus this suggests that the correct form for \mathbf{F} is

$$\mathbf{F} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}. \quad (8)$$

At time t an observation \mathbf{z}_t of the true state \mathbf{x}_t is made according to the measurement model

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \quad (9)$$

where \mathbf{H} is the observation model which maps the true state space into the observed space and \mathbf{v}_t is the observation noise which is assumed to be zero mean Gaussian white noise with covariance \mathbf{R}_t , $\mathbf{v}_t \sim N(0, \mathbf{R}_t)$. We only measure the position variables, i.e. the maximum peak detection of GCC-PHAT. Hence, we have

$$\mathbf{z}_t = \hat{\tau} \quad (10)$$

and then we have

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \quad (11)$$

The filter equations can be divided into a prediction and a correction step. The prediction step projects forward the current state and covariance to obtain an a priori estimate. After that the correction step uses a new measurement to get an improved a posteriori estimate. In prediction step the time update equations are

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1}, \quad (12)$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^T + \mathbf{Q}_{t-1}, \quad (13)$$

where \mathbf{P}_t denotes the error covariance matrix. In the correction step the measurement update equations are

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1}), \quad (14)$$

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_{t|t-1}, \quad (15)$$

where \mathbf{I} is the identity matrix and so-called Kalman gain matrix is

$$\mathbf{K}_t = \mathbf{P}_{t-1|t-1} \mathbf{H}^T (\mathbf{H}_t \mathbf{P}_{t-1|t-1} \mathbf{H}_t^T + \mathbf{R}_t)^{-1}. \quad (16)$$

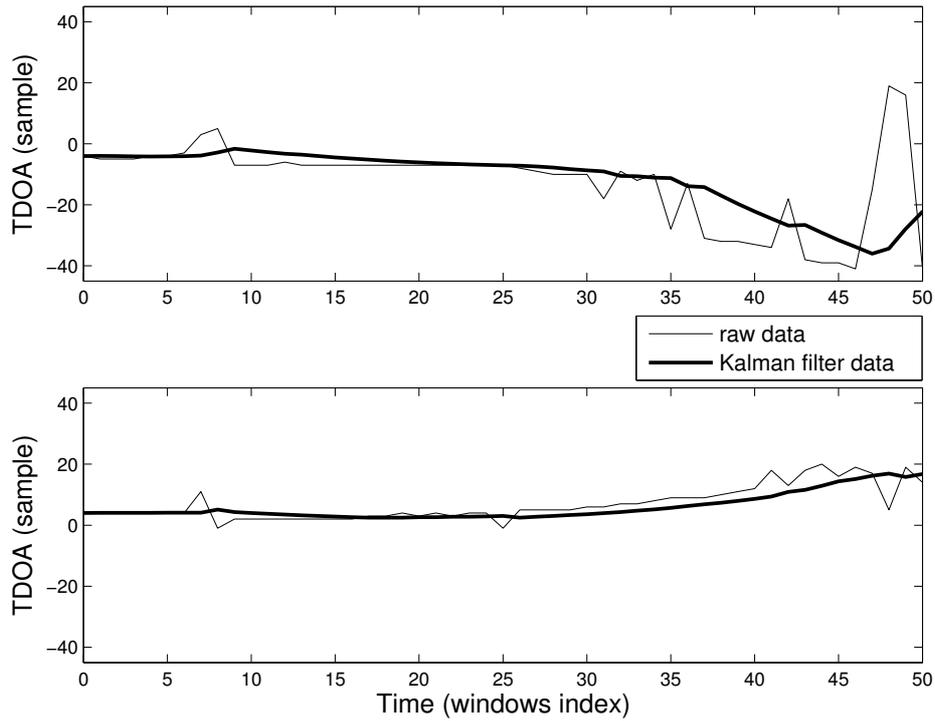


Figure 5. Comparison of TDOA estimation of human voice (on the top between microphone 1 and 2, below between 2 and 3). The Kalman filter data is represented by black lines and raw data by gray lines.

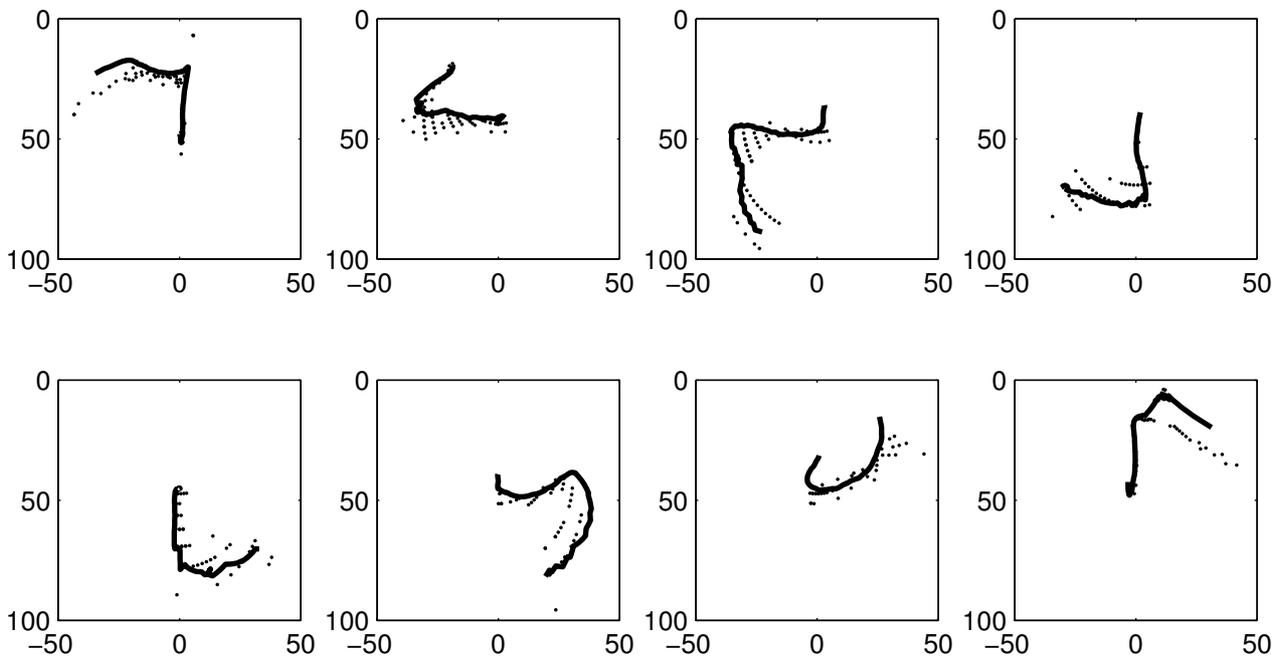


Figure 6. Acoustic source localization performance. Human voice moves in different directions (dots are the raw data), xy axis are in cm.

represent the filtered Kalman data.

Finally, the control interface was tested in connection with a sound spatialization system. VST plug-in based on binaural spatialization for headphone listening was used. An informal test of the system showed encouraging results: the performer has in fact been able to control in real time the position of a virtual sound source by small movements (of the order of tens of centimeters) of his/her mouth.

7. CONCLUSIONS

This paper presented a system that exploits microphone array signal processing to allow a performer to use the movement of a sounding object (voice, instrument, sounding mobile device) to control a sound spatialization system. A hardware/software prototype, composed by a linear array of three supercardioid microphones and a Max/MSP external object, was developed. Preliminary results with human voice show that the system can be used in a real scenario. GCC-PHAT and Kalman filter provides an accurate time delay estimation in moderately reverberant and noisy environment. However, new investigation must be done in order to work with harmonic sounds, or generally pseudo-periodic sounds, such as those traditional musical instruments in which the harmonic component greatly prevails on the noisy part. This is the main focus of our future work, which also will regard the use of the interface in a real live performance setup with a loudspeaker based spatialization system.

8. REFERENCES

- [1] J. Chowning, "The simulation of moving sound sources," *Journal of the Audio Engineering Society*, vol. 19, no. 1, pp. 2–6, 1971.
- [2] F. R. Moore, "A general model for spatial processing of sounds," *Computer Music Journal*, vol. 7, no. 3, pp. 6–15, 1982.
- [3] A. J. Berkhout, "A holographic approach to acoustic control," *Journal of the Audio Engineering Society*, vol. 36, no. 12, pp. 977–995, 1988.
- [4] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *Journal of the Acoustical Society of America*, vol. 45, no. 6, pp. 456–466, 1997.
- [5] M. A. Gerzon, "Ambisonics in multichannel broadcasting and video," *Journal of the Acoustical Society of America*, vol. 33, pp. 959–871, 1985.
- [6] D. de Vries, E. W. Start, and V. G. Valstar, "The wave-field synthesis concept applied to sound reinforcement restriction and solutions," in *Audio Engineering Society Convention*, 2 1994.
- [7] M. Schroeder, "Improved quasi-stereophony and colorless artificial reverberation," *Journal of the Acoustical Society of America*, vol. 33, no. 8, pp. 1061–1064, 1961.
- [8] F. Wightman and D. Kistler, "Headphone stimulation of free field listening I: stimulus synthesis," *Journal of the Acoustical Society of America*, vol. 85, pp. 858–867, 1989.
- [9] M. Marshall, J. Malloch, and M. Wanderley, "Gesture control of sound spatialization for live musical performance," in *Gesture-Based Human-Computer Interaction and Simulation*. Springer Berlin / Heidelberg, 2009, vol. 5085, pp. 227–238.
- [10] M. Naef and D. Collicott, "A VR interface for collaborative 3d audio performance," in *Proc. International Conference on New Interfaces for Musical Expression*, 2006, pp. 57–60.
- [11] M. Wozniowski, Z. Settel, and J. Cooperstock, "A framework for immersive spatial audio performance," in *Proc. International Conference on New Interfaces for Musical Expression*, 2006, pp. 144–149.
- [12] M. Marshall, N. Peters, A. Jensenius, J. Boissinot, M. Wanderley, and J. Braasch, "On the development of a system for gesture control of spatialization," in *Proc. International Computer Music Conference*, 2006.
- [13] N. Peters, S. Ferguson, and S. McAdams, "Towards a spatial sound description interchange format (Spat-DIF)," *Canadian Acoustics*, vol. 35(3), pp. 64–65, 2007.
- [14] R. O. Schmidt, "A new approach to geometry of range difference location," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-8 Issue: 6, pp. 821–835, 1972.
- [15] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 4, pp. 320–327, May 1976.
- [16] U. Klee, T. Gehrig, and J. McDonough, "Kalman filters for time delay of arrival-based source localization," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–15, 2006.
- [17] B. Champagne, S. Berdard, and A. Stephenne, "Performance of time-delay estimation in the presence of room reverberation," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 148–152, 1996.
- [18] J. Chen, Y. Huang, and J. Benesty, "A comparative study on time delay estimation in reverberant and noisy environments," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005, pp. 21–24.
- [19] M. Omologo and P. Svaizer, "Acoustic event localization using a crosspower-spectrum based technique," in *Proc. IEEE ICASSP*, vol. 2, 1994, pp. 273–276.
- [20] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

AN ANALOG I/O INTERFACE BOARD FOR AUDIO ARDUINO OPEN SOUND CARD SYSTEM

Smilen Dimitrov

Medialogy, Aalborg University Copenhagen
sd@{imi,create}.aau.dk

Stefania Serafin

Medialogy, Aalborg University Copenhagen
sts@{imi,create}.aau.dk

ABSTRACT

AUDIOARDUINO [1] is a system consisting of an ALSA (Advanced Linux Sound Architecture) audio driver and corresponding microcontroller code; that can demonstrate full-duplex, mono, 8-bit, 44.1 kHz soundcard behavior on an FTDI based ARDUINO. While the basic operation as a soundcard can be demonstrated with nothing more than a pair of headphones and a couple of capacitors - modern PC soundcards typically make use of multiple signal standards; and correspondingly, multiple connectors.

The usual distinction that typical off-the-shelf stereo soundcards make, is between *line-level* signals (line-in/line-out) - and those not conforming to this standard (such as microphone input/speaker output). To provide a physical illustration of these issues in soundcard design, this project outlines an open design for a simple single-sided PCB, intended for experimentation (via interconnection of basic circuits on board). The contribution of this project is in providing a basic introductory overview of some of the problems (PWM output in particular) in analog I/O design and implementation for soundcards through a real world example, which - while incapable of delivering professional grade quality - could still be useful, primarily in an educational scope.

1. INTRODUCTION

In contemporary terms, a soundcard is a device that is recognized by consumer users in having a specific role: it provides an analog input/output hardware interface to high-level PC audio software (from media players like VLC to processing environments like PureData). As such, the development of a soundcard can be seen as a cross-disciplinary effort, requiring understanding of both (analog and digital) electronics, and software engineering (from OS drivers to application level). Consequently, it is (for the most part) commercial enterprises that have the resources to develop soundcard systems, in terms of practical implementation.

However, thanks to the broader penetration of affordable technology in the mass market, as well as developments in open source software - the development of practical soundcard systems could be also within the reach of the DIY developer. This paper can be seen in the context of a wider project aiming to broaden the discussion on open, DIY

soundcard designs (as examples of PC controlled digital audio hardware): [2] discusses an obsolete, historic hardware design (with discrete parts) controlled by simple software - while the AUDIOARDUINO [1] project demonstrates a working, open soundcard system, which is based on: an FTDI based ARDUINO DUEMILLANOVE as hardware; specific microcontroller code for ARDUINO's ATMEGA328; and matching ALSA soundcard driver for Linux. Note that the main discussion of the streaming capability, software and microcontroller issues of AUDIOARDUINO is given in [1]; this paper can be seen as an extension, focusing on the issues of analog I/O.

The wider intent is to address those, that wish to start studying the interaction between digital audio hardware and software through practical examples - and as such, would represent 'novices': here understood as people that may have basic understanding of analog (e.g. op-amps) and digital (e.g. 74XX series) electronics, as well as software (C programming) - but not necessarily *practical* experience with application of these domains in the context of digital audio. As such, they are likely to encounter a similar situation, as some of the developers on this project: while there may be ample literature on best practices (also from industrial perspective) in both digital and analog audio, often times the discussion is in reference to simulation results or industry equipment; and it can be difficult to parse for a novice, without a previous practical insight. Then again, the practical insight can be difficult to gain, assuming the easiest (and most obvious) access to parts for such novices is an average electronics lab (typically offering 'classic' through-hole, discrete parts).

The board in this project follows that naive approach: the intent is not to provide a design competitive or comparable to commercial products; rather, it is to start discussing issues in analog I/O, from the perspective of AUDIOARDUINO, through a board that could relatively easily be assembled by novices. AUDIOARDUINO takes the CD quality (16-bit, 44.1kHz - supported by many soundcards) as a reference, and then attempts to achieve it, insofar as the hardware and the simplicity of the approach allow. Eventually as an 8-bit, mono device, AUDIOARDUINO allows for simpler conceptualization of the the 'analog sample' (as a building block of the digital audio stream) in the software domain; this board, then, aims to complement that approach, and simply allow for the 'analog sample' to be more easily traced in both analog and digital domains - in the same context as real soundcards. Furthermore, there are commercial offerings that use the ARDUINO for sound, such as WAVESHIELD, VOICEBOX SHIELD, SEEBESTUDIO ARDUINO MUSIC PLAYER SHIELD or RMP3, which

provide better fidelity than AUDIOARDUINO - however, they represent standalone audio players/recorders; while AUDIOARDUINO represents a soundcard. As such, while the study of these devices may hold the key for a quality analog I/O for AUDIOARDUINO, the intent here is more on documenting how basic elements behave in simple configurations - as opposed to obtaining performance. This is also the reason why established techniques like data compression, or companding (A-law or μ -law), are not addressed; while many other technical details are included for reference. In this way, the paper could serve as introductory material, especially to people with untraditional electronics engineering background - in particular the wider electronic music instrument community.¹ And while the provided technical details may be difficult to structure in a more meaningful way for non-specialists, they represent a kind of practical experience which, the authors believe, makes the difference between a theoretical concept and a practical exercise - and as such, would be of interest to novices.

Eventually, as this paper concludes, the use of this board does not necessarily deliver any advantage - as opposed to using the 'raw' analog I/O of the ARDUINO. Yet, documenting its development and performance, through this paper and media (schematics, video) on associated webpage [3], could be of use as a starting point to novices - even without building the board. Thus, the contribution of this paper is primarily educational - however, it could possibly lead to the development of an analog I/O board for AUDIOARDUINO, that would be close to matching the contemporary state-of-the-art.

2. PREMISE

In simple terms, the problem here can be stated as follows: while AUDIOARDUINO can demonstrate a soundcard operation, it does so by reading a 0-5V analog input, and providing a 0-5V PWM signal as analog output; however, soundcards typically feature line (line-in and line-out) connectors, as well as microphone input and speaker output - all of which operate with analog signals, with a different format from those that can be obtained directly from the ARDUINO. The question is then, what basic circuitry could we use to address the conversion between the analog interface already present in AUDIOARDUINO, and the typical analog interface found on a soundcard; and what could be expected from a practical implementation of the same.

As discussed in [1], an ARDUINO is *sufficient* to demonstrate the operation of a soundcard - in particular, because on-board facilities of ARDUINO'S ATMEGA328 are used to implement analog input/output (I/O). In a soundcard architecture sense, the ARDUINO board can be said to implement the functionality of both the digital *bus* interface, and the analog-to-digital (ADC) and digital-to-analog (DAC) conversion.

In terms of *analog input*, the ATMEGA328 contains a single 10-bit successive approximation ADC unit, which is

¹ which, partly thanks to platforms like the ARDUINO, may have experienced increased exposure to basic analog electronics issues - also for its members with primary expertise in other fields

in turn connected to an 8-channel *multiplexer* [4, p.251]. The use of an ARDUINO for AD conversion of diverse sensor signals is standard practice [5], where the usual user expectation is to obtain values in the 10-bit range (from 0 to 1023) - as representation of a voltage signal, in the range from 0 to 5V, brought to an analog input. Lacking any input filters, signals containing a constant (DC)² component can be sampled without a problem (in contrast to the main issue in [2]).

In terms of *analog output*, the ATMEGA328 offers three programmable 'Timer/Counters': Timer1 (16-bit), Timer0 and Timer2 (8-bit) [4]. Each of these can count monotonically up or down (which corresponds to a saw or triangular 'digital' signal in time domain); and have associated 'Output Compare' (OC) units, registers and pins - which allow for comparison of the current counter value with a preset value. The result of this compare operation, can be output on the respective OC pin as high (5V) or low (0V) voltage level. Thus, the OC units can be used as binary (i.e., 'square') signal generators - where the period of the signal is determined by: the clock frequency, maximum and minimum value of the counter (called 'top' and 'bottom' in [4]), the mode and direction of counting, and the result of the current compare operation. In AUDIOARDUINO, the analog representation of the incoming 8-bit/44100 Hz data stream is a binary PWM signal with a frequency of 62500 Hz - where the *duty cycle* of the PWM signal corresponds to the analog sample value.

3. ANALOG I/O AUDIO LEVEL STANDARDS

When discussing analog signal interfaces for audio, related literature often mentions three categories: [6] mentions three levels used in a recording chain - microphone level, line level and speaker level; while [7] talks about low-level analog signals, line-level signals and amplified signals. In principle, these categories would simply represent increasing signal levels - thus having a definition for 'line level' audio, would also partially specify the other two domains.

However, it is not easy to find a single definition of what line level audio is. The term 'line-level' itself may originate from early use in telephony [8], referring to pre-amplified microphone signals [9]; but its possibly easiest definition is as both 'the output level of a preamplifier', and 'the input level of a power amplifier' [8]. Furthermore, there could be differences between professional and consumer grade audio hardware: [6] mentions '+4 dBu, which is 1.23V' for professional, and '-10 dBV, which is 0.316 V' for consumer applications; [8] mentions '*any level above 25 mV RMS*' for consumer, and '*0 VU*' as reference for commercial applications - where 0 VU could be: 0.775 V RMS; 1.23 V RMS (+4 dBm); or 1.95 V RMS (+8 dBm). Also, [10] defines 0 dB for line-level via maximum amplitude of ± 0.7 V; while [11] reports ± 1.5 V as amplitude for line-level output of an IPOD.

Given how varying definitions for line-level signal can be, one could wonder whether there is a standard that could

² Note that while DC stands for 'direct current', it is often used, especially in power supply adapters, to describe a constant voltage signal (as in "12Vdc") - and a constant signal in general

be consulted. Bohn's article [12] is solely dedicated to the complexity involved in pursuing audio standards: the complexity arises from multiple organizations (not all of them solely dedicated to audio) issuing standards; these organizations merging and changing character through history, often results with the same set of standards issued under different names; the for-profit characters of standards committees allows for thousands of dollars in cost for complete sets of standards. Also, since standards are copyrighted, it is unclear to what extent they can be legally cited in an open-source project. That being said, 'line-level' signals are most likely defined either in IEC 60268³ "Sound System Equipment" (via [12]), or in EIA-RS-160³ "Sound Systems" (via [8] and search, see also [13]). However, for purposes of this document, we will consider *microphone* level signals to be below 25 mV RMS (35 mV peak) - and *line* level signals to be above microphone levels, and below 1.23 V RMS (1.74 V peak); the peak being expressed in terms of a sinusoid (according to $V_P = V_{RMS} \cdot \sqrt{2}$).

In AUDIOARDUINO, in terms of *input*, we can essentially abstract the ADC process, as we bring a given analog signal directly to a pin of ARDUINO's ATMEGA328. As such, conforming to line level on the input side will require nothing more than simple scaling⁴ in the analog domain. However, in terms of *output*, what we do obtain as a *final* analog signal representation is a PWM signal. Thus, if we want to conform to line levels on the output, we must first bring this PWM back to a format, where audio information is encoded as a voltage level - i.e., we need to perform a sort of a PWM to analog conversion (Sec. 5.1) - before we can apply filtering (as appropriate for DA conversion) and scaling⁴ (for line level conformance). Such conversion requires an overview of the PWM signal characteristics.

4. PWM AS ANALOG SIGNAL REPRESENTATION

Analog to digital conversion is often introduced through the concept of a *sampled* (discretized in time) signal: an analog signal $V(t)$ (a, Fig. 1) is periodically sampled with a frequency f_S ; the values of $V(t)$ at the moments of sampling, are taken as representation of the signal (are *held*) for the duration of the entire sampling period $T_S = 1/f_S$ (b, Fig. 1). The useful information carried by a sampled analog signal is encoded as an analog voltage level. The sampled values can further be *quantized* (discretized in amplitude) to a finite set of levels, separated by a step value. This allows for finite enumeration of the quantized level set (digitizing), and further encoding with e.g. a binary code.⁵ As such, there is a direct correspondence between analog sampled-and-held (SH) and *pulse width modulation* (PWM) [14] signals, which will be briefly outlined here

³ IEC: International Electrotechnical Commission; IHF: Institute of High Fidelity; EIA: Electronics Industries Association (now Alliance); RS: Recommended Standard

⁴ In this paper, *amplification/attenuation* refers to multiplication, i.e. $y(t) = a \cdot x(t)$; and *scaling* refers to multiplication and constant addition, i.e. full linear transform $y(t) = a \cdot x(t) + b$

⁵ Note that many resources may often refer to visualizations of sampled, quantized analog signals as "PCM"; however, pulse-code modulation (PCM) is a binary encoding technique, which can have the return-to-zero (RZ) or non-return-to-zero (NRZ) waveforms.

through Fig. 1.

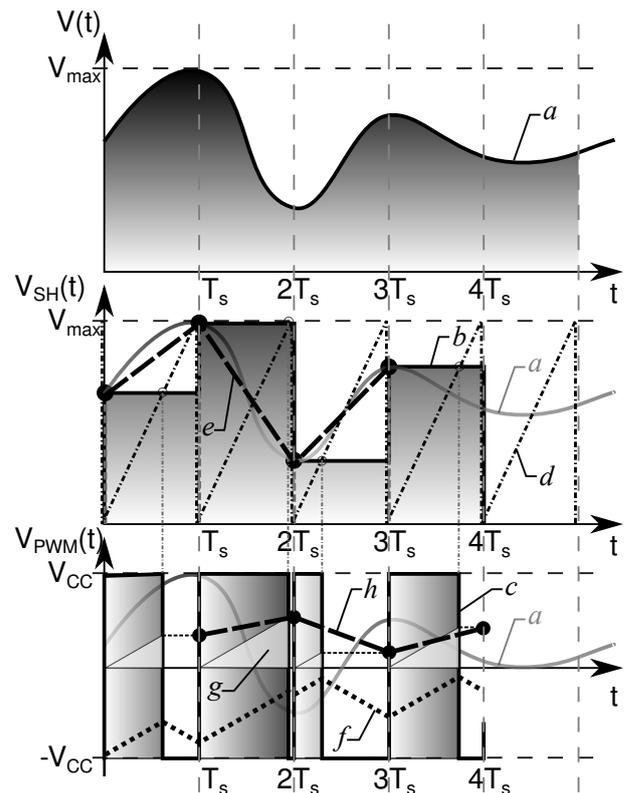


Figure 1. Comparison between analog signal (top, a); its sample-and-hold (middle, b) and its PWM representation (bottom, c) [signals emphasized with gradient filled areas, see text for the other markings].

To begin with, $V_{SH}(t)$ (b, Fig. 1) can be produced by running $V(t)$ (a) through a *sample-and-hold* circuit, clocked at frequency f_S ; while $V_{PWM}(t)$ (c) can be produced by running $V(t)$ through a *comparator* circuit, which compares $V(t)$ with a ramp (sawtooth or triangular) signal (d) of frequency f_S . The SH signal encodes the sampled voltage value as voltage (and the sample representation is in the same domain as the original value); while the PWM signal encodes the same voltage value as the duration of time in which the PWM signal has the high value (V_{CC}), known as *duty cycle* (thus the sample representation and the original value are not in the same domain). Note that these processes simply quantize the analog signal in time; obtaining a binary digital representation requires additional stages. Most AD converters work with voltage as input, enforcing a given sampling resolution, and can thus be directly applied to a SH signal - while obtaining a binary digital value from a PWM signal essentially requires resampling it with a frequency N times higher than the sampling frequency f_S , where N is the number of quantization levels;⁶ and then counting the number of times the PWM signal has been high within a period.

Using a sawtooth signal (d, Fig. 1) for the PWM comparison results with single-edge PWM - and in terms of [15], the PWM signal (c) on Fig. 1 is a single-edge, or specifi-

⁶ e.g., for $n = 8$ bit values, there are $N = 2^n = 256$ quantization levels, so resampling must occur at frequency $256 \cdot f_S$

cally *uniform-sampling trailing-edge* PWM signal; the same kind which is generated by ATMEGA328's Timer0 in 'Fast PWM' mode. For this kind of PWM in particular, we can easily establish a correspondence to SH: on Fig. 1, the actual signals in each case have the 'area under the curve' filled with a gradient. For $V_{SH}(t)$ on Fig. 1, the dots indicate the sampled values and the moments of sampling of the original analog signal - the thick dashed line connecting them (*e*) shows a *linear interpolation* reconstruction of the original signal, based on these sampled values. Simultaneously, the $V_{SH}(t)$ diagram shows a sawtooth signal (*d*), which explains how the particular PWM signal below is derived: first, the sample value for PWM is the same as the one for SH - because the saw period is the same as the SH sampling period, and the sample value is taken at the beginning of the period. Thereafter, while this sampled value (same as the SH signal level) is higher than the current sawtooth value, $+V_{CC}$ is output as PWM signal value; as soon as the sawtooth signal becomes higher than the sampled value, $-V_{CC}$ is output - hence, there is a linear correspondence between the PWM duty cycle and SH level,⁷ in representing a single sample value. This is also visualized on Fig. 1: the grey triangles (*g*) within $V_{PWM}(t)$ represent the result of integration of $V_{PWM}(t)$ *only* while it is in the duty cycle (active or high). If the integrated value at end of each duty cycle is translated at the end of the PWM period and taken to be the sample value (indicated by dots on $V_{PWM}(t)$), then the linear interpolation between these points (indicated again by a dashed line) (*h*) will be a scaled version of the linear interpolation of the SH signal (*e*).

In terms of spectrum, we can approximate the SH signal to a multiplication of the original modulating signal with a comb (infinite Dirac pulse sequence) signal with frequency f_S - a basic result in sampling theory is that this corresponds to convolution of the original and the comb spectra, which results with sideband images (of the baseband spectrum) around the harmonics of f_S that extend to infinity. To reconstruct a baseband signal with a spectrum limited by frequency f_{max} , Nyquist-Shannon's sampling theorem $f_S \geq 2f_{max}$ must be satisfied. A PWM signal can be approximated to a square one [16], and thus to a series of odd harmonics at $f_S, 3f_S, 5f_S, \dots$; it can be shown that the PWM spectrum will contain: the original modulating signal (baseband); the harmonics; and sidebands around the harmonics (note, however, the PWM spectrum in reality is more complex [17]). Thus, Nyquist-Shannon's theorem should be applicable to PWM as well - meaning that it should be, in principle, possible to reconstruct an analog representation of a PWM signal just by using a low-pass filter. In fact, simple LPF has been used for PWM reconstruction at least since 1937 (Reeves patent [18]); see [19] for a practical note on using simple RC filters.

Furthermore, use of PWM is common in audio amplifiers, known as 'Class-D'; often, PWM is used directly [20] to drive a loudspeaker, counting on the speaker's filtering properties; an approach already used for raw demon-

stration of AUDIOARDUINO in [1]. Given that a typical loudspeaker⁸ uses a coil to electromagnetically move a diaphragm membrane, electrically it can be approximated to a coil (inductor). As the magnetic field that moves the diaphragm is caused by *current*, the speaker can be considered (electrically) to be a current-driven element. Note however, that while $V_{PWM}(t)$ on Fig. 1 pulses between $-V_{CC}$ and $+V_{CC}$ - the PWM signal generated by the ARDUINO pulses between 0V and 5V. This, in turn, means that the current generated from such a PWM source will be strictly *unidirectional*⁹ - causing the diaphragm to move in one direction only (which is audibly weaker in comparison to a diaphragm driven both ways). This problem - generating bidirectional current in absence of negative voltage supplies - is often addressed electronically with H-bridge circuits (mentioned also in Sec. 5.2).

5. BOARD DESIGN / IMPLEMENTATION

The design approach for the board was to allow for quick testing of several approaches to analog I/O with AUDIOARDUINO; hence the board consists of several modules, charted on Fig. 2.

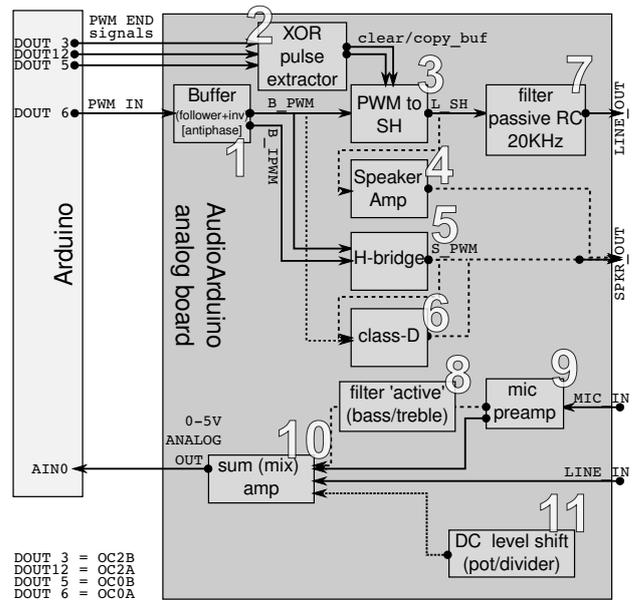


Figure 2. Block diagram, representing units of the AUDIOARDUINO analog I/O board.

Fig. 2 shows that the PWM to SH conversion is performed by a buffer (1), XOR pulse extractor (2) and PWM to SH circuit (3); further discussed in Sect. 5.1. There are three types of amplifiers: an 'analog' speaker amplifier (4), and PWM-oriented H-bridge (5) and Class-D (6) amps; further discussed in Sect. 5.2. There are two filters; passive RC (7), and active 'bass/treble' filter (8); and for handling input, there is a mic preamp (9), mixer or summing amplifier (10) and a DC level shifter (11); further discussed in Sect. 5.3.

⁸ including most headphones; however, excluding piezoelectric and electrostatic speakers

⁹ i.e., the current will flow in one direction during the duty cycle; and outside of it, current will not flow at all

⁷ as percent of time when the signal is high vs. percent of the voltage level in respect to the range represented by V_{max}

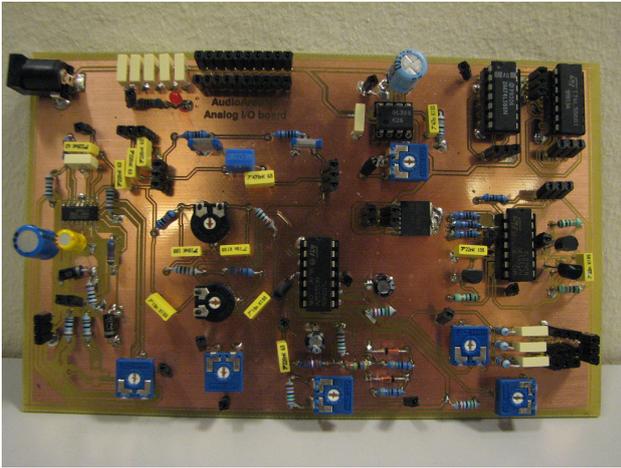


Figure 3. Photo of a finished AUDIOARDUINO analog I/O board.

The board schematics and PCB layout files have been implemented in the open-source `kicad` software, and have been released on [3]. The board has a single-sided design, implemented on a UV photosensitive PCB, which hosts both surface-mounted parts, and through hole ones (like resistors) soldered on the surface. Each of the aforementioned units is essentially a standalone module, with only some connections (like power) implemented on the board; for establishing connections, multi-pin single-row IDC socket pins are split in single connectors and soldered, in which wire can directly be inserted (to allow for simple manual wire-wrapping). Thus, the connections on Fig. 2 are not fixed: the full lines simply represent a starting configuration, and the dashed lines represent alternative ones; the completed board (without wires) is shown on Fig. 3. The board also has an adapter socket to accept DC power supply, which is distributed to most (but not all) parts of the board – and audio connectors can be added, by attaching them to respective pin sockets.

5.1 PWM to analog (SH) conversion

As noted previously, low-pass filtering is standard practice for reconstructing a PWM signal in the analog domain; the terms usually applied are PWM "reconstruction", "filtering" or "demodulation". However, a technique known as "ramp and hold" can be considered a PWM to sampled, quantized analog (or "PWM to SH", for sample-and-hold) signal converter: the quantization of a microcontroller-generated PWM duty cycle will be effectuated as quantized analog voltage levels; and due to a dependence on a clocked, regular 'clearing', quantization in time is inherent. This section describes the reasoning behind a simple "ramp and hold" implementation with discrete parts, utilized here as PWM to SH converter.

To discuss the PWM to SH conversion, note first that PWM audio power amplifiers typically work with switching (PWM) frequency between 200 kHz and 800 kHz [16] – even if Sec. 4 implied that 44.1 kHz (which, by Nyquist, cover the audible 22.05 kHz analog spectrum) is applicable as lower bound for PWM frequency. Then, let's re-

turn to AUDIOARDUINO's method of generating PWM on the ATMEGA328. The 'Timer0' timer/counter is used for this purpose: once set, it runs continuously at the specified timer clock frequency (in 'parallel' with the main code execution). The achievable frequency of the waveform it can generate is: $f_{PWM} = f_{clk_io}/256N$ for Fast PWM mode [4, p.103], or $f_{PCPWM} = f_{clk_io}/510N$ for Phase Correct PWM mode [4, p.104]; where N is the 'prescale factor' (1, 8, 64, 256, or 1024). Thus, the highest possible PWM frequency on ATMEGA328 with 16 MHz clock is $16000000/256 = 62500$ Hz; achieved in Fast PWM mode with prescaler $N = 1$.

A PWM signal with frequency of 62.5 kHz should be able to reproduce the content of a 44.1 kHz digital stream in the analog domain, given that the sample sizes are the same (here 8-bit). The next possible waveform frequency (for $N = 2$) is 31.25 kHz, which - being lower than 44.1 kHz - is not suitable for reproduction; and that is the case for all other PWM frequencies achievable on this ARDUINO. Here, let's note that the Timer0 counter simply increases the value of the register (variable) TCNT0 at each clock tick; since TCNT0 is 8-bit wide, when it reaches 255 it 'overflows' on the next tick (that is, it is reset to 0) - and this is what sets the PWM period. A matching register, OCR0A, is used to set the duty cycle - it is continuously (at each clock tick) compared to TCNT0: and if it is bigger, the matching pin OC0A is set to high voltage (V_{cc}); else it is set to 0 (ground).

We can now identify some sources of error in this arrangement. The microcontroller code writes a single sample to PWM (that is, OCR0A) at 44.1 kHz; the PWM runs independent of that at 62.5 kHz. At the moment of writing, the counter may still process the previous (analog sample) value - and the new value will be output first at the beginning of the next PWM period. This could be also seen as (analog) SH samples being displaced from their default positions, and as such could be considered 'analog' jitter of sorts (see [21] for jitter measurements in professional equipment). Additionally, note that "the extreme values for the OCR0A Register represent special cases [4]" - meaning that reproduction errors could be experienced for values 0 and 255, the limits of the 8-bit range. Assuming that these errors will be tolerable, the problem now is how to extract a SH type of voltage from the PWM signal, to conform with 'line-level' format.

Now, let's briefly return to the case of a loudspeaker, which we can discuss as an inductor with inductance L . The current through an inductor L is the integral of the voltage across: $i(t) = 1/L \int_0^t v(t)dt$ - ideally,¹⁰ constant voltage would result in linear ramp current. Thus, if an inductor is driven by a *voltage* signal $V_{PWM}(t)$ as on Fig. 1 - then the *current* through it (shown as the dotted line on the PWM signal, f on Fig. 1) will be: an upward ramp during the duty cycle (for $+V_{CC}$); and downward ramp outside of it (when the voltage is $-V_{CC}$). Ideally, since $abs(-V_{CC}) = +V_{CC}$, the upward and downward ramp of the current would

¹⁰ A problem is that we cannot really approximate a speaker to an ideal inductor; taking resistances into account, we now discuss RL circuits - which instead of a linear ramp, will produce an *exponential* current (which further increases the reproduction error; see [16] for the same problem, but in terms of coding carrier linearity).

have the same slope (30° on f , Fig. 1) - however, note that even in this ideal case, the current signal thus obtained does *not* have a shape that follows the shape of the SH voltage interpolation (the dashed line e on Fig. 1) we are interested in. While this can be addressed by increasing the PWM frequency,¹¹ we would still have inaccurate sample reproduction in the audio domain (for the particular type of PWM signal on Fig. 1), even in the case of an ideal inductor (speaker).

However, if we perform integration *only* during the duty cycle, the integrated values at end of *each* PWM period will correspond to the SH values. And this is ideally what happens with the unipolar ($0/+V_{CC}$ V) PWM generated by ARDUINO fed to a speaker: integration is performed during duty cycle; and off duty cycle, there is no current and thus no ramp in the other direction (although current should leak). We can apply the same thinking to obtain integrated signal in the voltage domain, by replacing the inductive speaker with an *integrator* circuit. The basic integrator typically charges a capacitor (CInt0 on Fig. 5) as a way to obtain an integrated signal. Thus, to obtain integration *per each* PWM period, we must discharge the capacitor at end of each PWM period - which means we must somehow detect the beginning and end of the PWM period.

Technologies like clock recovery or phase-locked loop circuits are usually needed to extract a signal describing the PWM period from an unspecified PWM signal. Here, we instead use remaining timers on the ARDUINO to generate these signals: we can set the other 8-bit timer/counter Timer2 to run in sync with Timer0 (causing counter values TCNT0 == TCNT2 at all times) - and set the remaining OC pins (OC2A, OC2B, OC0B) to turn high when counter reaches values 252, 253 and 254, respectively¹² - thus indicating the end of PWM period. These signals all turn zero at the start of the PWM period, when the counter is 0 - meaning that they overlap in time (Fig. 4 bottom left). To extract mutually *exclusive* pulses indicating counter value 252, 253 and 254 (Fig. 4 bottom right), we can employ the XOR extraction circuit, shown on bottom part of Fig. 4.

For electronic buffering of the PWM signal, XOR circuits (from a 74LS86 chip) are used, because they can be configured (in terms of binary TTL level signals) to work as either *follower* or *inverter* buffers.¹³ The pulse extractor XOR circuit (Fig. 4, bottom) accepts the end-of-PWM-period OC pins' signals, and outputs buffered end-of-period pulses (xBEOP) corresponding to 4, 3 and 2 counter ticks (counts 252, 253 and 254) before start of next PWM period. These end-of-PWM-period pulses can then be applied as COPY and CLEAR (respectively, xBEOP-3 and xBEOP-2 on Fig. 4) of the PWM-to-SH circuit, whose schematic is shown on Fig. 5. The PWM-to-SH converter on Fig. 5 consists of: a basic *integrator* (resistors RInt1-

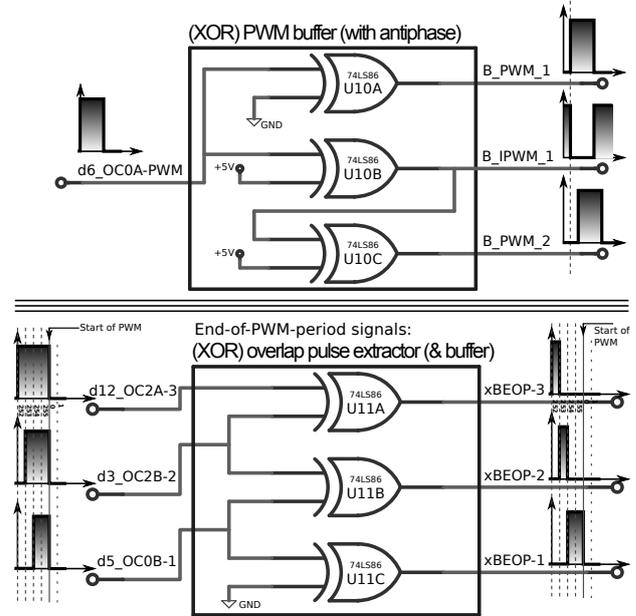


Figure 4. Schematics of PWM buffer (top) and PWM end-of-period pulse extractor (bottom), implemented with XOR gates.

4=RInt, opamp (A) and capacitor CInt0); structure for discharging the capacitor (transistor QInt_Empty0 and resistor RbEmpty0); a structure known as *transmission gate* or *pass transistor* which behaves as an analog switch (transistors Q_Pass1,2 and resistors RbPass1,2); a 'copy buffer' capacitor C_cpbuf_0 (which together with the analog switch forms a 'Sample and Hold' circuit); and intermediate (B) and output (C) buffer followers.

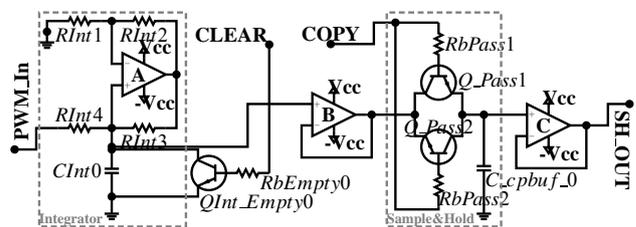


Figure 5. Schematics of the PWM to SH converter.

The integrator used here is a *non-inverting* integrator (see [22]),¹⁴ so as to preserve the phase of the integrated signal as on Fig. 1. As in most RC structures, this circuit too will generate exponentially changing voltage as the result of the processing of a constant voltage input; however, if the RC product is much smaller than the PWM period, the exponential voltage change can be approximated to a linear ramp. The transmission gate structure is based on the assumption that its transistors are either off or saturating, making the structure appear as either high or low resistance. This approximation to an analog switch will be better for transistors with higher β (h_{FE} , forward current gain) parameter, and very low cut-off currents. While FET transistors would be more appropriate for this role - BJT

¹⁴ Note that the simplest opamp integrator circuits typically represent inverting integrators

¹¹ Note that (unlike Fig. 1) literature may often show a single sinusoid period, sampled by eight [14] up to tens of PWM periods [16]; higher PWM frequency forces the integrated signal to more closely resemble the original one

¹² Using values 253, 254 and 255 turns out to be problematic, due to the special status of 255 as range boundary

¹³ Note that the LS (Low Power Schottky) family of 74XX TTL series has a standard propagation delay of 10 ns; thus inverting a signal twice, results with a pulse which is delayed (in respect to a pulse buffered by a follower - which is shown on top part of Fig. 4).

transistors were used here, simply because they may be conceptually more accessible to novices,¹⁵ and it would be instructive to observe their behavior in a circuit, based on just the previously stated assumptions. For that reason, the transistors on the actual board are simply those with the highest β , available to the project at the time (in this case, BC337-25); this is also the reason why a transmission gate is implemented through discrete components, instead of using an integrated bilateral switch (such as 4016 or 4066).

The principle of the Fig. 5 circuit operation is: PWM input is brought to the integrator; during duty cycle, the +5V of PWM input cause a constant current to charge CInt0, which develops a linear ramp voltage $V_{C_{\text{Int0}}}(t)$ (as integral of the constant current). This voltage $V_{C_{\text{Int0}}}(t)$ is buffered through (B) and brought to input of the analog switch structure. When COPY is high, the analog switch turns ON, which should allow (B) to quickly charge C_cpbuf_0 to the same voltage held by CInt0 (i.e., $V_{C_{\text{Int0}}}(t)$). When COPY is low, the analog switch is turned off - so C_cpbuf_0 can *not* discharge, being buffered by (C); and should hold its voltage constant. On the other hand, when CLEAR is high, QInt_Empty0 turns on, and short-circuits CInt0 - forcing its voltage to 0. Thus, the following sequence of events can be pursued: at start of PWM period, CInt0 is charged for as long as the duty cycle is active, and keeps this value off duty cycle; near end of PWM period, COPY is triggered, and C_cpbuf_0 is set to same voltage as CInt0; then COPY is deactivated, which isolates C_cpbuf_0 from CInt0, causing it to hold the last copied value; then CLEAR activates, which discharges CInt0; then finally CLEAR turns off, just in time before next PWM period starts - where integration can start again, by charging a newly emptied CInt0 capacitor.

Applying xBEOP-3 and xBEOP-2 (Fig. 4) to COPY and CLEAR respectively, would ensure that COPY and CLEAR run in sequence just before the end of PWM period. Ultimately, this should result with the SH_OUT signal which is a 62.5 kHz analog SH voltage representation of the 62.5 kHz PWM signal, in the voltage range from 0V to +5V. In AUDIOARDUINO context, SH_OUT would be an oversampled (but jittered - for more, see [1]) reproduction of the original 44.1 kHz audio stream played back by high-level software. Running this SH_OUT signal through low-pass filter, DC-blocking capacitor, and amplifier, should finally result with a audio voltage signal conforming to the *line-level* range of $\pm 1.95V$ around a zero volt reference.

Note that this specific copy/clear PWM-to-SH process guarantees errors in reproduction for levels above the COPY pulse locations (i.e. above 252). Additionally, most opamps work properly as followers (as on Fig. 5) only when fed by a symmetric power supply ($\pm V_{cc}$); powering basic opamps like TL074 with a single supply (between GND and Vcc) will introduce additional errors in operation.¹⁶ Most of the experiment videos on [3] have been performed with a sin-

gle supply of 5V.¹⁷

5.2 Speaker amp, H-bridge and Class-D

Whereas for 'line-out', we had to consider some form of a PWM to SH conversion, for a speaker output we need to ensure that the signal is properly amplified - and there are ICs that can work either with either type of signals. However, the choice can often be overwhelming for a novice, as PWM parts can often be intended for motor use; therefore the board includes several parts for comparison.

As mentioned, there are three amplifier structures on board. The 'speaker amp', is based on a NATIONAL SEMICONDUCTOR LM386¹⁸ (marketed as 'Audio Power Amplifier') and intended to drive a small loudspeaker from line-level input. The other two amplifiers are intended to handle raw PWM input: a standalone RAHM BD6211F H-bridge driver chip, marketed as 'full-bridge driver for brush motor applications' with 400 ns dead time; and class-D amplifier based on INTERNATIONAL RECTIFIER'S IRS20954S¹⁸ IC, marketed as half-bridge 'protected digital audio driver', with selectable dead time between 15 and 45 ns.

All of these parts make use of an *H-bridge* structure, where transistors are used as switches to cause bidirectional flow of current through a load from a single supply: LM386 and IRS20954S use half-bridge (while BD6211F uses full-bridge) configuration. Note that LM386 as 'class-B push-pull' amp uses BJT (whereas the others, as class-D, feature FET) transistors as bridge switches (IRS20954S also needs additional MOSFETs). Also, BD6211F needs two PWM inputs (in antiphase), which is provided by XOR buffer in Fig. 4. The H-bridge structure typically has an issue when mutually exclusive switches briefly stay turned on together, which causes a short-circuit of the bridge output (alias shoot through); this is usually handled by introducing so-called *dead-time* [16]. Note that BD6211F's dead time of 0.4 μs represents 2.5% of the 16 μs period of the 62.5 kHz PWM signal.

5.3 Analog filters and input preamplification

The filter units can be used for either input or output through wirewrapping. Assuming that the output PWM signal from AUDIOARDUINO has a baseband spectrum up to 20 kHz, and harmonics (with sidebands) starting at 62.5 kHz, an ideally steep low-pass filter with cut-off at 20 kHz would be able to reconstruct the baseband analog signal. The board provides the RC (7, Fig. 2) unit as a passive low-pass filter, designed for a cut-off frequency at 20 kHz - simply as a means for experiencing the influence of the simplest filter design on an audio PWM signal (even if, as a first order filter, it cannot be expected to achieve anything close to ideal reconstruction). The card design used as a base in [2], serves as a source of: the 'bass/treble' filter (8, Fig. 2); and the mic preamplifier (9,11 Fig. 2) unit - while the mixer (10, Fig. 2) unit is a basic inverting op-amp summer; all these units (8-11) are based on generic opamps (i.e. TL074).

¹⁵ as BJT are often used as a starting point in discussing semiconductor transistors

¹⁶ although, there exist pin-compatible alternatives, known as 'rail-to-rail' op-amp

¹⁷ Further notes about the operation and performance of the board's circuits can be found on [3].

¹⁸ Used circuit design taken from part datasheet.

6. CONCLUSIONS

This paper outlined some basic issues in analog I/O for soundcards, primarily by discussing a 'line-level' interface for the AUDIOARDUINO open soundcard system, implemented by the analog I/O board. As the line input was deemed to be manageable by scaling - the focus was mostly on introducing the role of PWM voltage as digital output signal, and it's relationship to analog voltage in context of audio; supported with a basic, first-principles proposal for a simple PWM to (discretized) analog converter utilizing specific capabilities of ATMEGA328.

Further media on [3] documents that this board, in its current form, does not bring about any advantage to the use of AUDIOARDUINO as a soundcard; it is useful only as a subject of introductory study. In particular, the media documentation on [3] aims to provide experiential familiarity to novices with the approached outlined here, even without the need to actually build the board. As such, this paper and project aim to provide a basis for further development - and in that, promote the discussion of DIY digital audio hardware implementations among researchers and hobbyists.

7. ACKNOWLEDGMENTS

The authors would like to thank the Medialogy department at Aalborg University in Copenhagen, for the support of this work as a part of a currently ongoing PhD project.

8. REFERENCES

- [1] S. Dimitrov and S. Serafin, "Audio Arduino - an ALSA (Advanced Linux Sound Architecture) audio driver for FTDI-based Arduinos," in *Proceedings of the 2011 conference on New interfaces for musical expression*, 2011.
- [2] S. Dimitrov, "Extending the soundcard for use with generic DC sensors," in *NIME++ 2010: Proceedings of the International Conference on New Instruments for Musical Expression*, 2010, pp. 303–308.
- [3] —, "AudioArduino Analog Board homepage," WWW: <http://imi.aau.dk/~sd/phd/index.php?title=AudioArduino-AnalogBoard>.
- [4] www.atmel.com, "Atmel ATmega48A/48PA/88A/88PA/168A/168PA/328/328P datasheet," WWW: http://www.atmel.com/dyn/resources/prod_documents/doc8271.pdf, Accessed: 29 Dec, 2010.
- [5] arduino.cc, "Arduino homepage," <http://www.arduino.cc/>.
- [6] T. Amyes, *Audio post-production in video and film*. Focal Pr, 1998.
- [7] L. Ahlzen and C. Song, *The Sound Blaster Live! Book: A Complete Guide to the World's Most Popular Sound Card*. No Starch Pr, 2003.
- [8] G. White and G. Louie, *The audio dictionary*. Univ of Washington Pr, 2005.
- [9] R. Donald and T. Spann, *Fundamentals of television production*. Wiley-Blackwell, 2000.
- [10] D. Walters, *How to Build a Radio Station*. Lulu. Com, 2006.
- [11] mitat.tuu.fi, "Line level audio signal voltage," WWW: <http://mitat.tuu.fi/?p=45>, Accessed: 5 January, 2011.
- [12] D. A. Bohn, "The bewildering wilderness—navigating the complicated and frustrating world of audio standards," *S&VC*, September, vol. 2000, pp. 56–64, 2000, URL: RANE reference,<http://www.rane.com/pdf/bewilder.pdf>.
- [13] J. A. Caffiaux, "A brief review of eia standards in the audio field," *J. Audio Eng. Soc.*, vol. 16, no. 1, pp. 21–25, 1968.
- [14] K. Nielsen, "A review and comparison of pulse width modulation (pwm) methods for analog and digital input switching power amplifiers," in *PREPRINTS. AUDIO ENGINEERING SOCIETY*, 1997, p. 57pp, URL: <http://www.icepower.bang-olufsen.com/files/convention/4446.pdf>.
- [15] Z. Song and D. Sarwate, "The frequency spectrum of pulse width modulated signals," *Signal Processing*, vol. 83, no. 10, pp. 2227–2258, 2003.
- [16] A. Knott, "Improvement of out-of-band behaviour in switch-mode amplifiers and power supplies by their modulation topology," Ph.D. dissertation, Technical University of Denmark, Department of Electrical Engineering, Electronics, 2010. [Online]. Available: <http://orbit.dtu.dk/getResource?recordId=270897&objectId=1&versionId=1>
- [17] A. Knott, G. Pfaffinger, and M. A. Andersen, "On the Myth of Pulse Width Modulated Spectrum in Theory and Practice," in *Audio Engineering Society Convention 126*, 2009.
- [18] W. Kester, Analog Devices, Inc. *et al.*, *Data conversion handbook*. Newnes, 2005.
- [19] A. Palacherla, "Using PWM to Generate Analog Output," 1997, Microchip Technology, (AN538).
- [20] F. T. Agerkvist and L. M. Fenger, "Subjective test of class d amplifiers without output filter," in *117th Audio Engineering Society Convention*, 2004.
- [21] J. Dunn, "Jitter: Specification and assessment in digital audio equipment," in *Presented at AES 93rd Convention*. Citeseer, 1992.
- [22] J. W. Marshall Leach, "Ideal operational amplifier (op amp) circuits," 2010, ECE3050 Analog Electronics Class notes, Georgia Institute of Technology. WWW: <http://users.ece.gatech.edu/mleach/ece3050/sp04/OpAmps01.pdf>.

DESIGNING AN EXPRESSIVE VIRTUAL PERCUSSION INSTRUMENT

Brian Dolhansky
Drexel University
bdol@drexel.edu

Andrew McPherson
Drexel University
apm@drexel.edu

Youngmoo E. Kim
Drexel University
ykim@drexel.edu

ABSTRACT

One advantage of modern smart phones is their ability to run complex applications such as instrument simulators. Most available percussion applications use a trigger-type implementation to detect when a user has made a gesture corresponding to a drum hit, which limits the expressiveness of the instrument. This paper presents an alternative method for detecting drum gestures and producing a latency-reduced output sound. Multiple features related to the shape of the percussive stroke are also extracted. These features are used in a variety of physically-inspired and novel sound mappings. The combination of these components provides an expressive percussion experience for the user.

1. INTRODUCTION

With the widespread use of mobile devices by the general public, musical applications and games have been particularly popular. On the Apple App Store there are several percussion applications that attempt to give the user the ability to play a full or partial drum set on their phone. However, the degree of expressivity granted by these percussion applications is limited by simple trigger-type input methods. Playing a stored drum sample via a trigger does not allow the user to modify the qualitative aspects of a single drum hit.

While, to some, percussion instruments appear to lack expression, in reality a drummer is able to change their playing style by not only modifying their drumming rhythm, but also by adjusting parameters such as the energy they impart to each drum hit, the stick type used and the location of the hit on the drum.

Unlike the aforementioned drum applications, we present an implementation that replicates some of these expressive qualities by extracting features from an accelerometer profile generated when a user moves a mobile device like a physical drum stick. Several percussive gesture features regarding the timing and shape of the preparation and stroke are extracted from the accelerometer signal and mapped to an output sound. A predictive triggering method substantially reduces the latency between gesture and output



Figure 1: The virtual percussion system presented in this paper aims to provide a system that closely replicates an actual instrument. It was inspired by analyzing actual drum strokes.

sound. This system provides a musical experience that better replicates playing a physical instrument.

2. PREVIOUS WORK

Several studies have focused on using accelerometers and gesture recognition in an attempt to replicate the motions used when playing real physical instruments. Dahl [1] described the correlation between stroke shape and strike velocity through user studies. Hajian et al. [2] studied human drumming styles by recording the accelerometer profiles of actual drum strikes. Tindale et al. [3] assessed various stick augmentation techniques, including using accelerometers and gyroscopes to aid in drum gesture detection. The AoBachi system by Young et al. [4] replicated Japanese taiko drumming by using accelerometers placed in the large “bachi” drum sticks. Bott et al. [5] used a standard Wii Remote to simulate playing a number of instruments, including drums, by making gestures corresponding to the typical motions involved in playing those instruments.

Cook explored expressive synthesis of percussion sounds with the Physically Informed Stochastic Event Modeling (PhISEM) algorithm [6] and detailed its use in software- [7] and hardware-based [8] applications. Accelerometers were attached to simple controllers, such as a user’s foot for a toe-tapping instrument, and players were able to shape pre-recorded drum loops. Cook argued for simple instruments that are intuitive for an inexperienced user to play.

Heise et al. [9] used a Wiimote to integrate the PhISEM model and a physical controller. This system was able to simulate various percussion instruments, including the maraca and rainstick.

There has also been research conducted on the viability of replicating drum playing on a mobile device. The Shoogle system proposed by Williamson et al. [10] used haptic feedback to notify the user of certain events. This system could be extended to provide a realistic feeling when playing a virtual drum instrument. Tanaka [11] presented a collaborative musical creation system that used mobile devices augmented with a suite of sensors so that a group of users could compose a piece of music. The ShaMus system, developed by Essl et al. [12], used phone tilt information calculated by an on-board accelerometer to control a virtual drum. The user made a strike gesture by tilting the device. Some expressive control was afforded by measuring the rate at which the device was tilted past the horizontal plane. Weinberg et al. [13] implemented the music creation tool ZooZBeat, which used gestural input as a composition tool. Accelerometer onset detection was used for note input. The energy of the onset was calculated to allow the user to change the pitch of the input note.

Our system extends the work presented in these studies by capturing percussive motions on a mobile device and mapping these gestures to sound synthesis. This system uses an intuitive input and output method, as the user simply has to move the mobile device like a drum stick. We propose the extraction of multiple independent features per swing. The output sound is produced exactly when the user expects it, and with modifications that are directly related to the extracted features.

3. OVERVIEW

To better understand the acceleration characteristics of a percussion stroke, we first recorded the movements of musicians playing real instruments using the accelerometer in an iPod Touch. For comparison, we carried out perceptual tests where users swung only the mobile device in a motion mimicking a percussive stroke. We identified several key features that affect the quality of the sound of a drum hit and designed a system to extract these features in real time on a mobile device.

The implementation of an expressive virtual percussion instrument requires several integrated components that must operate quickly enough to provide an experience that mimics playing a real instrument. The system must be simple to reduce computation time, but it must also provide enough expressive control to make playing the instrument interesting for more than several minutes.

The proposed system has three subsystems: hit prediction, feature extraction and feature mapping. Each time the device receives a new accelerometer sample, it checks to see if a hit is imminent. If so, the device examines the past motion information to determine what type of hit will occur and extract other related gestural features. The device then maps these features to an output sound.

4. DEFINING INSTRUMENT EXPRESSIVENESS

4.1 Expression of Physical Instruments

The term “expression” is difficult to define. Any musical piece can be played in different manners by changing not only what notes are played, but *how* the notes are played. An instrument’s ability to qualitatively modify notes or sounds constitutes its expressiveness.

Generally, the degree of expression afforded by an instrument depends on the physical properties of that instrument. For example, it is possible for a violin player to continuously manipulate the pitch and timbre of the notes they play, allowing for a wide variety of techniques, such as slides or vibrato.

Although most drums lack melodic expression, they are not entirely “one-dimensional” instruments. A percussion instrument (or set) has several musical degrees of freedom: *what* drum a drummer plays, *when* they play it, and *how* they play it. A drummer can modify how they play a particular percussion instrument by using different sticks or hitting different parts of the drum. Changing the stick type or hit location affects the timbre of the sound.

Implementing these qualitative aspects is especially important when designing a virtual percussion instrument, as it would give the musician using the virtual instrument nearly the same amount of control and expression as if they were playing a real instrument. In addition, it is important to intuitively provide access to these qualities so that the user immediately understands the effects their gestures have on the produced sound.

4.2 Replicating Drum Expression on a Mobile Device

In order to replicate these degrees of freedom, a mobile device can record a user’s gestures using an accelerometer. The most intuitive motion for a user is to swing the device as though they are using an actual drumstick. In order to avoid breaking immersion, the system must not only detect each stroke, but the output sound must be produced concurrently with its apparent impact. Latency must therefore be addressed to provide a consistent experience.

The accelerometer magnitude profile of a drum hit is similar even among various instruments (see Section 5). Therefore, it is possible to extract certain features from the percussive gesture to modify the output sound accordingly. For instance, if the user swings the device quickly, a louder sound should be produced, mimicking the behavior of a physical drum. Other non-physical mappings were explored, such as triggering different instrument types based on the shape of the stroke (a discrete mapping) or affecting the pitch based on the stroke time (a continuous mapping).

5. REAL PERCUSSIVE ACCELEROMETER PROFILES

A pre-study was conducted to record the accelerometer profiles of the movements required to play different percussion instruments. Accelerometer data of over 300 percussive motions spanning 8 different instruments was recorded.

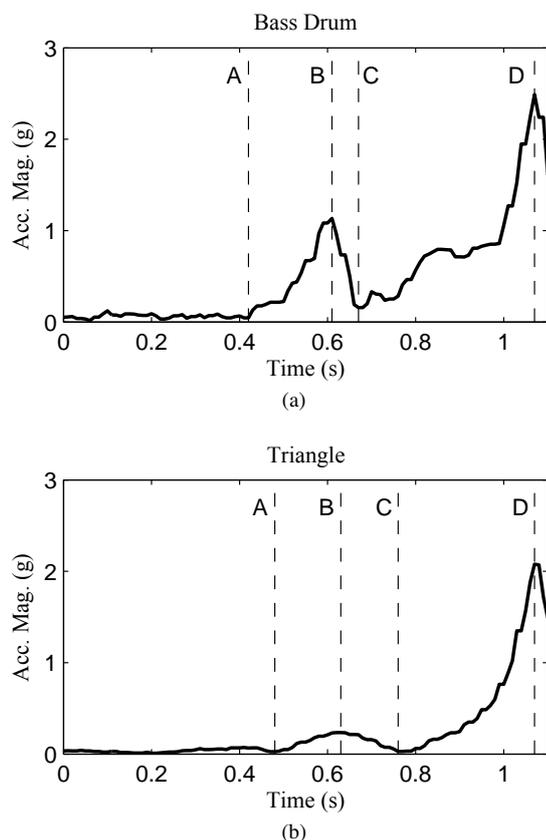


Figure 2: Accelerometer profiles for two percussion instruments (each of which requires a different playing style) played at the same velocity. The regions of the stroke are labelled, with point A corresponding to the start of the back swing, point B being the point of maximum acceleration of the back swing, point C being the midpoint of the back stroke, and D occurring at the point of impact.

The instruments included the bass drum, glockenspiel, slapstick, snare drum, tam-tam, toms, triangle and wood block. Three different users held an accelerometer-equipped mobile device (protected by a foam covering) and a drumstick in their dominant playing hand (Figure 1). This setup was used because playback occurs on an iPod device, and it is important to analyze accelerometer data from the same perspective in both recording and playback (from the base of the user’s hand, rather than the tip of the drumstick.)

The users were asked to hit each instrument 5 times at 3 different velocities. Some extra profiles were recorded, including quick successive hits, rolls and swells. All subjects had musical experience, although only one was an expert percussionist.

A generalized drum stroke was observed. A user first made a back swing away from the drum to prepare for the hit. The user then began accelerating the device in the opposite direction to strike the drum. The length of these motions depended on the instrument and the playing style.

Figure 2 shows the accelerometer magnitude profiles of two instruments that have different playing styles, specifically the bass drum and triangle. The magnitude is shown for one second preceding the drum strike. Note the distinguishable points in both examples:

- Point A: the point in time when the user begins the percussive back swing
- Point B: the peak acceleration of the first half of the back swing
- Point C: the midpoint of the back stroke, or where the user began accelerating the device towards the drum
- Point D: the point where the drum stick impacts the drum surface

Some of the feature mappings examined later were inspired by these physical trials. For instance, the bass drum generally has a longer wind up when compared to other percussion instruments. The mallet used to strike a bass drum is significantly larger than typical wooden drum sticks and the percussionist requires a longer time and distance to accelerate the mallet to the appropriate velocity. Smaller instruments such as the triangle use a smaller stick that requires less energy to move and therefore require a smaller wind up. In addition, the slope of the forward swing used to strike these smaller instruments tends to be steeper, as hitting a triangle involves a whipping motion as opposed to the grandiose swing required for a concert bass drum.

6. LOW-LATENCY HIT DETECTION

Percussion instruments are often used to keep rigid time. It is especially important to minimize or eliminate the lag between the detection of a hit and sound production, as any perceptible amount of latency diminishes both the rhythmic accuracy and the playing experience. We therefore developed a system to accurately predict a hit before its actual point of impact.

6.1 Platform Limitations

The accelerometer installed on the 4th generation iPod touch used for this implementation is software limited to 100 Hz in order to conserve battery life. This is non-ideal, especially for high frequency time-sensitive applications. In addition, the accelerometer data is very noisy, as it is mainly meant to ascertain the orientation of the device as opposed to measuring movement or displacement.

The latency inherent in the platform is partially due to the time it takes to sample the accelerometer. A larger amount is due to the architecture of iOS’s sound API, Core Audio. Empirically, the latency between when a hit was detected and when the sound was produced ranged from 10 to 20 milliseconds.

6.2 Determining the Characteristics of a Virtual Hit

It was expected that the acceleration profile of a user swinging only the mobile device would differ from when the user swung both the mobile device and a large drum mallet. It was important to not only determine the physical characteristics of a hit, as in Section 5, but the perceptual, user-defined qualities of a virtual hit. For instance, users may think that a sound should be triggered slightly before or after the actual peak in acceleration magnitude.

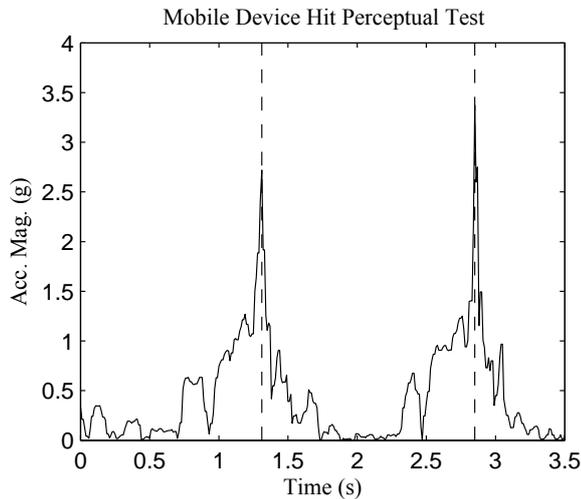


Figure 3: The locations of user triggered hits versus the acceleration profile of a percussive motion made with the mobile device.

A program was developed to record accelerometer and screen touch data. Users were asked to touch the screen when they thought a hit should occur. As can be seen in Figure 3, users correlated a peak in acceleration magnitude with when the percussion sound should be produced. The act of reversing the device direction, which mimics striking a drum, corresponds to a peak in acceleration magnitude. Accordingly, users associate peaks in acceleration with an expected sound.

6.3 Difficulties of Real-Time Hit Detection

Because sound should be produced when there is a peak in acceleration magnitude, the system should possess a robust peak-picking algorithm. However, the proposed system is causal and cannot implement a traditional peak picking algorithm that attempts to find maxima using information on both sides of the peak. In addition, even if latency were ignored, it is possible that a detected peak was a local maximum that occurred slightly before the larger hit acceleration peak.

6.4 Hit Prediction via Onset Detection

Our solution for causality and latency is a variation of onset detection that performs hit *prediction*. This solution was developed based on the characteristics of the accelerometer profiles produced by the virtual drum user tests. Any large increase in acceleration magnitude means that a hit is imminent. A person can only swing the device at a high rate for a limited amount of time before the extent of possible motion is reached.

The system should also detect hits of variable magnitude. A dynamic threshold is implemented with an envelope follower so that both hard and soft hits are recognized.

The threshold follows the leading edge of large onsets and decays from peaks with a per-sample decay rate of 3%. The decay accounts for the accelerometer peaks produced by the rebounding motion of the user after they make a hit gesture. A noise floor is also utilized in order to ignore

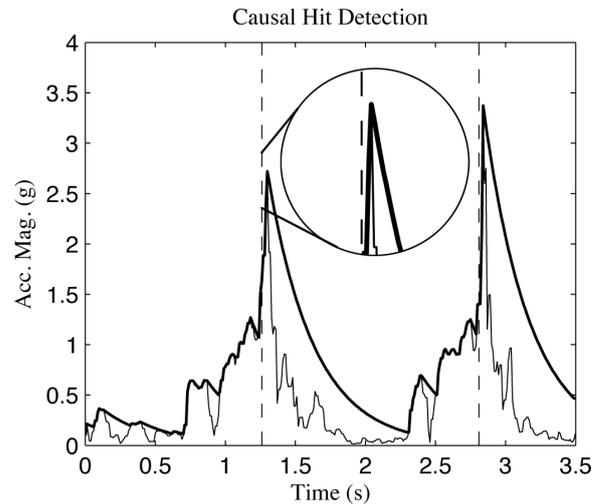


Figure 4: A demonstration of the hit prediction system. The thick line is the hit detection threshold, while the vertical dashed lines indicate where a hit prediction was made.

small fluctuations produced by the accelerometer hardware and accidental user input. This floor is set at a magnitude of 0.1 g. iOS's Core Motion API filters out the acceleration due to gravity automatically, so the gravity vector can be ignored.

If, in the current time step, the accelerometer magnitude exceeds the envelope's previous value at a high enough rate, the system determines that a hit is imminent. The slope needed for hit detection at the threshold crossing can be varied to create different sensitivities. A small rate, such as 0.25 g per sample, produces a very sensitive onset detector that can be triggered with very small movements. Increasing this rate to 0.4 g per sample requires large movements. This rate can therefore be tweaked to provide a balance between sensitivity and noise robustness.

Figure 4 shows the data recorded from user tests along with the causal hit prediction. Note that the onset detections occur several samples before the actual peak, giving our system extra computation and sound synthesis time. The mean prediction time afforded by our system was 2.09 samples (around 20 ms), i.e. our system predicted a hit 20 ms before a user expected the hit to occur. This is greater than the inherent system latency.

7. FEATURE EXTRACTION AND SOUND MAPPING

After a hit has been detected, features describing the hit must be extracted. These percussive stroke features can be extracted deterministically from the buffered accelerometer samples. Features of the future hit peak must also be estimated. Finally, these extracted features must be mapped to the output sound.

7.1 Causal Feature Extraction

When an imminent hit is detected, the system has access to past accelerometer samples. Several features describing

the stroke can be extracted from these samples. Specifically, the following features are calculated:

- Length of entire stroke
- Length of back and forward swings
- Ratio of times spent in back and forward swings
- Velocity estimate (integral of acceleration over each segment) of back and forward swings

Each of these features must be calculated automatically in a short amount of time. Our system uses the accelerometer profile generalizations observed in Section 5. We first determine points A, B and C, shown in Figure 2, and then segment the percussive stroke for feature extraction.

First, the location of the largest maximum that occurred during the first half of the back (denoted B in Figure 2) swing is calculated by using a general noise-robust peak picking algorithm.

Next, the minimum sample between point B and the current accelerometer sample is determined. This is point C and is where the user begins accelerating the device towards the drum, or where the application of force to the device changed direction. Finally, the start of the stroke (point A) is calculated by finding the first accelerometer magnitude sample that is less than 10% of the value at point B.

The length of the entire stroke, in samples, is the difference between point A and the current accelerometer sample. Similarly, the length of the back swing is related to the difference between points point A and point C. The velocity estimate of the back swing is calculated by summing the magnitudes over the segment between A and C. Because the accelerometer is noisy, calculating the velocity directly is prone to drift. In addition, there is no information about the initial velocity from the accelerometer magnitude.

7.2 Non-Causal Feature Extrapolation

The characteristics of a drum hit that determine the volume of the produced sound are mainly the velocity of the hit, the type of stick or mallet used, and the drum type. The type of stick or drum used is a programmatic choice made when designing a specific implementation of our system. However, it is important to accurately emulate the correlation between the velocity of the drum stick and the energy of the output sound, as this is one of the key techniques a drummer uses when playing a piece expressively.

Due to our use of hit prediction, we do not have immediate access to the actual peak accelerometer magnitude value. Therefore, we must use another predictive step to estimate the peak acceleration. Two causal features were used in this prediction and were motivated by the accelerometer pre-study. First, a large slope in the forward swing region (the samples immediately preceding D) corresponded to a sharper play style and louder output sounds. In addition, a large amount of forward swing acceleration corresponded to a very high strike velocity, specifically for mallet strikes on the bass drum. The features used for prediction were:

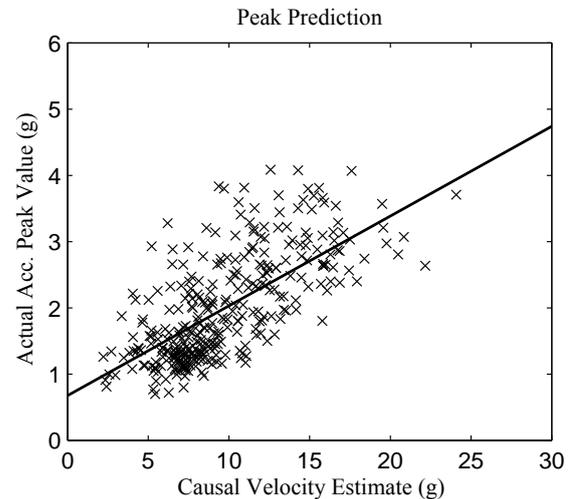


Figure 5: Visual representation of the correlation between the velocity estimate (Equation 1) and the peak acceleration.

- Slope (derivative) estimate of forward swing
- Forward swing velocity (integral) estimate

The slope estimate was determined by finding the slope between the accelerometer sample at the time of hit detection and the accelerometer value k samples prior. Different values of k were tested and $k = 10$ (or 0.1s) provided the best peak estimate. The forward swing velocity estimate was calculated via Equation 1, where $m[n]$ is the magnitude of the acceleration at time n , and i is the current sample.

$$\hat{v} = \sum_0^k m[i - k] \quad (1)$$

In order to prove the validity of applying these empirical observations, we calculated the above features for 331 specific hits and calculated the correlation between each feature and the actual accelerometer peak value Figure 5. A strong correlation between the features and the actual peak value was found, with $r = 0.74$ for the slope estimate feature and $r = 0.80$ for the velocity estimate feature.

7.3 Feature Mapping

Once the percussive stroke and peak estimate features have been calculated, they must be mapped to the output sound. Several mappings were explored, including those that used some of the physical characteristics of a drum as inspiration, and those that were based on novel (non-physical) mappings.

7.3.1 Physically-Inspired Mappings

Even for those with little or no percussion experience, it makes intuitive sense that hitting a drum harder should produce a louder noise. It is obvious that the mapping between the velocity estimate feature and the output sound should be proportional. However, the amount of energy imparted to a surface by a moving object is not directly

proportional to the moving object's velocity. A moving drum stick possesses a certain amount of kinetic energy, given by $E_k = \frac{1}{2}mv^2$. Although kinetic energy is not a direct measure of loudness, mapping the squared velocity estimate to the output sound's volume provided a natural continuum of sound.

Different drums require different playing styles. As Figure 2 shows, the concert bass drum requires the use of a large mallet and large strokes, while the triangle uses a small stick and small, quick movements. These observations can be used to select different instrument types based on the type of stroke used. A mapping was explored that switched between a triangle and bass drum output sound based on the length of the entire stroke. The type of instrument selected was independent of the final output sound volume, so a user was able to switch between instruments at will and play each with variable loudness.

7.3.2 Non-physical Mappings

Several other unique mappings were created that specifically utilized the stroke features. For this initial work, we felt it was important to use simple mappings so that it was possible for the user to discover the relationship between their gestures and the effects on the output sound. One mapping used a pitched drum output sound and changed the pitch based on the length of the stroke, with longer strokes producing lower pitches. This mapping used the strokes required to play large instruments as inspiration. Large melodic drums produce lower pitches and require large impulses.

Another mapping adjusted the frequency of a low pass filter based on the ratio between the segments A-C and C-D. This allowed a user to produce bright sounds with relatively quick forward swings, and muffled sounds with relatively slower forward swings.

These implementations are simple examples. It is possible for a developer to create many other unique percussive mappings using this system.

8. FUTURE WORK

We plan to incorporate gyroscope data for noise reduction and for access to 3 dimensional position features. These position features would allow a user to play different drum types by making strike gestures in different locations.

Other avenues for expressive control will be explored. For instance, physical modelling, specifically the percussive convolution synthesis detailed by Aimi [14], will be examined. Because the general acceleration magnitude envelope is already available, it is possible to match this envelope to prerecorded audio-rate drum impulses and convolve these with the impulse responses of different drums. This will provide a more realistic and expressive output sound because it is based on the real response of a drum.

9. REFERENCES

- [1] S. Dahl, "On the beat: Human movement and timing in the production and perception of music," Ph.D. dissertation, KTH Royal Institute of Technology, 2005.
- [2] A. Z. Hajian, D. S. Sanchez, and R. D. Howe, "Drum roll: Increasing bandwidth through passive impedance modulation," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 1997.
- [3] A. R. Tindale, A. Kapur, G. Tzanetakis, P. Driessen, and A. Schloss, "A comparison of sensor strategies for capturing percussive gestures," *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME '05)*, 2005.
- [4] D. Young and I. Fujinaga, "Aobachi: A new interface for Japanese drumming," *Proceedings of the Conference on New Interfaces for Musical Expression (NIME '04)*, pp. 23–26, 2004.
- [5] J. N. Bott, J. G. Crowley, and J. J. LaViola, Jr., "Exploring 3D gestural interfaces for music creation in video games," *Proceedings of the 4th International Conference on Foundations of Digital Games*, 2009.
- [6] P. R. Cook, "Physically informed sonic modeling (PhISM): Synthesis of percussive sounds," *Computer Music Journal*, vol. 21, no. 3, pp. 38–49, 1997.
- [7] —, "Toward physically-informed parametric synthesis of sound effects," *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1999.
- [8] —, "Principles for designing computer music controllers," *Proceedings of the Conference on New Interfaces for Musical Expression (NIME '01)*, 2001.
- [9] J. L. Sebastian Heise, "A versatile expressive percussion instrument with game technology," *Multimedia and Expo, 2008 IEEE International Conference on MultiMedia and Expo (ICME '08)*, pp. 393–396, 2008.
- [10] J. Williamson, R. Murray-Smith, and S. Hughes, "Shoogle: Excitatory multimodal interaction on mobile devices," *Proc. of the Conference on Human Factors in Computing Systems (CHI '07)*, 2007.
- [11] A. Tanaka, "Mobile music making," *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME '04)*, 2004.
- [12] G. Essl and M. Rohs, "ShaMus - a sensor-based integrated mobile phone instrument," *Proceedings of the International Computer Music Conference (ICMC '07)*, 2007.
- [13] G. Weinberg, "ZooZBeat: a gesture-based mobile music studio," *Proceedings of the 2009 Conference on New Interfaces for Musical Expression (NIME '09)*, 2009.
- [14] R. M. Aimi, "Hybrid percussion: Extending physical instruments using sampled acoustics," Ph.D. dissertation, Massachusetts Institute of Technology, February 2007.

ACTIVE PRESERVATION OF ELECTROPHONE MUSICAL INSTRUMENTS. THE CASE OF THE “LIETTIZZATORE” OF “STUDIO DI FONOLOGIA MUSICALE” (RAI, MILANO)

Sergio Canazza, Federico Avanzini

SMC Group, Dept. of Information Engineering
University of Padova
Via Gradenigo 6/B, 35131 Padova
canazza@dei.unipd.it
avanzini@dei.unipd.it

Maria Maddalena Novati

RAI, Milano
Archivio di Fonologia
novati@rai.it

Antonio Rodà

AVIRES Lab., Dept. of Informatics
Viale delle Scienze, Udine
University of Udine
antonio.roda@uniud.it

ABSTRACT

This paper presents first results of an ongoing project devoted to the analysis and virtualization of the analog electronic devices of the “Studio di Fonologia Musicale”, one of the European centres of reference for the production of electroacoustic music in the 1950’s and 1960’s. After a brief summary of the history of the Studio, the paper discusses a particularly representative musical work produced at the Studio, *Scambi* by Henri Pousseur, and it presents initial results on the analysis and simulation of the electronic device used by Pousseur in this composition, and the ongoing work finalized at developing an installation that re-creates such electronic lutherie.

1. INTRODUCTION

Despite the fact that electroacoustic music is a young form of art, it is necessary to take care of its preservation, due to the limited life of the supports where electroacoustic music works are preserved, of the reading systems of the data, and of the instruments. Moreover, preservation and restoration of this works raises peculiar technical and philological issues. With particular regard to electrophone instruments, many technological generations have passed since the appearance of the first instruments, and many electronic components used in their construction do not exist anymore or are only available with difficulty.¹

The potential damages produced by a bad conservation or an inadequate restoration are irreversible.

The aim of this paper is to report on initial results of an ongoing project devoted to the preservation, analysis and virtualization of the analog electronic devices of the Studio di Fonologia Musicale. The final goal is to develop an

¹ Electrophones are considered to be the only musical instruments which produce sound primarily by electrical means. Electrophones are one of the five main categories in the Hornbostel-Sachs scheme of musical instrument classification [1]. Although this category is not present in the original scheme published in 1914, it was added by Sachs in 1940 [2], to describe instruments involving electricity.

installation consisting of a SW-HW system that re-creates the electronic lutherie of the Studio, allowing users to interact with such lutherie. In particular, the production setup originally employed to compose *Scambi* is considered as a relevant case study. Achieving the goal of the project implies (i) analyzing the original devices through both project schemes and direct inspection; (ii) validating the analysis through simulations with *ad-hoc* tools (particularly Spice – Simulation Program with Integrated Circuit Emphasis, a software especially designed to simulate analog electronic circuits [3]); (iii) developing physical models of the analog devices, which allow efficient simulation of their functioning (according to the *virtual analog* paradigm [4]); (iv) designing appropriate interfaces to interact with the virtual devices. The paper is organized as follows. Section 2 discusses the issues posed by preservation and restoration of electrophone instruments. Section 3 briefly summarizes the history of the Studio di Fonologia and of *Scambi*. Finally, Sec. 4 presents initial results on the analysis and simulation of the electronic lutherie used by Pousseur for the composition of *Scambi*.

2. PRESERVATION AND RESTORATION

In most cases, the electroacoustic musical piece, as the author has produced it, is made of various elements like a score, recorded music, suggestions for interpretation, and other materials which are often important for understanding the making of the piece itself. This lead to the need of preserving both graphics and textual materials (score, schemes, suggestions) and audio materials (musical parts or the whole piece), software (for sound synthesis, live electronics, etc.), and electrophone instruments. The first materials are usually on paper and are thus concerned with the more general problem of paper materials preservation. Audio materials are recorded on various supports in which a rapid degradation of the information occurs. A new interesting field is the preservation of electrophone musical instruments.

Preservation can be categorized into *passive* preservation, meant to defend the original instruments from external agents without altering the electronic components, and *active* preservation, which involves a new design of the instruments using new electronic components. Active preservation is

needed to prevent the equipments from disappearing, and it is desirable because it allows to access them on a wide scale (e.g. active preservation may allow to access the instrument in virtual spaces that can be accessed even remotely by large communities of users). Collaboration between technical and scientific competences (informatics as well as electronic engineering) and historical-philological competences is also essential.

In the field of audio documents preservation some relevant guidelines have been sketches along the years [5, 6], but most questions regarding the safeguard and the preservation of electrophone instruments remain unanswered, as the regulations in force do not provide for specific care or legislative obligations.

2.1 The instruments

Electroacoustic music instruments differ from traditional ones in many respects: the use of electric energy as the main sound producing mechanism, rapid obsolescence, the dependence on scientific research and available technology. Unlike Sachs [2], we prefer (from the standpoint of preservation) to cluster electrophone instruments in three categories: electroacoustic, electromechanical, and electronic (analogic or digital).

In an electroacoustic instrument, transducers transform acoustic vibrations into a voltage variation representing the acoustic pressure signal. Sound is produced through an amplification system, while the original acoustic sound is hardly perceivable. Examples are the microphone, the electromagnetic pick-up of the electric guitar, the piezoelectric pick-up of the turntable.

In electromechanical instruments, voltage variations are caused by sound storage on a rotating disk or a tape according to electromechanical, electrostatic, or photoelectric principles. The main electromechanical generator is the audio-wheel first used by Thaddeus Cahill in the early 1900's, for his Telharmonium. Successful electromechanical instruments include the Hammond organ (audio-wheel) and the Mellotron (magnetic tape). Unlike the electroacoustic case, in electromechanical instruments sound could be heard only through the amplification.

In electronic instruments, sound is synthesized by one or more electronic generators without any acoustic or mechanical vibrations. Electronic components used for synthesis range from valves and semiconductors to VLSI circuits, with analogue technologies being replaced by digital ones. Sound is synthesized through combination and interconnection of "primitive" components like oscillators, noise generators, filters, modulators, etc. Examples of electronic instruments are electronic organs and synthesizers.

Preservation of these instruments poses several problems. First of all, "today, probably, more electronic than acoustic instruments are produced and, within short time, it is likely that more electronic instruments will be produced than all the acoustic instruments made in the human history" [7]. Secondly, these instruments should be preserved not only for museal purpose, but also to preserve their functionality. In our opinion, it is necessary to keep alive the music in the present time independently from its original instru-

ments, whose careful preservation protects a cultural heritage useful to historical and musicological research.

It is also necessary to make a distinction between commercial instruments, produced on a large scale, and experimental prototypes realized in musical research labs. The former are typically closed and compact instruments whose operational aspects are well documented and often protected by patents. Large-scale production makes less problematic their preservation, in terms of availability of replacement components. On the contrary, experimental prototypes are harder to preserve, because of lacking technical documentation, as well as "cannibalism", i.e. the practice of reusing some components for the assemblage of new devices. This phenomenon also makes difficult to date prototypes, and to know their characteristics at the time when the musical work was realized.

Often, electroacoustic music production is not linked to a particular instrument, but to a *system* composed of several instruments. This requires preservation of the laboratories where all the steps of the musical work production process were performed. As an example, the study of electronic music in Köln has been reconstructed in the same configuration used in the 1950's. A similar approach has been followed for the Institute of Sonology of the Utrecht University, active in the 1960's. The exhibition at the Cité de la Musique in Paris includes a section dedicated to electrophone instruments related to the experience of real-time computer music in the 1980's.

3. THE STUDIO DI FONOLOGIA MUSICALE

3.1 History

The Studio di Fonologia Musicale [8] was founded in 1955 at the Milan offices of the Italian Radio-Television (RAI), under the initiative of the Italian composers Luciano Berio and Bruno Maderna. In a few years, the Studio became one of the European centres of reference for the production of electroacoustic music, by deploying cutting-edge devices for the generation and processing of sound. Often these devices were especially designed and crafted by Alfredo Lietti: oscillators, noise generators, filters, dynamic and frequency modulators. These were unique pieces, created with great care to meet the needs of the composers who attended the Studio.

In 1967 the Studio underwent a partial renovation. As a consequence, much of the older equipment was dismantled and has been lost. However, thanks to records kept in archives (photographs, schemas, drawings and articles) it is possible, in many cases, to know the characteristics and the functionality of most equipments that no longer exist. The Studio was closed in 1983 and the devices were disassembled and transported to Turin, where they remained packed in storage until 2003, when they were returned in the RAI headquarters in Milan.

The electronic lutherie of the Studio di Fonologia Musicale has recently been transferred to the Milan Museum of musical instruments: this inestimable technological and cultural heritage is now accessible to the general public in a permanent museum exhibition. However the electronic



Figure 2. The front panel of the *Selezionatore di ampiezza* (photo courtesy of M. Novati [8]).



Figure 3. Rear view of the *Selezionatore di ampiezza* (photo courtesy of M. Novati [8]).

to be simulated using electronic engineering tools (particularly Spice [3]).

The circuit of the *Selezionatore di ampiezza* utilized by Pousseur for *Scambi* is depicted in Figure 1. The figure reproduces the RAI project schemes, which are slightly different from the ones originally presented by Lietti in [13]. The circuit has two operating modes, which depend on the activation status of the EF50 pentode.

1. When the pentode is off, no current flows through the potentiometer P2, so that the secondary of the input transformer CC4201 is connected to ground. In this case, the input signal, scaled by the input transformer, passes unchanged through the twin diode 6H6. The following bridge, composed by three resistances and the potentiometer P1, renders the signal symmetric: by means of the connectors and the switch positioned in the rear of the device (see Figure 3) it is possible to tune the potentiometer P1 until the amplitudes in the upper and in the lower side of the bridge are equal. Finally, the dual triode 6SN7 amplifies the signal to drive the output stage.
2. When the pentode is on, the current flowing through the potentiometer P2 polarizes the secondary of the input transformer to the voltage V_p (depending on the position of the potentiometer). As a result, the current will flow through one of the diodes of the

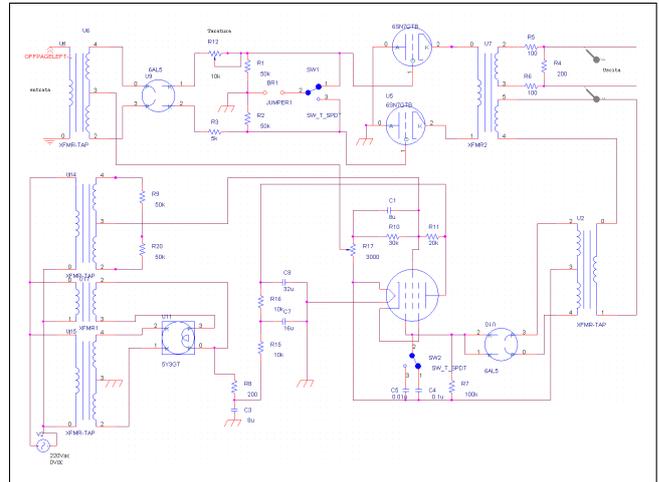


Figure 4. Electrical scheme designed to simulate the device.

6H6 tube only when the voltage of the input signal is, in absolute value, greater than the bias voltage V_p . If on the other hand the amplitude of the input voltage is less than V_p , the twin diode 6H6 is off and the output voltage will be zero. The knob at the bottom left of the front panel of the device (see Figure 2) lets the operator control the resistance value of P2 and the V_p threshold.

The activation status of the pentode EF50 depends by the feedback circuit: the output signal is drawn from the connectors 6 and 7 of output transformer G100, it is rectified by the twin diode 6H6, it is filtered by the RC circuit and, finally, is applied to the suppression grid of the pentode EF50.

If a signal is present in the output stage, the twin diode 6H6 is on and the current flows through the RC circuit, biasing the suppression grid to a negative potential, in respect to the cathode. In this condition, the flow of current is inhibited and the pentode is off. Conversely, when there is no signal in the output stage, no current flows through the RC circuit and then the grid will be at the same potential of the cathode. Under these conditions, the pentode is on. The biasing of the pentode is provided by the power supply circuit, that rectified the alternate power supply through the tube 5Y3. The speed at which changes the pentode is switched on and off depends on the speed at which the RC circuit responds to changes in the feedback signal, i.e. on the time constant of the circuit $\tau = RC$. The switch at the bottom right of the front panel (see Figure 2) lets the operator select between two time constants: $\tau_1 = 0.001s$ and $\tau_2 = 0.01s$.

4.2 Simulations

The circuit of the *Selezionatore di ampiezza* has been replicated in Spice. To this end, datasheets and libraries for all the circuit components have been found. Figure 4 shows a snapshot of the resulting Spice replica of the original circuit.

The output of the circuit was simulated in response to

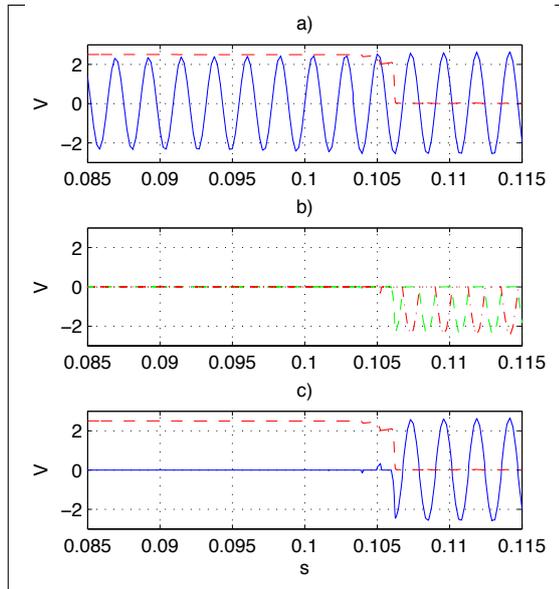


Figure 5. Spice simulation of the circuit excited with a sinusoidal signal with slowly increasing amplitude. a) input signal (blue solid line) and the voltage V_p (red dotted line); b) voltages at the two anodes of the twin diode 6H6; c) output signal (blue solid line) and the voltage V_p (red dotted line). The selector is turned on $C_2 = 0.1\mu F$.

- sinusoidal voltage signals with slowly increasing amplitude;
- stochastic voltage signals with zero mean.

Figure 5 shows the results of a Spice simulation. The input sinusoidal signal V_i is plotted in the upper frame (blue solid line), together with the voltage V_p (red dotted line), i.e. the bias voltage of the input transformer. As long as the input signal is below the voltage V_p , the twin diode 6H6 is off and the voltage at both its anodes is zero (see Figure 5 b). Therefore, the output voltage is also zero (see Figure 5 c) and the pentode EF50 is on. When the peak amplitude of the input voltage exceeds V_p , as at time $t = 0.103s$, the diode 6H6 begins to conduct (at least when $|V_i| > V_p$). Therefore, the output voltage is equal to the portion of the input waveform with amplitude greater (in absolute value) than V_p , the pentode EF50 starts to shut down, and the voltage V_p starts to decrease. This transition phase takes about 4 ms. After the transition, the twin diode 6H6 is always on (the two parts of the diode conduct either in correspondence of the positive and negative half-wave respectively), the output signal is approximately equal to the input, the pentode EF50 is powered off completely and the voltage V_p is close to zero.

Figure 6 shows the behavior of the circuit in response to a stochastic signal with a brownian spectral density. Initially, the input signal (a) is maintained below the voltage V_p and the output (b) is zero. At $t = 0.014s$ the stochastic signal exceeds, in absolute value, the threshold V_p and this causes the switching off of the pentode and the lowering of V_p . Until the input signal has a high average amplitude the pentode is off. When the input amplitude decreases, around $t = 0.05s$, the pentode starts to turn on, the voltage

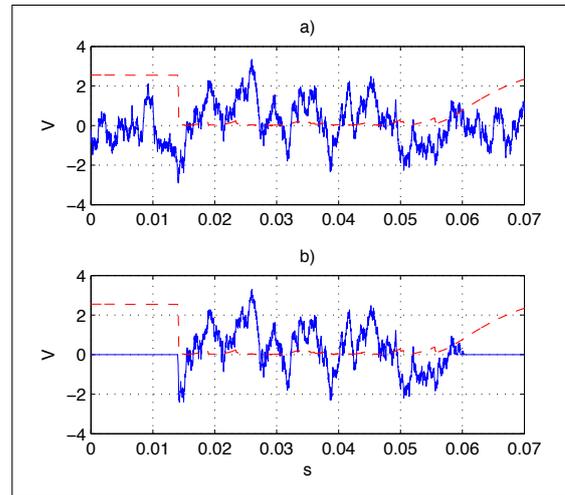


Figure 6. Spice simulation of the circuit excited with a stochastic signal. a) input signal (blue solid line) and the voltage V_p (red dotted line); b) output signal (blue solid line) and the voltage V_p (red dotted line). The selector is turned on $C_1 = 0.01\mu F$.

grows and the V_p signal output changes gradually to zero.

The speed with which the pentode turns on, as a result of a lower input voltage, depends on the time constant of the RC circuit connected to the pentode grid. A selector on the front panel of the amplitude selector allows you to choose between two capacitor values, $C_1 = 0.01\mu F$ and $C_2 = 0.1\mu F$, which correspond to the time constants $\tau_1 = 1ms$ and $\tau_2 = 10ms$. The results of Figure 6 are obtained with the selector on C_2 and it can be seen that the rise time of V_p is, as expected, about $2 \cdot \tau_2$.

Figure 7 compares the device responses to the same input signal as a function of the time constant of the RC circuit. Frames (a) and (b) are related to the time constant $\tau_1 = 1ms$ (selector on $C_1 = 0.01\mu F$): the rise time of V_p is faster than in the case showed in Figure 6. Moreover, it can be seen that the circuit reacts differently if the time constant is changed (see frames (c) and (d)). Unfortunately, no audio comparison and assessment of the end-result with respect to original signals is possible: as it is well known, no recordings of this device exist (we have only the complete sequences, where the Liettizzatore's outputs are processed).

5. CONCLUSION

The advent of digital technologies allowed to overcome many of the technical limitations of analog electroacoustic devices. However the question is whether the electroacoustic community is exploiting these digital resources for new experiments in form. The authors strongly believe that now the composers are able to explore in exhaustive way the potential of open forms using new media and new Human Computer Interfaces But, in order not to constantly "re-invent the wheel", works such as *Scambi* must be regarded as being more important now than fifty years ago.

In this sense, the authors are developing the *Music Bar* for active listeners. starting from the original project and

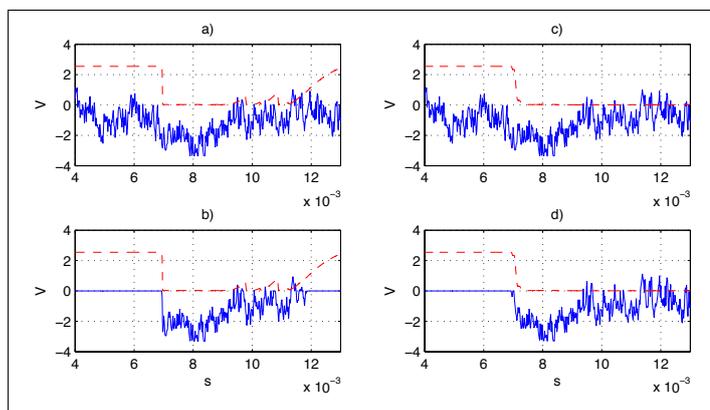


Figure 7. Spice simulation of the circuit excited with a stochastic signal. a) input signal (blue solid line) and the voltage V_p (red dotted line), selector on $C_2 = 0.1\mu F$; b) output signal (blue solid line) and the voltage V_p (red dotted line), selector on $C_2 = 0.1\mu F$; c) input signal (blue continuous line) and the voltage V_p (red dotted line), selector on $C_1 = 0.01\mu F$; d) output signal (blue solid line) and the voltage V_p (red dotted line), selector on $C_1 = 0.01\mu F$.

schemas of *Selezionatore di ampiezza*, the authors developed a system that allows the user-performer-composer to surf among the existing performances of *Scambi* and to create his own. Specifically, the installation will allow users to creatively interact with (i) virtual counterparts of the electronic devices of the Studio di Fonologia, and (ii) the production system of *Scambi* realized by Pousseur. The user-performer-composer will be able to surf among the existing performances of *Scambi* (e.g. by Luciano Berio and others), and to create his own, by selecting the original audio sequences used by Pousseur, and following (or not) the connecting rules proposed by the composer.

Future work will be devoted to the development of accurate and efficient virtual analog models of the original devices. Recently proposed techniques for the efficient simulation of nonlinear electric systems will be employed [14], and results from spice simulations of the circuits will be used to evaluate the accuracy of the virtual analog models.

A second key point for the effectiveness of the final installation is the design of the user interface. As future work, the authors intend to develop a tangible interface, able to recreate the corporeity, the materiality of the original interfaces: the inherent latencies between the user gestures and the corresponding effects on sound generation; the resistance and viscosity of the tape, which was slowed by hand by the composer-performer; and so on. All these physical characteristics influenced the composer and his way of interacting with the devices, and need to be preserved in their virtual counterparts.

6. ACKNOWLEDGEMENT

The project EA-CEA 2010-1174/001-001: *DREAM – Digital Reworking/reappropriation of ElectroAcoustic Music* has been funded with support of the Culture Programme of the European Commission. This publication reflects the

views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

7. REFERENCES

- [1] E. M. von Hornbostel and C. Sachs, "Classification of musical instruments," *Galpin Society Journal*, vol. 14, pp. 3–29, 1961, translated from the Original German by Anthony Baines and Klaus P. Wachsmann.
- [2] C. Sachs, *The history of musical instruments*, 1st ed. W W Norton & Co Inc (Np), 1940.
- [3] L. W. Nagel and R. A. Rohrer, "Computer analysis of nonlinear circuits, excluding radiation," *IEEE Journal of Solid State Circuits*, vol. SC, no. 6, pp. 166–182, 1971.
- [4] V. Valimaki, F. Fontana, J. O. Smith, and U. Zölzer, "Introduction to the special issue on virtual analog audio effects and musical instruments," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, pp. 713–714, Apr. 2010.
- [5] S. Canazza and A. Vidolin, "Preserving electroacoustic music," *Journal of New Music Research*, vol. 30, no. 4, pp. 351–363, 2001.
- [6] IASA-TC 03, *The Safeguarding of the Audio Heritage: Ethics, Principles and Preservation Strategy*, D. Schüller, Ed. IASA Technical Committee, 2005.
- [7] H. Davies, Personal communication, 2001.
- [8] M. Novati, Ed., *Lo Studio di Fonologia – Un diario musicale 1954-1983*. Milano, Italy: BMG Ricordi Publications, 2009, in italian.
- [9] J. Dack, "The 'open' form – literature and music," Paper presented at the 'Scambi Symposium', Goldsmiths College, 2005.
- [10] H. Pousseur, "Scambi," *Gravesaner Blätter*, no. IV, pp. 36–54., 1959.
- [11] M. Wilkinson, "Two months in the 'studio di fonologia'," *The Score*, no. 22, pp. 41–48, February 1958.
- [12] R. Fencott and J. Dack, "An interactive surface realisation of henri pousseur's 'scambi'," in *Proc. Int. Conf. Sound and Music Computing (SMC2011)*, 2011, in press.
- [13] A. Lietti, "Soppressore di disturbi a selezione di ampiezza," *Elettronica*, vol. 5, pp. 1–3, Sep. 1955, in italian.
- [14] F. Fontana and F. Avanzini, "Computation of delay-free nonlinear digital filter networks. Application to chaotic circuits and intracellular signal transduction," *IEEE Trans. Sig. Process.*, vol. 56, no. 10, pp. 4703–4715, Oct. 2008.

DESIGN AND APPLICATIONS OF A MULTI-TOUCH MUSICAL KEYBOARD

Andrew McPherson
Drexel University
apm@drexel.edu

Youngmoo Kim
Drexel University
ykim@drexel.edu

ABSTRACT

This paper presents a hardware and software system for adding multiple touch sensitivity to the piano-style keyboard. The traditional keyboard is a discrete interface, defining notes by onset and release. By contrast, our system allows continuous gestural control over multiple dimensions of each note by sensing the position and contact area of up to three touches per key. Each key consists of a circuit board with a capacitive sensing controller, laminated with thin plastic sheets to provide a traditional feel to the performer. The sensors, which are less than 3mm thick, mount atop existing acoustic or electronic piano keyboards. The hardware connects via USB, and software on a host computer generates OSC messages reflecting a broad array of low- and high-level gestures, including motion of single points, two- and three-finger pinch and slide gestures, and continuous glissandos tracking across multiple keys. This paper describes the system design and presents selected musical applications.

1. INTRODUCTION

Over the past decades, a great many electronic music controllers have been developed, but few approach the ubiquity of the piano-style keyboard. The keyboard's versatility and its large number of skilled performers ensure that it will maintain a prominent place in digital music performance for the foreseeable future.

The keyboard is by nature a discrete interface: on the acoustic piano as well as on most MIDI keyboards, notes are defined solely by onset and release, giving the performer limited control over their shape. Though certain MIDI keyboards are equipped with aftertouch (key pressure) sensitivity, this arrangement tends to lack nuance and flexibility. A key must be completely pressed before aftertouch can be used, so aftertouch cannot control articulation. Aftertouch is also difficult to use in rapid passage-work, and control is limited to a single dimension.

Separately, interest has been growing in multi-touch music interfaces, particularly touch-screen devices such as the Apple iOS family. Touch-based devices can be used for continuous or discrete control, and relationships between

gesture and sound can be dynamically adjusted. On the other hand, touch-screen interfaces lack the tactile feedback that is foundational to many musical instruments, and they require the consistent visual attention of the performer.

In this paper, we explore a synthesis between keyboard and multi-touch interfaces by integrating multiple touch sensitivity directly into each key. We present a new capacitive sensor system which records the spatial location and contact area of up to three touches per key. The sensors can be installed on any acoustic or electronic keyboard. The sensor hardware communicates via USB to a host computer, which uses the Open Sound Control (OSC) protocol [1] to transmit both raw touch data and higher-level gestural features to any sound synthesis program.

Integrating touch sensitivity into the keyboard creates a wide range of new expressive possibilities. In this paper, we will describe the hardware and software design of the touch system and present selected musical applications.

2. RELATED WORK

The idea of integrating touch sensitivity into the piano keyboard was first explored by Moog and Rhea [2], who constructed "multiply-touch-sensitive" keyboards recording the lateral and front-to-back position of the player's finger on the key surface as well as the continuous vertical position of the key itself. The sensors were installed in piano and organ-style keyboards, and the data was made available through a custom microcontroller interface. In this way, each key could be used to control up to three independent musical parameters.

Other authors have explored replacing the MIDI keyboard's discrete triggering with a continuous position measurement. Freed and Avizienis [3] demonstrate a keyboard featuring continuous position measurement and high-speed network communication. Our own previous work [4] uses optical sensing to measure continuous key position on the acoustic piano at 600Hz sampling rate. The sample rate is sufficient to capture anywhere from 10 to 100 points during the brief interval a key is in motion, recording not just the velocity but the *shape* of each key press. In addition to velocity, we demonstrate techniques for extracting up to four additional dimensions (percussiveness, weight into the keybed, depth, and finger rigidity) from each press [5].

The Haken Continuum [6] erases the mechanical boundaries between keys entirely, providing an interface capable of recording up to 10 touches in three dimensions. The continuous design facilitates glissando and vibrato gestures

as well as continuous note shaping, though maintaining correct intonation requires precision on the player's part.

Capacitive touch sensing (and, more broadly, electric field sensing) has also seen musical applications beyond the keyboard. Paradiso and Gershenfeld [7] discuss applications ranging from the classic Theremin to baton and bow tracking. Gaus et al. [8] use capacitive touch sensing to measure a guitarist's fingering on the fretboard. The Snyderphonics Manta¹ provides a hexagonal array of capacitive touch sensors whose mapping can be dynamically assigned in software. The Buchla Thunder², among other Buchla controllers, is also based on capacitive sensing.

2.1 Revisiting the Touch-Sensing Keyboard

Though previous touch-sensitive keyboard designs exist, we believe that this a concept ripe for future exploration. Open Sound Control [1] offers faster, more flexible communication options than MIDI, enabling the practical use of high-bandwidth performance interfaces. Driven by widespread adoption in consumer electronics, capacitive sensing technology has become both cheaper and more robust, with several integrated digital controller systems available.

Simultaneously, increasing availability of complex, multidimensional performance interfaces has stimulated an interest in the *mapping* problem: given a large input space of performer gestures, how can a performer's actions be translated into sound in the most flexible, intuitive manner? Highly multidimensional interfaces can be difficult to control, but recent work has identified strategies which go far beyond the traditional one-to-one linear mappings between input sensors and synthesizer parameters [9].

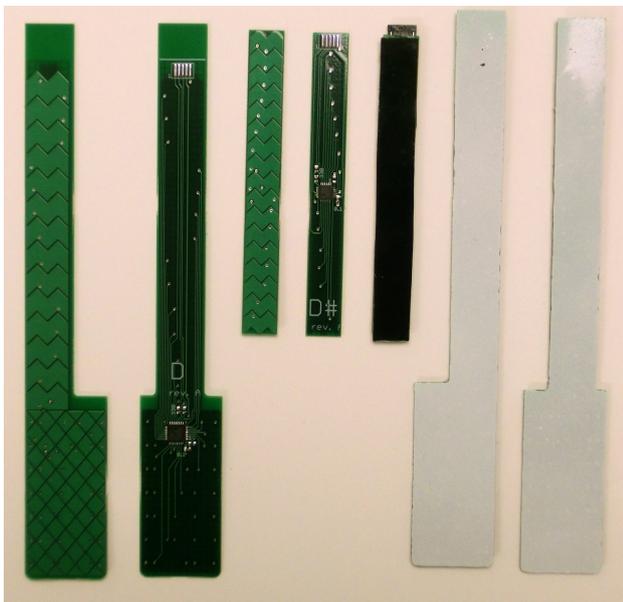


Figure 1. Multi-touch-sensitive piano keys. Boards shown from top, bottom, and with surface laminates.

¹ <http://www.snyderphonics.com>

² <http://www.buchla.com/historical/thunder/>



Figure 2. Two octaves of keys with controller board (top), mounted on a MIDI keyboard.

3. DESIGN OVERVIEW

In this context, we introduce a new system for measuring the position and size of up to three touches on each piano key. Black key position is sensed along a single front-to-back axis; white key touches are sensed in two axes on the wide front of the key and a single axis along the narrow back portion. Each key consists of a circuit board with an integrated circuit controller; the front surface of the board is laminated with thin plastic to provide a similar feel to the traditional key surface (Figure 1). The back is laminated with a thicker plastic sheet cut out around the components to provide a flat mounting surface. The entire 3mm thick assembly can replace conventional key tops (on acoustic pianos) or be fastened atop an existing keyboard (for molded plastic keyboards).

Figure 2 shows two octaves of keys attached to the controller board, which communicates with a computer via USB. Each key is scanned 125 times per second; the host computer processes the raw data to extract higher-level features including the addition and removal of touches, motion and resizing of existing touches, and multi-finger gestures including pinches and slides. These features are further described in Section 5.

4. SENSOR HARDWARE

Each key uses a Cypress Semiconductor *CapSense* controller [10] on a circuit board with either 17 or 25 sensor pads (black and white keys, respectively). Each pad forms a capacitor with respect to free space and a ground plane internal to the circuit board; a finger on or near the sensor surface increases its capacitance, which the controller reads as an 10-bit value.³ On startup, the controller reads baseline values for each sensor with no finger present, subtracting these baselines from subsequent readings. No electrical contact is required between the per-

³ Detailed principles of capacitive sensing can be found in Paradiso [7] and the Cypress datasheet [10].

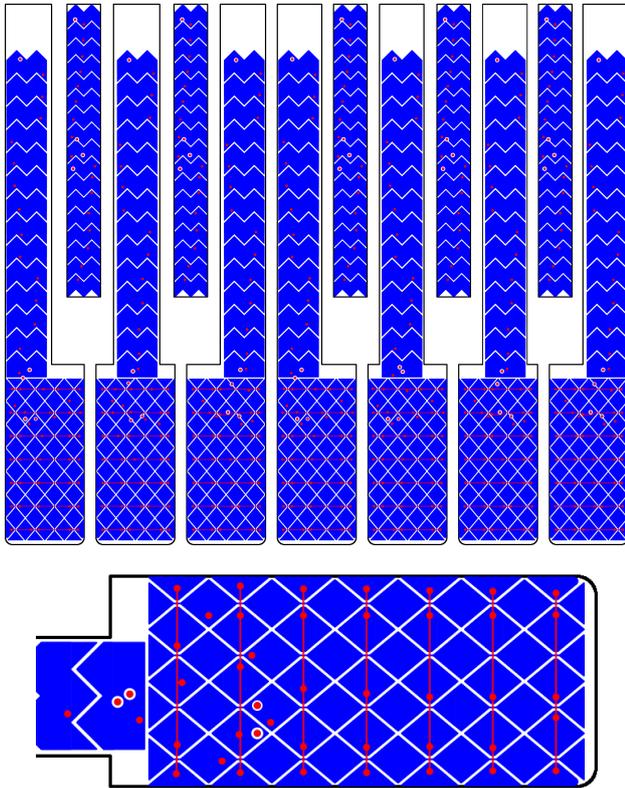


Figure 3. Top: Sensor pad layouts for one octave of keys. Blue pads are on the top layer of the board. Bottom: Close-up of XY sensor layout on white keys. Red wires on an inner layer connect pads into rows, blue wires on the top layer connect into columns.

former’s finger and the sensors, and unlike resistive touch sensors, no pressure is required.

Figure 3 shows the sensor pad layout, which is designed so that any touch will activate several adjacent sensors. On the black keys and the narrow part of the white keys, the pads form a linear slider capable of measuring touch position in one dimension. On the wider front of the white keys, small diamond-shaped pads are collected into an interlocking grid of 7 rows and 4 columns using two circuit board layers, allowing horizontal and vertical position to be sensed. Figure 4 demonstrates the calculation of touch position and size. Position is calculated as the centroid of

a group of adjacent active sensors:

$$Centroid = \left(\sum_{k=I_s}^{I_f} kC_k \right) / \left(\sum_{i=I_s}^{I_f} C_k \right) \quad (1)$$

where $I_s \leq k \leq I_f$ defines a range of sensor indices and the C_k represent capacitance values. Touch size (contact area) is proportional to the sum of all sensors in a group:

$$Size = \sum_{k=I_s}^{I_f} C_k \quad (2)$$

Raw centroid and size values are scaled to 0-1 range for later processing. Multiple independent touches can be sensed as long as their spacing exceeds the distance between sensor pads (4.75mm for black keys, 6.6mm for white). Centroid calculations are performed on each CapSense controller; a limit of 3 touches per key was chosen to provide a reasonable cap on calculation time. A complete sensor scan and calculation of centroids takes 4ms. Calculated vertical spatial resolution on the black keys is .08mm. On the white keys, resolution is .11mm in the vertical dimension and .09mm in the horizontal dimension.

4.1 Digital Communication

Figure 4 shows a diagram of the communication path. Each key transmits its calculated centroid and size data on a 400kbps I2C bus. I2C bandwidth and bus capacitance limitations preclude all keys from sharing a single bus. Instead, each octave of keys uses a separate I2C bus controlled by an Atmel AVR microcontroller. These “octave controllers” gather the data from each key and transmit it across a shared SPI connection running at 4Mbps. Flat ribbon cables connect the keys to the octave controllers to allow unimpeded key motion. System operation is ultimately controlled by an Atmel AVR with native USB capability (the “host controller”), which gathers the data from the SPI bus and transmits to the computer. The host controller is also responsible for regulating the timing of the sensor scans, initialization, and managing scan parameters.

Up to 8 octaves of keys can be managed by a single host controller. Each controller board (Figure 2) contains two octaves of keys, with a connection for a 13th key on the upper octave (since most keyboards end with a high C).

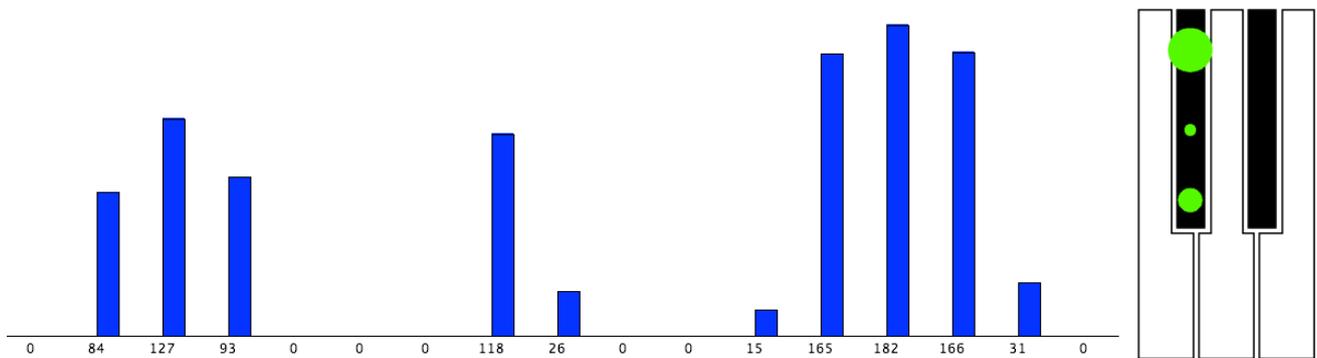


Figure 4. Discrete sensor pad readings are converted to touch position and size by calculating the centroid of multiple adjacent sensor values. Bars show sensor readings for C# key; green circles represent touch location and size.

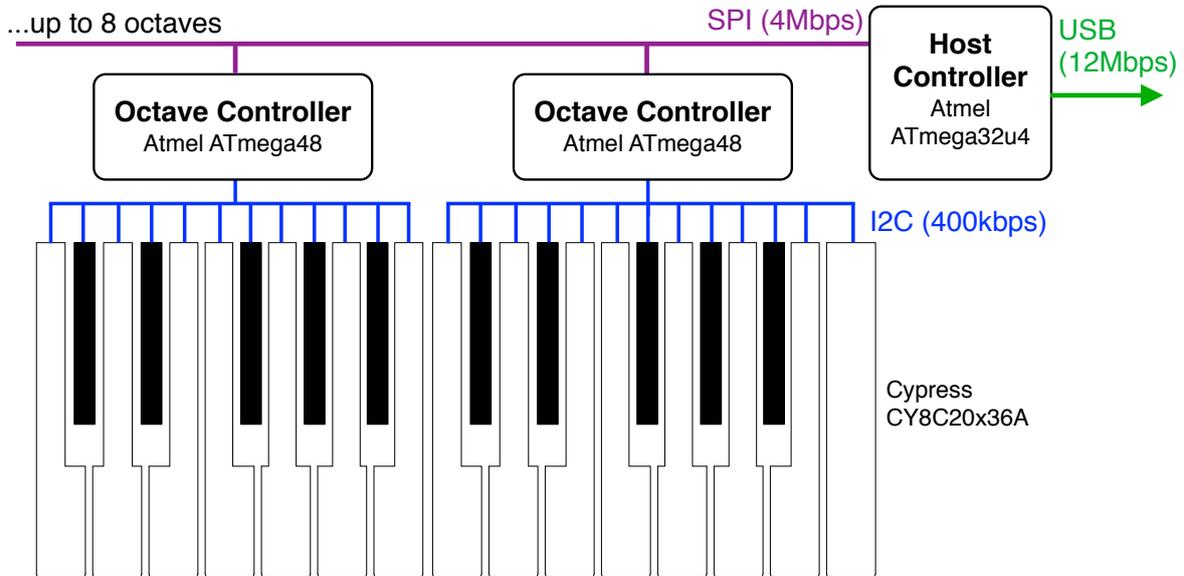


Figure 5. Communication architecture of the touch sensing system. Each octave of keys shares an I2C bus, with all octaves sharing a faster SPI connection. Communication with the computer takes place over USB.

To use more than two octaves, multiple boards are daisy-chained through flat ribbon cable connectors on either end. As only one host controller is needed, this part of the board can be broken off on the attached boards. On startup, the host controller dynamically determines the number of attached octaves.

5. DATA PARSING AND FEATURE EXTRACTION

The host controller appears to the computer as a USB communication class (CDC) device, which is natively supported by all major operating systems. Parsing software reads the incoming frames of touch data, producing OSC messages reflecting both raw touch values and higher-level gestural features. OSC messages can be sent to any port on the local machine or on the network, allowing connection to a broad array of synthesis software. Several programs exist which can further convert these OSC messages to MIDI data.

5.1 Raw Data Frames

For each key, transmitted OSC frames contain the octave and pitch class, the position and size of 3 touches (range 0-1, or -1 when not active), and for the white keys, a single horizontal position (-1 if the touch is not on the wide front of the key). Though it is possible to measure the vertical location of up to 3 touches, the sensor design only allows one unique horizontal position.

5.2 Multi-Key Gestures

Horizontal position sensing on the white keys allows the keyboard to emulate a ribbon controller: a touch can be tracked as it slides laterally across multiple keys. When it reaches the upper edge of one key, a new touch will register at the lower edge of the next. Our software stitches these touches together into a dedicated OSC “sweep” message (Table 1) containing a continuous location on the keyboard

as well as information on the keys currently sensing the sweep. This mode of interaction is particularly well-suited for musical interactions based on glissandos or heavy pitch vibrato, though slide messages could also be mapped to any continuous control application.

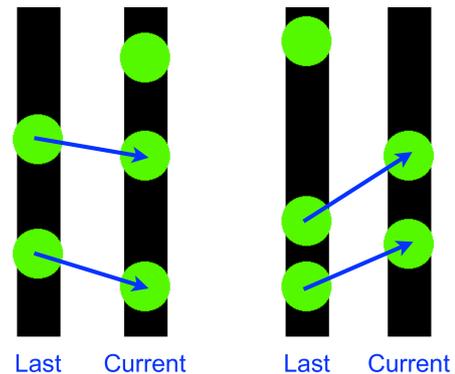


Figure 6. When the number of touches on a key changes, decide which touch was added or removed by minimizing the total motion of the other touches.

5.3 Gestural Feature Extraction

Data arrives from the keys as a series of discrete frames, but to provide expressive control over musical processes, the frames should be stitched together into a continuous picture of the performer’s gestural interaction with the keyboard. As Figure 6 demonstrates, this requires some calculation when multiple touches are considered. We assign each new touch a unique ID that an OSC client can track from one frame to the next. When the number of touches changes from one scan to the next, we decide which touch was added or removed by minimizing the overall distance traveled by the continuing touches.

Table 1 summarizes the higher-level features we extract,

which include not only single-finger gestures but multi-finger pinch and slide gestures often found on multi-touch mobile and tablet devices. Each mode of interaction can be mapped to distinct sound production behavior.

6. APPLICATIONS

The primary goal of adding touch sensitivity to the keyboard is to create an interface capable of *continuous, expressive* control. At the same time, the performer should not be burdened with an overly complicated interface, nor should he be expected to touch each key with pinpoint accuracy in order to produce acceptable sounds. The following set of examples demonstrate musical mappings that increase the range of subtlety available to the performer while still remaining straightforward to play.

6.1 Expressive Plucked String Synthesis

On the guitar, harp, and other plucked string instruments, the performer interacts directly with the string. Not only is a wide dynamic range possible; the player can also change the timbre of the note by plucking at different locations on the string, or by using the fingernail versus the softer fingertip. By contrast, traditional keyboards allow only dynamic control.

We use Csound⁴ to create a virtual instrument based on the `pluck` opcode, which simulates a plucked string using the Karplus-Strong algorithm. The timbre of the synthesized pluck depends heavily on the initial conditions of the virtual string. We use touch location and touch size (measured at the time of note onset) to control the location and sharpness of the pluck. Specifically, the string's initial position is given by two cubic segments (Figure 7); the

⁴ <http://www.csounds.com/>

location of the peak along the string corresponds to touch location on the key, and a smaller touch size produces a sharper curvature. Both of these mappings simulate the conditions of actual physical string plucks: for example, a touch with the fingertip near the end of the key will produce a bright, thin sound, where a touch with the ball of the finger in the middle of the key will produce a rounder timbre with reduced high-frequency content.

6.2 Physically-Modeled Piano with Dynamic Action

The pianist interacts with the instrument's strings through a mechanical abstraction: the key controls a complex series of levers terminating in a felt-covered hammer. Strike point, hammer hardness, and the parameters of the bridge and soundboard are all fixed by mechanical design. On the other hand, physically-modeled piano synthesis has made great strides over the past decade, and it presents no such mechanical restrictions.

This mapping uses the Modartt PianoTeq synthesis software⁵, which allows all major mechanical parameters to be dynamically assigned by MIDI Control Change messages. The Osculator program⁶ is used to map OSC messages from our system to MIDI Control Change values. Key velocity retains its usual meaning. Vertical touch position at onset is mapped to strike point within a constrained range around its default point, giving the pianist more control over the timbre of each note while ensuring sensible musical results. Touch size maps to hammer hardness (smaller touches produce a harder hammer). Like the preceding example, these mappings give the keyboard player an intuitive sense of interacting directly with the piano strings.

⁵ <http://www.pianoteq.com/>

⁶ <http://www.osculator.net/>

| OSC Path | Types | Data Contents | Description |
|-------------------------------|----------|--|--|
| /touchkeys/raw | iiffffff | octave (0-7), note (0-12), location/size pairs: [0, 1, 2] (range 0-1, or -1 if not active), horizontal location (-1 for black keys or upper portion of white keys) | Raw touch data |
| /touchkeys/on | ii | octave, note | Key became active |
| /touchkeys/off | ii | octave, note | All touches ended |
| /touchkeys/add | iiiifff | octave, note, touch ID, total # touches (1-3), new vertical location (0-1), new size (0-1), new horizontal location | New touch added |
| /touchkeys/remove | iiii | octave, note, ID, # remaining touches (1-2) | Existing touch removed |
| /touchkeys/move | iiiff | octave, note, ID, vertical location, horizontal location | Existing touch moved |
| /touchkeys/resize | iiif | octave, note, ID, size | Existing touch changed size |
| /touchkeys/twofinger/pinch | iiiiif | octave, note, ID 0, ID 1, distance between touches | Two fingers pinched together or pulled apart |
| /touchkeys/twofinger/slide | iiiiif | octave, note, ID 0, ID 1, (unweighted) centroid between touches | Two fingers moved up or down together |
| /touchkeys/threefinger/pinch | iiiiif | octave, note, ID 0, ID 1, ID 2, distance between outer touches | Pinch with three fingers on key |
| /touchkeys/threefinger/slide | iiiiif | octave, note, ID 0, ID 1, ID 2, (unweighted) centroid of all three touches | Slide with three fingers on key |
| /touchkeys/multikey/sweep | iiiiiif | sweep ID, sweep octave position, sweep note position, key 0: [octave, note, touch ID, horizontal position], key 1: [octave, note, touch ID, horizontal position] | Continuous sweep across multiple white keys |
| /touchkeys/multikey/sweep-off | i | sweep ID | Multi-key sweep ended |

Table 1. List of OSC messages sent by the touch-sensitive keys, reflecting low-level data and higher-level gestural features. Types `i` and `f` specify integer and floating point data, respectively.

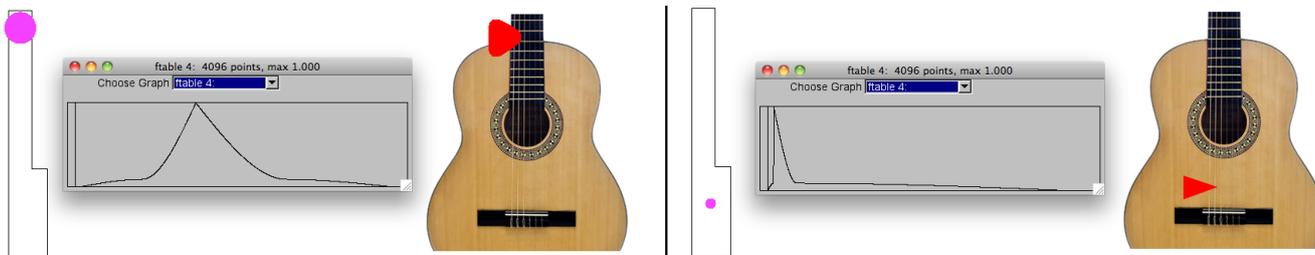


Figure 7. Expressive plucked string synthesis by mapping key touch location and size to pluck conditions. From left to right in each example: key touch, pluck initial conditions, physical analogy.

We explored several possibilities involving multiple fingers, including a mapping where a note played with two fingers increased the unison width of the piano strings in proportion to the distance between the touches, creating a “honky-tonk piano” effect when the two fingers were widely spaced. We also explored using the width between two fingers to modulate the impedance of the bridge, which affects the note decay. Widely spaced fingers create an unusually long sustain, and closely spaced fingers create notes with a clipped, muted quality. There is no obvious physical analogy for multi-finger touches, so the best mapping will depend on the specific musical situation.

In practice, this application was limited to monophonic playing, since changing PianoTeq settings affected all notes. Also, while some PianoTeq settings, including strike point and hammer hardness, took effect immediately, instrument-wide parameters such as bridge impedance exhibited a lag of up to one second, limiting their utility in practical performance situations. With suitable software adjustments, polyphonic dynamic piano modeling should be possible.

6.3 Continuous Timbre-Shaping

The preceding examples simulate (and extend) acoustic instruments whose note onsets are essentially discrete. We next show how touch sensing can be used to continuously modulate the sound of a note. We created a Csound instrument in which a harmonically-rich pulse waveform is passed through a resonant low-pass filter, similar to many classic analog synth topologies (Figure 8).

Using one finger, vertical position on the keys controls the filter cutoff frequency, and on the white keys, horizontal position can be used to bend the pitch up and down. The volume of the note can be changed with a two-finger “pinch” gesture, with a wider distance between fingers corresponding to higher amplitude. When three fingers are used, total finger spacing (“pinch”) moves the note’s fundamental frequency up the harmonic series of that key, with wider distance selecting a higher harmonic. Average position of the three fingers (“slide”) controls filter cutoff.

6.4 Compound Instruments

A simple mapping produces novel and useful results: on each key press, the number of fingers on the key selects between one of three instrumental voices. Incoming MIDI Note On messages are routed to one of three channels depending on how many fingers are currently on the corre-

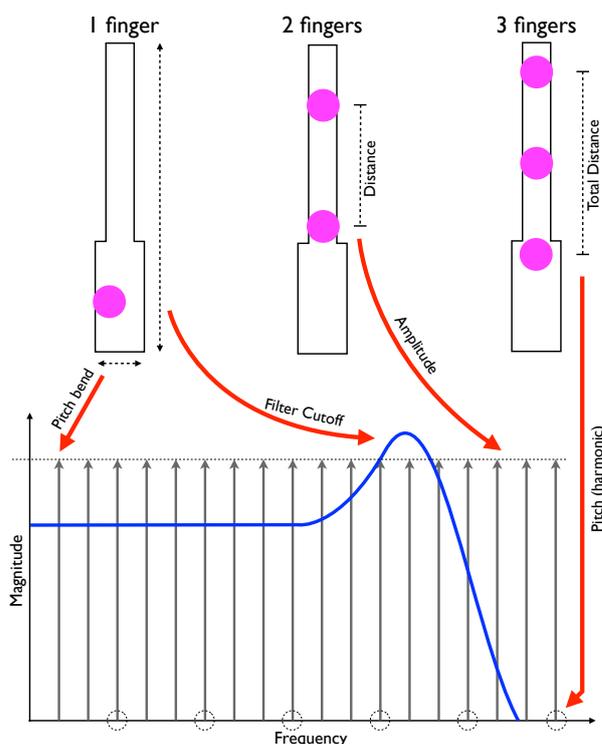


Figure 8. Continuous note-shaping using one, two, and three finger gestures.

sponding key (Figure 9). Note Off messages are sent to all three channels to avoid stuck notes.

Channel programs are configured so that one finger produces a piano sound, two fingers produce a bass sound, and three fingers produce percussion sounds. This arrangement allows a performer to play multiple instruments simultaneously from a single keyboard, with instrument selection performed on a note-by-note basis.

7. DISCUSSION

7.1 Guidelines for Effective Mappings

In developing mappings, two principles should be considered. First, the best musical results are often obtained from a subset of the possible controls. Assigning a separate meaning to every dimension can ultimately degrade the quality of performance by making the system needlessly complex. Second, the inherent asymmetry between the white keys, which sense horizontal position, and the black

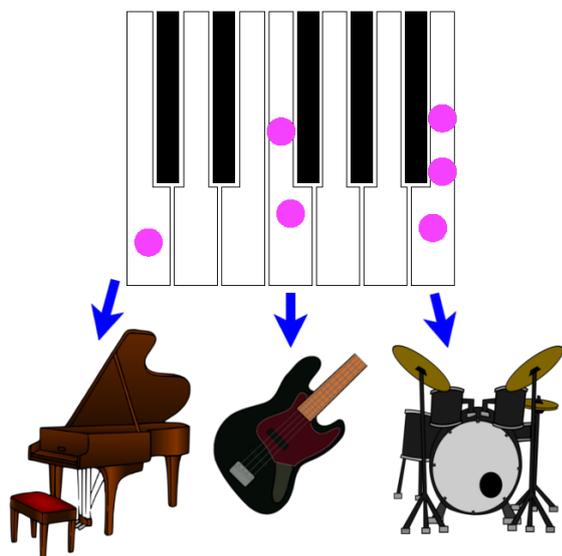


Figure 9. Compound instrument which selects one of three voices depending on the number of fingers on the key.

keys, which do not, must be addressed. This asymmetry reflects the keyboard itself, where the black keys offer a smaller contact area. In most cases, the most important musical parameters should be controllable from every key and not only the white keys.

7.2 Future Work

We recently used continuous position sensing to quantify key press gestures in multiple dimensions [5]. In a similar vein, future work will employ our touch sensors to examine the mechanics of traditional piano performance, with the goal of identifying relationships between expressive intent and finger motion. Dahl et al. [11] provide an overview of current approaches to expressive gesture sensing in keyboard performance, including measurements of pianists' hands, arms, torso and head. Real-time finger position on each key will be a valuable addition to these techniques, particularly since pianists devote a great deal of attention to the details of key "touch" (finger-key interaction).

A thorough understanding of finger motion in traditional performance also opens the door to new mapping strategies designed specifically around existing keyboard practice. We anticipate particular application to our work developing an electronically-augmented acoustic piano which uses electromagnets to shape string vibrations in real time [4], creating an intuitive performance interface extending the capabilities of the traditional piano.

7.3 Conclusion

We have presented a new interface which augments the keyboard with multiple touch sensitivity. The sensors install on existing keyboards, and associated software provides OSC messages for both raw touch information and higher-level gestures. We show several musical mappings, but the possibilities go well beyond these examples. Touch sensitivity can substantially enhance the expressivity of the keyboard by providing continuous control over several as-

pects of an instrument, and in judiciously designed applications, these additional dimensions need not substantially increase the complexity of performance.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant # 0937060 to the Computing Research Association for the CIFellows Project.

8. REFERENCES

- [1] A. Freed and A. Schmeder, "Features and future of Open Sound Control version 1.1 for NIME," in *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*, Pittsburgh, PA, USA, 2009.
- [2] R. A. Moog and T. L. Rhea, "Evolution of the keyboard interface: The Bösendorfer 290 SE recording piano and the Moog multiply-touch-sensitive keyboards," *Computer Music Journal*, vol. 14, no. 2, pp. 52–60, Summer 1990.
- [3] A. Freed and R. Avizienis, "A new music keyboard featuring continuous key-position sensing and high-speed communication options," in *Proceedings of the International Computer Music Conference*, Berlin, Germany, 2000.
- [4] A. McPherson and Y. Kim, "Augmenting the acoustic piano with electromagnetic string actuation and continuous key position sensing," in *Proceedings of NIME*, Sydney, Australia, 2010.
- [5] —, "Multidimensional gesture sensing at the piano keyboard," in *Proceedings of the 29th ACM Conference on Human Factors in Computing Systems (CHI)*, Vancouver, Canada, 2011.
- [6] L. Haken, E. Tellman, and P. Wolfe, "An indiscrete music keyboard," *Computer Music Journal*, vol. 22, no. 1, pp. 30–48, 1998.
- [7] J. A. Paradiso and N. Gershenfeld, "Musical applications of electric field sensing," *Computer Music Journal*, vol. 21, no. 2, pp. 69–89, 1997.
- [8] E. Gaus, T. Ozaslan, E. Palacios, and J. L. Arcos, "A left hand gesture caption system for guitar based on capacitive sensors," in *Proceedings of NIME*, Sydney, Australia, 2010.
- [9] M. Wanderley and P. Depalle, "Gestural control of sound synthesis," *Proceedings of the IEEE*, vol. 92, no. 4, pp. 632–644, 2004.
- [10] Cypress Semiconductor, "PSoC CY8C20xx6A Family Technical Reference Manual," <http://www.cypress.com/?docID=25608>.
- [11] S. Dahl, F. Bevilacqua, R. Bresin, M. Clayton, L. Leante, I. Poggi, and N. Rasamimanana, "Gestures in performance," in *Musical Gestures: Sound, Movement and Meaning*, R. Godøy and M. Leman, Eds., New York, NY, 2010, pp. 36–68.

IMPROVED FREQUENCY ESTIMATION IN SINUSOIDAL MODELS THROUGH ITERATIVE LINEAR PROGRAMMING SCHEMES

Vighnesh Leonardo Shiv
Catlin Gabel School
shivv@catlin.edu

ABSTRACT

Sinusoidal modeling systems are commonly employed in sound and music processing systems for their ability to decompose a signal to its fundamental spectral information. Sinusoidal modeling is a two-phase process: sinusoidal parameters are estimated in each analysis frame in the first phase, and these parameters are chained into sinusoidal trajectories in the second phase. This paper focuses on the frequency estimation aspect of the first phase. Current methods for estimating parameters rely heavily on the resolution of the Fourier transform and are thus hindered by the Heisenberg uncertainty principle. A novel approach is proposed that can super-resolve frequencies and attain more accurate estimates of sinusoidal parameters than current methods. The proposed algorithm formulates parameter estimation as a linear programming problem, in which the L^1 norm of the residual component of the sinusoidal decomposition is minimized. It achieves 3.5 times the frequency resolution of Fourier-based approaches.

1. INTRODUCTION

Sinusoidal modeling, the problem of representing a signal as a summation of quasi-stationary sinusoids, is a fundamental task in sound and music signal processing. Sinusoidal representations have many important applications. From an analysis standpoint, they transform pulse code modulated signals into meaningful representations for perceptual tasks like auditory scene analysis [1]. From a synthesis viewpoint, they make signals amenable to manipulations including time-scale and pitch-scale modifications, and can be efficiently coded at a low frame rate due to their slowly time-varying nature [2].

State-of-the-art sinusoidal modeling systems adopt a two-phase approach to the problem. The first phase involves extracting the sinusoidal parameters—amplitude, frequency, and phase—in each analysis frame of the signal. The second phase chains these parameters across analysis frames into sinusoidal trajectories, generating sinusoidal “births” and “deaths” as necessary. This paper is concerned with the parameter estimation phase of these systems.

Most recent parameter estimation algorithms employ approaches based on Fourier analysis [3]. First, peaks in the short-time Fourier transform (STFT) of a given analysis frame are detected, either by detecting local maxima greater than a fixed threshold [2] or through more sophisticated algorithms like the cross-correlation method [1]. Then, sinusoidal frequency estimates are refined. Methods for estimating frequencies include interpolation techniques [4], [5], [6], time-frequency reassignment approaches [7], and derivative-based methods [8], [9], [10]. Finally, sinusoidal amplitudes and phases are estimated, either directly from the frame’s frequency spectrum or through more intricate methods like iterative analysis [11], [1].

However, Fourier analysis-based systems are subject to the Heisenberg uncertainty principle: they have limited simultaneous resolution in the time and frequency domains. Thus, long analysis frames are necessary to distinguish nearby frequencies, while short analysis frames are necessary to capture rapid parametric changes and short sound events like musical notes. For an analysis frame of length w and a rectangular window function, the Fourier transform can resolve frequencies $\frac{2}{w}$ apart [4]. Additionally, spectral leakage can cause amplitudes to be overestimated from the frequency spectrum. For these reasons, pure Fourier analysis-based systems may not be optimal for sinusoidal parameter estimation.

In this paper, a novel algorithm for sinusoidal frequency estimation is proposed, that has the ability to super-resolve frequencies with high precision. The algorithm formulates the sinusoidal model as a linear program, in which the objective function is the L^1 norm of the residual component of the signal’s sinusoidal decomposition. The quasi-stationary nature of the sinusoids, in conjunction with properties of the simplex algorithm [12], can be exploited to estimate parameters from analysis frame to analysis frame at a high frame rate efficiently.

Note that the problem of trajectory continuation in sinusoidal modeling is beyond the scope of this paper. This problem has been addressed in such works as [13], etc. Existing trajectory continuation algorithms can be applied to the proposed parameter estimation algorithm in order to extend it to a complete sinusoidal modeling system, although future work would include a trajectory continuation system optimized to the time domain-based nature of the proposed system.

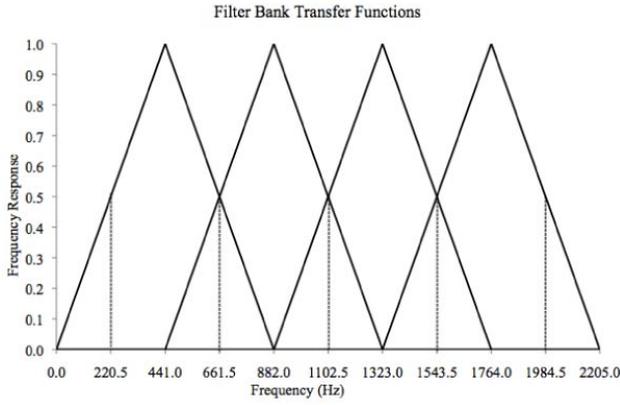


Figure 1: A partial graphical representation of the frequency responses of the filters employed in the system. While only four filters are illustrated in this figure, the filter bank used consists of 25 filters extending beyond 10 kHz. The filters not shown follow the same pattern as the filters displayed in this diagram.

2. SIGNAL PRE-PROCESSING

The input signal is first pre-processed by passing it through a filter bank. Each bandpassed component is more sinusoidally sparse than the raw signal, simplifying sinusoidal analysis on a per-component basis and improving the computational efficiency of parameter estimation. Separating the sinusoids into separate bands also lowers the dependence of sinusoidal analysis quality on signal complexity, increasing robustness.

The proposed system passes the input signal through a filter bank composed of 25 linearly spaced 9 ms-wide Blackman windowed-sinc filters, such that filter r had cutoff frequencies $441r - 220.5$ and $441r + 220.5$ Hz, $r \in [0, 25)$. The filters have approximately triangular responses within 882 Hz-wide bands with 50% overlap. Up to 10 kHz, each frequency has a frequency response of at least 0.5 in some filter band, preventing instability when recombining parameter estimates in bandpassed components.

3. LINEAR PROGRAM

This section describes the formulation of the sinusoidal modeling problem as a linear programming problem. This linear programming setup forms the core of the proposed algorithm. A hypothesis set of K sinusoidal frequencies is assumed; the next section will describe how these K frequencies are generated and how the discussed linear program can be used for sinusoidal parameter estimation.

The traditional sinusoids-plus-noise model takes on either of the following equivalent forms:

$$x(t) = \sum_{k=1}^K a_k(t) \sin(\phi_k(t)) + r(t) \quad (1)$$

$$x(t) = \sum_{k=1}^K a_k(t) \sin\left(\int_0^t \omega_k(u) du + \phi_k(0)\right) + r(t) \quad (2)$$

where a_k , ω_k , and ϕ_k are the time-varying amplitude, frequency, and phase respectively of the k th sinusoid, and r is the noise residual. a_k and ω_k are assumed to be locally stable, ϕ_k is assumed to be locally linear, and r is assumed to be a stochastic process. An analysis frame l of length w and a rectangular window function is traditionally defined as

$$x^l(t) = \begin{cases} x(t + hl) & \text{if } t \in \left[-\frac{w}{2}, \frac{w}{2}\right), \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where h is the hop size between analysis frames. Over a short analysis frame, the sinusoidal amplitudes and frequencies can be assumed to be constant. Thus, the sinusoids-plus-noise model of the l th analysis frame can be represented in the form

$$x^l(t) = \sum_{k=1}^K a_k^l \sin(\omega_k^l t + \phi_k^l) + r^l(t), \forall t \in \left[-\frac{w}{2}, \frac{w}{2}\right) \quad (4)$$

where a_k^l , ω_k^l , and ϕ_k^l are the amplitude, frequency, and initial phase respectively of the k th sinusoid in the l th analysis frame, and r^l is the residual component in the l th analysis frame. Given the assumption that the hypothesis set of frequencies is known, the goal here is to determine the sinusoidal amplitudes and phases.

The problem is first formulated as an optimization problem in which the L^1 norm of the residual component of the sinusoidal decomposition is minimized, as follows:

$$\begin{aligned} & \text{Minimize } \sum_{t=1}^w |r^l(t)| \\ & \text{such that, } \forall t \in \left[-\frac{w}{2}, \frac{w}{2}\right), \\ & x^l(t) = \sum_{k=1}^K a_k^l \sin(\omega_k^l t + \phi_k^l) + r^l(t). \end{aligned} \quad (5)$$

From this form, the problem can then be reformulated into a linear programming problem. First, by defining the variables

$$c_k^l \doteq a_k^l \cos(\phi_k^l), \text{ and} \quad (6)$$

$$s_k^l \doteq a_k^l \sin(\phi_k^l), \quad (7)$$

the sinusoidal component of the sinusoidal decomposition can be represented as a linear expression in terms of c_k^l and s_k^l for each k . If c_k^l and s_k^l can be solved for, a_k^l and ϕ_k^l can be trivially derived. This transforms the optimization problem to:

$$\begin{aligned} & \text{Minimize } \sum_{t=1}^w |r^l(t)| \\ & \text{such that, } \forall t \in \left[-\frac{w}{2}, \frac{w}{2}\right), \\ & x^l(t) = \sum_{k=1}^K [\sin(\omega_k^l t) c_k^l + \cos(\omega_k^l t) s_k^l] + r^l(t). \end{aligned} \quad (8)$$

The constraints are now linear in terms of all the used variables, but the objective function is not. This can be fixed

by expressing each residual data point as a difference of nonnegative variables:

$$r^l(t) = r_+^l(t) - r_-^l(t), \quad (9)$$

$$r_+^l(t), r_-^l(t) \geq 0. \quad (10)$$

The constraints are thus still linear expressions in terms of the used variables. As $r_+^l(t)$ and $r_-^l(t)$ are not independent variables, they cannot both be in the basis of the optimal solution found by a linear programming solver. Thus, for each t , one of $r_+^l(t)$ and $r_-^l(t)$ must equal zero, implying that

$$|r^l(t)| = r_+^l(t) + r_-^l(t). \quad (11)$$

As the objective function is now a linear expression, the final linear programming problem can be written:

$$\begin{aligned} & \text{Minimize } \sum_{t=1}^w [r_+^l(t) + r_-^l(t)] \\ & \text{such that, } \forall t \in \left[-\frac{w}{2}, \frac{w}{2}\right), \\ x^l(t) &= \sum_{k=1}^K [\sin(\omega_k^l t)c_k^l + \cos(\omega_k^l t)s_k^l] + r_+^l(t) - r_-^l(t), \\ & r_+^l(t), r_-^l(t) \geq 0. \end{aligned} \quad (12)$$

Given a hypothesis set of K frequencies, this program can be solved. While either the simplex algorithm or interior-point methods can be used to solve the program, the simplex algorithm is preferred for reasons that will be discussed in Section 6.

4. PARAMETER ESTIMATION

This section entails the linear programming-based algorithm for estimating the sinusoidal parameters. The general concept is to use linear programming with a temporal model to iteratively reduce the size of the hypothesis set of frequencies, while also retrieving information about amplitudes, phases, and phase derivatives. The result is that frequencies can be super-resolved and amplitudes and phases can be fit optimally to the input signal.

The input signal, the output of the original signal passed through a single filter, is first partitioned into short overlapping analysis frames of length 46 milliseconds, or $w = 2048$ for CD-quality audio. To estimate the sinusoidal parameters in a given analysis frame l , the analysis frame is first zero-padded with zero-padding factor z . z represents the theoretical increase in frequency resolution of the proposed algorithm over Fourier-based methods, although practical issues set a bound on how high z can be, as discussed in Section 5. A fast Fourier transform of the zero-padded analysis frame is then taken, resulting in spectrum $\hat{x}^l(t)$. The frequencies corresponding to each frequency bin can be considered an initial hypothesis set of sinusoidal frequencies. This hypothesis set can be pruned by eliminating frequency bins of low magnitude, since although an FFT will result in false positives due to spectral leakage, it will rarely result in false negatives. A threshold can be set proportional to the overall energy in the signal; frequencies with magnitudes below this threshold can be eliminated.

The linear program in (12) can be solved using the hypothesis set of frequencies, to determine estimates for the amplitudes and phases of all frequencies in the set. From these values, sinusoids of low amplitudes can be attributed to overfitting noise or modulations in the signal. These sinusoids' frequencies can thus be eliminated from the hypothesis set, and the linear program can be resolved to sharpen the sinusoidal parameters. This iterative pruning procedure of eliminating low-amplitude frequencies from the hypothesis set and resolving the linear program can be repeated until the hypothesis converges.

Fourier-based methods operating on a non-zero-padded rectangular-windowed analysis frame can resolve frequencies two frequency bins apart [4]. The proposed method aims to resolve frequencies two frequency bins apart in the zero-padded spectrum, although it may be able to resolve frequencies even more finely under certain conditions. The reason for this is that the procedure thus far makes an implicit assumption that the frequencies present in the signal will be near enough to a single frequency in E such that the linear program can accurately model the signal. However, this assumption is unsafe to make, as a frequency could very well lie, in the worst case, half way between two frequency bins, undermining the procedure. In such cases where frequencies are close to two adjacent frequencies, a smearing effect will occur making both frequencies appear significant, a phenomenon similar to spectral leakage in Fourier-based methods. If a single sinusoid can be assumed to be responsible for up to two nearby frequencies occurring in the hypothesis set, the proposed algorithm should have a frequency resolution of $\frac{2}{zw}$, z times more sensitive than Fourier analysis.

One remaining problem is that of how to estimate the frequency of a sinusoid "activating" the two frequencies adjacent to it. If the frequency is close enough to a frequency in the original hypothesis set, the smearing effect will not occur; in most cases, however, the smearing effect has been resolved somehow. The proposed approach here begins by estimating the number of resolvable sinusoids given the hypothesis set. The hypothesis set is partitioned into the minimum number of non-overlapping subsets H_1, H_2, \dots, H_n such that each subset contains only frequency bins adjacent to one another. As a single sinusoid will activate one or two frequencies around it and the algorithm's resolving power is two frequency bins, it thus follows that the number of resolvable sinusoids in H_i is

$$\left\lceil \frac{|H_i| + 1}{2} \right\rceil. \quad (13)$$

From here, the sinusoidal frequencies should be estimated. For each H_i , there are two cases to consider: where $|H_i|$ is even and where it is odd. In the former case, H_i can be further partitioned into two-element subsets of adjacent frequency bins. The sum of the two sinusoids in a given subset H_{ij} can be represented as third sinusoid with time-varying amplitude and frequency. At time $t = 0$, that

time-varying frequency is [1]:

$$f = \frac{f_2 + f_1}{2} + \frac{(f_2 - f_1)(a_2 - a_1) \left(\sec^2 \left(\frac{\phi_2 - \phi_1}{2} \right) \right)}{2(a_2 + a_1) \left(1 + \tan^2 \left(\frac{\phi_2 - \phi_1}{2} \right) \left(\frac{a_2 - a_1}{a_2 + a_1} \right)^2 \right)} \quad (14)$$

where the two amplitudes, frequencies, and phases are those corresponding to the two elements of H_{ij} . Thus, this equation can be used to estimate the true sinusoidal frequency with sufficient accuracy.

In the latter case, H_i contains an odd number of elements. In this case, a new set H'_i can be generated, containing a number of elements equal to $|H_i| + 1$. The distances between adjacent frequencies in H'_i are of the same spacing as in H_i , and the mean frequencies of both sets are equal. Once all sets H'_i have been generated, a linear program can be solved with a hypothesis set that replaces all H_i subsets with H'_i subsets. This allows this case to be reduced to the case of H_i having an even number of elements, for which a method has already been described.

Once the final set of frequencies is determined, a final linear program involving all of these frequencies can be performed to determine the amplitudes and phases of each sinusoid within an analysis frame.

5. FREQUENCY SPACING

One parameter of the algorithm that must be tuned is the frequency spacing between frequency bins of the FFT taken. The larger the frequency spacing is, the lower the frequency resolution of the algorithm. The algorithm may fit parameters poorly to the input signal, and the information determined may not be useful for high-level processing or generalizable to future analysis frames. One inherent advantage to using a temporal model is the ability to super-resolve frequencies, causing a bias toward a smaller frequency spacing. However, if this spacing is made too small, overfitting will occur. Many spurious frequencies may be detected in an attempt to fit a sinusoidal model as closely to the original signal as possible, undermining the purpose of sinusoidal modeling and making analysis of sinusoidal births and deaths difficult during trajectory continuation. A balance between these two extremes must be determined empirically.

6. EFFICIENCY CONSIDERATIONS

The proposed linear programming-based algorithm brings into question the preference for linear programming over a linear least squares approach. Indeed, a similar approach could be developed in which the L^2 norm of the residual component of the sinusoidal decomposition is instead minimized through linear regression, as opposed to the L^1 norm through linear programming. The main reason to opt for the linear programming approach is the efficiency gain it provides. Redundant information from linear program to linear program can be exploited when the simplex algorithm is used. The same cannot be said for linear least

squares, however. While efficiency is not the main focus of this paper, as linear programming is still a slow process, these efficiency measures should be included for the sake of both completeness and comparison to least squares methods.

The simplex algorithm is, in essence, a greedy search method that traverses a convex polytope representing the linear program to solve. Conceptually, with each iteration it moves to an adjacent vertex on the polytope such that the objective function becomes closest to optimal. After enough such iterations, the simplex method reaches a global maximum, as all local maxima are global maxima in a linear programming problem. Given the greedy nature of the algorithm, it makes sense that the efficiency measures discussed in this section focus on using redundancy to determine an initial basis close to the optimal solution, thus skipping phase I of the simplex algorithm and dramatically shortening phase II.

Redundant information can be exploited in the proposed system in two ways: from linear program to linear program both within and across analysis frames. Consider the former case first. When analyzing a single analysis frame, the numerous linear programs solved differ only in that some number of variables are removed from the system from linear program to linear program, changing the coefficient matrix. The dual simplex algorithm can be employed to efficiently make use of this redundancy. Rather than removing variables from the system, constraints can be added fixing those variables to zero. The dual simplex can then use the previously computed optimal basis as an initial dual feasible basis. As the variables removed were near zero to begin with, this initial basis should be only a few iterations away from the optimal basis, increasing efficiency. In contrast, linear least squares systems cannot increase their efficiency in this way. If the coefficient matrix changes in a linear least squares system, the entire system must be computed again. As the coefficient matrix slightly changes for each linear system solved for a given analysis frame as frequencies are eliminated, linear least squares becomes an expensive process for this task, unlike linear programming.

Now consider the latter case, where redundancy can be exploited across analysis frames. Given that the sinusoids are assumed to have quasi-stationary parameters, the final parameters found in the previous analysis frame should ideally be reused as the initial parameters for the current analysis frame. Given a high frame rate, it is possible to take advantage of this redundancy using the simplex algorithm. The frame rate is assumed here to be the original sampling rate of the signal to maximize the amount of redundancy between frames, so $h = 1$. The most efficient way to take advantage of this redundancy involves slightly altering the analysis frame system. Rather than defining an analysis frame as a shifted signal that is windowed from its first time sample, it can instead be defined as a non-shifted signal that is multiplied by a shifted window. While these definitions appear to be equivalent, the latter definition allows more efficient transitions between linear programs. A linear programming system requires

only a single row change from frame to frame, making matrix inversions more efficient. The linear program under this definition of an analysis frame l is thus

$$\begin{aligned} & \text{Minimize } \sum_{t=1}^w [r_+^l(t) + r_-^l(t)] \\ & \text{such that, } \forall t \in \left[l - \frac{w}{2}, l + \frac{w}{2} \right), \\ & x(t) = \sum_{k=1}^K [\sin(\omega_k^l t) c_k^l + \cos(\omega_k^l t) s_k^l] + r_+^l(t) - r_-^l(t), \\ & r_+^l(t), r_-^l(t) \geq 0. \end{aligned} \quad (15)$$

Note that the phases retrieved from c_k^l and s_k^l must be shifted accordingly, such that

$$\phi_k(l) \equiv \arg(c_k^l + i s_k^l) + \frac{\omega_k^l l}{f_s} \pmod{2\pi}. \quad (16)$$

Now that the analysis frame has been redefined such that a single row of the linear programming system changes from frame to frame, the previous analysis frame's parameters can be reused almost completely as the current analysis frame's initial parameters efficiently. Two situations should be considered here: when one of the two residual variables in the changing row is basic, and when both are non-basic, i.e. zero. Consider the former situation first. Luckily, this situation is both more common and more efficient to work with. Every initial value of c_k^l and s_k^l can simply be set to c_k^{l-1} and s_k^{l-1} , respectively, and every residual variable $r_+^l(t)$ and $r_-^l(t)$, for all $t < l + \frac{w}{2} - 1$, can be set to $r_+^{l-1}(t)$ and $r_-^{l-1}(t)$, respectively. The last variables to determine, $r_+^l(l + \frac{w}{2} - 1)$ and $r_-^l(l + \frac{w}{2} - 1)$, can simply be solved for using the values of the other variables and the restriction that one of the variables must equal zero. This supplies an initial primal feasible basis close to the optimal basis. Considering the second situation, the same steps can be repeated. However, the resulting values do not represent a basic feasible solution, as the basis contains one excess basic variable. An easy way to deal with this issue is to remove the variable that has the value closest to zero, and simply solve for the values of the other variables using a linear system. A single solution of a linear system is generally efficient enough given the sparsity of the matrix. Given an initial basis, the primal simplex method can be employed to traverse to the optimal basis.

7. SYSTEM EVALUATION

The system was evaluated experimentally in two ways. First, the ability of the system to detect the number of resolvable sinusoids was tested, in order to tune the zero-padding factor of the system. The zero-padding factor z and the number of sinusoids n were varied. The test set consisted of synthetic signals of n sinusoids separated in frequency by the expected resolution, where the lowest frequency was randomly generated. The metric used to evaluate the system was the average absolute error in the number of sinusoids detected.

As the data shows in Figure 2, the detection error behaves similarly for different numbers of sinusoids. For zero-



Figure 2: Curves indicating the relationship between the average absolute error in detecting the number of frequencies and the zero-padding factor used. Each curve represents signals of a different number of sinusoids.

padding factors of 3.5 or less, the system consistently detects the correct number of sinusoids, while the behavior of the system becomes more erratic with an upward trend in detection error as the zero-padding factor gets even higher. Taking this information into account, a zero-padding factor of 3.5 was selected for the second experiment.

The purpose of the second experiment was to directly test the frequency estimation quality of the system. A test set of synthetic signals was constructed in which, in addition to the sinusoidal amplitudes, the exponential decay rates of the sinusoids were varied, so as to test the robustness of the system to common musical modulations. The decay rate of the sinusoids k , where the time-varying amplitude is of the form ae^{-kt} , and the number of sinusoids s were varied. The metric measured here was the average absolute frequency estimation error among sinusoids. As Figure 3a shows, the average frequency estimation error is consistently low and robust against both the number of sinusoids and their exponential decay rates, with the average error never exceeding 2.5 Hz.

Although this paper focuses only on frequency estimation in sinusoidal models, the linear program additionally determines amplitudes and phases of the sinusoids. Thus, it is interesting to analyze the error in determining the amplitudes and phases. When the sinusoids are far apart in frequency, the amplitudes and phases are found with high precision, so it is more interesting to study the effects that sinusoids near each other in frequency have on each other when solving for amplitudes and phases. On the same test set used for measuring frequency estimation error, the average Euclidean distance between target and output amplitude-phase vectors as a percent of the magnitude of the target vector was also measured, a standard metric used in sinusoidal modeling research [1]. The target vector incorporated the amplitude and phase at the center of the analysis frame. The data in Figure 3b indicates that sinusoids near each other do impact each other's computed parameters. There are clear upward trends in amplitude-phase error with respect to both the exponential decay rate and number of sinusoids, which should be studied further in future work.

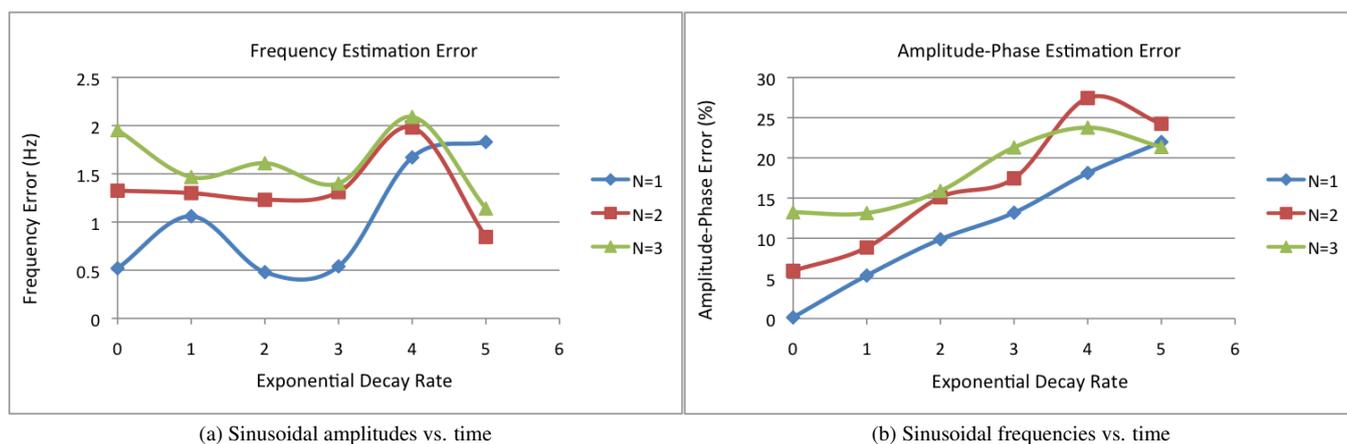


Figure 3: (a) Curves indicating the relationship between the average absolute error in estimating the sinusoidal frequencies and the exponential decay rate of the synthetic sinusoids used. (b) Curves indicating the relationship between the average error in estimating the sinusoidal amplitude and phase, as per the stated metric, and the exponential decay rate of the synthetic sinusoids used.

8. CONCLUSION

In this paper, a novel algorithm for sinusoidal frequency estimation was proposed. The algorithm was based on using linear programming to super-resolve frequencies present in an analysis frame. The frequency estimation system is able to super-resolve frequencies by a factor of 3.5 compared to Fourier-based systems on synthetic signals, and is robust to amplitude modulations. Future work should be directed toward improving amplitude-phase estimation and constructing trajectory continuation systems based on temporal models, as current temporal sinusoidal modeling techniques perform poorly over note transitions in comparison to spectral sinusoidal modeling techniques.

9. REFERENCES

- [1] T. Virtanen, "Audio signal modeling with sinusoids plus noise," Master's thesis, Tampere University of Technology, 2000.
- [2] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1986.
- [3] M. Betser, P. Collen, B. David, and G. Richard, "Review and discussion on STFT-based frequency estimation methods," in *presented at the Audio Engineering Society 120th Convention*, 2006.
- [4] J. O. Smith and X. Serra, "PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation," in *Proceedings of the International Computer Music Conference*, 1987.
- [5] B. G. Quinn, "Estimating frequency by interpolation using Fourier coefficients," *IEEE Transactions on Signal Processing*, 1994.
- [6] M. Abe and J. O. Smith III, "Design criteria for simple sinusoidal parameter estimation based on quadratic interpolation of FFT magnitude peaks," in *presented at the Audio Engineering Society 117th Convention*, 2004.
- [7] F. Auger and P. Flandrin, "Improving the readability of time-frequency and time-scale representation by the reassignment method," *IEEE Transactions on Signal Processing*, 1995.
- [8] M. Desainte-Catherine and S. Marchand, "High-precision Fourier analysis of sounds using signal derivatives," *Journal of Acoustic Engineering Society*, 2000.
- [9] S. Marchand and P. Depalle, "Generalization of the derivative analysis method to non-stationary sinusoidal modeling," in *Proceedings of the International Conference on Digital Audio Effects*, 2008.
- [10] M. Betser, "Sinusoidal polynomial parameter estimation using the distribution derivative," *IEEE Transactions on Signal Processing*, 2009.
- [11] P. Depalle and T. Hsliè, "Extraction of spectral peak parameters using a short-time Fourier transform and no sidelobe windows," in *IEEE 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*, 1997.
- [12] G. Dantzig, Ed., *Linear Programming and Extensions*. Princeton University Press, 1963.
- [13] M. Lagrange, S. Marchand, and J.-B. Rault, "Enhancing the tracking of partials for the sinusoidal modeling of polyphonic sounds," *IEEE Transactions on Audio, Speech, and Language Processing*, 2007.

PERSONALITY AND COMPUTER MUSIC

Sandra Garrido

University of New South
Wales
sandra.garrido
@student.unsw.edu.au

Emery Schubert

University of New South
Wales
E.Schubert@unsw.edu.
au

Gunter Kreutz

Carl von Ossietzky University
Oldenburg
gunter.kreutz@gmail.com

Andrea Halpern

Bucknell University
ahalpern@bucknell
.edu

ABSTRACT

Research has suggested that music preferences and an attraction to computers and technology are related to specific personality traits. This paper will argue that so-called ‘music-systemizing’ may be predictive of a preference for electronica, techno and computer-generated music. We report a preliminary study in which listeners who enjoy computer music based genres demonstrated a trend towards a higher mean score on the music-systemizing scale than those who enjoy love songs.

1. PERSONALITY AND MUSIC

Numerous studies have been conducted on the influence of personality traits on music listening preferences. Little and Zuckerman [1], for example, found that individuals with high scores in sensation-seeking evidenced a preference for highly stimulating music such as rock. Dollinger [2] found that extraversion was positively related to one kind of music with high arousal properties (jazz), and excitement seeking to another (hard rock). Openness to experience related to enjoyment of a variety of musical forms outside the mainstream of popular and rock music. Robinson, Weaver and Zillmann [3] reported that respondents scoring high on psychoticism or reactive rebelliousness enjoyed hard/rebellious rock more than low scorers. Similarly, a study by McCown, Keiser, Mulhearn and Williamson [4] found that psychoticism, gender, and extraversion were positively related to preference for enhanced bass. Schwartz and Fouts [5] also found considerable support for the hypothesis that adolescents’ listening choices were related to particular personality traits.

Further evidence of the influence of personality variables on music preferences is demonstrated by Chamorro-Premuzic and Furnham [6]. They reported that intellectually engaged individuals with higher IQs tended to use music in a different way from neurotic, introverted and non-conscientious individuals. In addition, it has been argued that individual differences in dissociation and absorption can influence enjoyment of sad music [7] and was found to be correlated with musically induced arousal [8]. Other studies have discussed gender-related differences in response to music [9].

2. PERSONALITY AND TECHNOLOGY

Evidence also suggests that personality has an influence on computer use. McNulty, Espiritu, Halsey and Mendez [10] found that personality traits measured on the Myer Briggs Type Indicator influenced the level to which medical students utilized Computer Aided Instruction (CAI). They found that students with a “sensing” preference tended to utilize CAI applications more than “intuitives”. Contreras [11] found that computer confidence was predicted by cognitive flex. Slate, Manuel and Brinson [12] reported a gender difference in attitudes towards computers and the Internet. Since both music preferences and general attraction to technology and computers appear to be related to certain aspects of personality, it could be expected that an attraction to electronica or computer-generated music would also be influenced by personality.

3. PERSONALITY AND ELECTRONICA

One of the most comprehensive studies on personality and music preferences was an investigation by Rentfrow and Gosling [13]. They examined the music preferences of over 3,500 individuals and identified four categories of music for which their participants demonstrated a preference: Reflective and Complex, Intense and Rebellious, Upbeat and Conventional, and Energetic and Rhythmic. Preferences for these music dimensions were found to be associated with the well-established ‘Big-Five’ and other personality factors. The Energetic and Rhythmic dimension was defined as including rap/hip-hop, soul/funk and electronica/dance music. This dimension was positively related to Extraversion, Agreeableness, blirtatiousness - “the tendency to respond to others quickly and effusively” [14], liberalism, self-perceived attractiveness and athleticism. It was negatively related to social dominance and conservatism. Thus the authors describe individuals who enjoy this kind of music as “talkative, full of energy, are forgiving, see themselves as physically attractive, and tend to eschew conservative ideals” (p.1249).

In relation to electronica, this study appears to have focused on the liveliness and rhythmic characteristics of the music. No distinction was made between the different genres within the ‘Energetic and Rhythmic’ dimension. In regards to computer-generated music and electronica, a different personality element may also be involved.

A similar study was conducted in the Netherlands involving 1044 adolescent participants [15]. In that study

four dimensions of music preference, similar to those in the Rentfrow and Gosling study, were labeled: Rock, Elite, Urban and Pop/Dance. In that sample, the trance/techno genre (comparable to the electronica/dance genre in the American study) loaded onto the Pop/Dance factor. Again it was found that adolescents who enjoyed Pop/Dance, tended to be high in Extraversion and Agreeableness. However, once again, no distinction was made between computer-generated music/electronica and other forms of pop or dance music.

The above cited research appears limited in that computer music encompasses a broad range of styles, covering popular, steady beat styles suited to dancing, through to experimental pieces that exploit freedom from the musical score with which more traditional forms are often associated. No attempt was made to distinguish computer-generated forms of music from other types of pop or dance music. The personality aspects reported in the above studies seem logically to relate to the popular/dance music end of the computer-music spectrum: Extraversion, Agreeableness, and 'blirtatiousness'. We here examine personality traits that may correlate with a tendency to listen to (in the present study) but also to create computer music.

4. EMPATHIZING-SYSTEMIZING

Baron-Cohen, Knickmeyer and Belmonte [16] developed a model based on broad empirical support of the Empathizer-Systemizer Theory (E-S-theory). This theory distinguishes two cognitive styles. Empathizing is defined as the capacity to respond to the emotions of other individuals, whereas systemizing represents the capacity to construct systematic relationships or to identify regularities of objects and events. A study by Nettle [17] found that these distinct cognitive styles were also related to differences in levels of interest in the arts or technology.

Kreutz, Schubert, and Mitchell [18] developed an instrument for the measurement of 'music empathizing' (ME) and 'music systemizing' [MS] by adapting a general empathizing-systemizing measure [19] to music. Kreutz et al. argue that the empathizing-systemizing distinction is a more accurate predictor of musical preference than gender, and that an individual's appreciation of music may be based on an attraction to the structural features of the piece for one person (systemizing), and the emotional content for another (empathizing).

Our viewpoint is that musicians who make their music mainly while interacting with an object (a computer) are more likely to be systemizers than those who primarily interact with other musicians (e.g. in a band, ensemble, orchestra and so forth). However, we also speculate that *listeners* of such musical styles may also populate these cognitive styles. This hypothesis can be tested experimentally, and in the following section we present a small-scale study which is part of a larger project, to see if aspects of our hypothesis might be supported.

5. METHODS

Two hundred and seventy-five participants were recruited into a larger study on emotion, personality characteristics and memory. The study was conducted online and commenced by requesting participants to nominate two favourite pieces of music and two of their least favourites. Later in the study they completed the ME and MS music cognitive styles questionnaire reported above [18]. Items from the ME/MS questionnaire designed to measure music empathizing focused on the emotion in and from the music such as 'I feel when listening to music I can understand the emotions the writer/performer is trying to express' and 'Music is important to me mainly because it expresses something personal and touching'. Items measuring music systemizing concentrated on structural features such as 'I like hearing the different layers of instruments and voices in a song/piece of music' and 'I especially like the organised way that music is laid out.'

Participants were asked to give the name of each piece, the composer and/or performer's name, a link to an online recording if possible, and to provide a brief description of the piece and why they chose it. The music nominated by participants was classified according to genre/style primarily based on the descriptions given by the participants. Given the possible overlap between genres, this method of classification was chosen since it provides some indication of the aspect of the music the participants were attracted to. In particular, for the present study, romantic/love songs and electronic/computer music styles were the key musical items we sought to identify. The romantic/love songs category included both ballad-type popular songs and instrumental/romantic music where the participants descriptions of the music indicated an attraction to its emotive qualities. Similarly, the electronica/computer-music category included music from various genres that have some electronic element.

The study had a 'listener' focus, meaning that we did not explicitly seek computer music composers. We wanted (1) to see how many electronic/computer music related pieces were *spontaneously* selected and (2) to compare their ME and MS scores against a control group, which was based on a random selection of the remaining participants who selected a love song as their favourite piece, but did not select a computer music piece (see Appendix for a list of participant-selected songs). 'Love songs' were considered a genre that would exemplify the preference for emotive music of the music-empathizer group. If our hypothesis is supported, we would not expect Love Song fans to demonstrate any systematic response due to music systemizing, unlike the computer music fans. If anything, Love Song fans may show a tendency to empathize or 'music-empathize', but this is a moot point, and an exploratory focus of the present investigation.

An alternative approach would be to seek out people who might like computer music. Our approach, we believe, is more robust because it was not possible for the participant to know that we were seeking computer

(among other) music lovers. The disadvantage of our approach is many participants are required to improve the chance of obtaining a statistically sufficient number of computer music loving participants. In an attempt to manage this problem but restrict ourselves to the current data set, two analyses were conducted.

6. ANALYSIS 1

After the first analysis, 18 participants were identified as nominating at least one piece that could be classified under the broad heading of computer music. A control group was also extracted from the data set, another 18 participants who selected at least one love/romantic song as their favourite piece, but no pieces that could be classified as computer music/electronica.

There were seven males and 11 females in the Love Song group and six females and 12 males in the Computer Music group. This ensured that any possible confounds due to gender (e.g. males being more likely to systemize) rather than cognitive style were reduced. The Love Song group had a mean age of 36 years and the Computer Music group averaged 26.3 years of age.

Scoring was performed as indicated in Kreutz et al [18], without adjustment that normalizes the scores to a standard deviation of about ± 10 . Figure 1 summarizes the results of the comparison of the groups. Neither group is significantly different on either the music empathizing scale or the music systemizing scale, as demonstrated by the large, overlapping Standard Error bar.

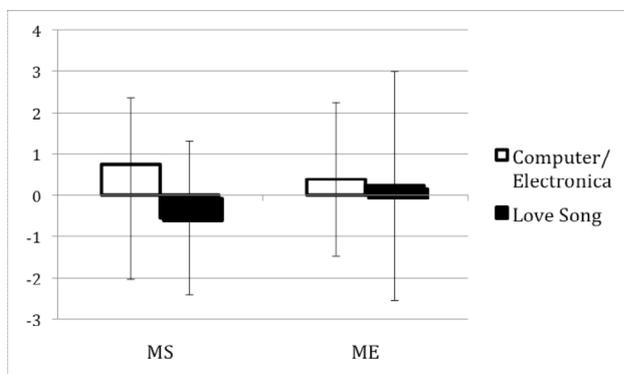


Figure 1. Results of Analysis 1, showing mean Music Systemizer (MS) and Music Empathizer (ME) scores by Computer/Electronica (n=18) vs. Love Song (n=18) favourite-piece selection in the survey. Error bar is $\pm 1SE$.

An ANOVA confirmed this, returning non-significant differences for both groups ($F(3,70) = .0675$, $p = .98$). Cohen's $d = 0.064$ for ME difference and $.095$ for MS difference, so effect sizes were negligible. While this result suggests that there is no difference in music cognitive style between Computer music/Electronica and Love song fans, it may also be the case that a real difference was hidden, for example in our group selection regime or due to lack of statistical power. Also, the mean age was higher in the love song group, suggesting another possible confound. Thus we conducted a second analysis,

with more stringent criteria for selection of groups, and with a slightly larger sample size.

7. ANALYSIS 2

In the second analysis, all of the 275 participants in the larger study were given a score for *exclusively* liking computer/electronica music. Recall that participants selected two of their favourite songs, and two of their least favourite. Participants were given 1 point for each electronica piece selected as a favourite or a score of -1 if a love song was one of their favourites. If a love song was one of the *least* favourite (hated) it received a score of 1. But if an electronica/computer music piece was chosen as a least favourite, a score of -1 was given to that participant. This approach means that the more exclusively the individual loves electronica, at the exclusion of (non-electronica/computer music) love songs, the higher the score, with a maximum possible score of +4. Conversely, a fan of love songs, but hater of electronica/computer music, will score closer to -4. All other styles (non-electronica, non-love songs, whether hated or loved) were scored zero. This time it was decided to exclude instrumental music or classical music of the Romantic period from the love/romantic songs group in order to obtain a closer mean age between groups.

Most participants had small or negative total scores (overall preferring love songs and not preferring Computer/Electronica). Nineteen participants were identified as 'exclusive electronica lovers' receiving a score of 1. A second group was extracted from the data set to balance this, which included a further 22 participants with the most extreme negative scores: that is, exclusive love song fans, scoring -2 or -3 (the lowest scores in the data set out of a possible -4). There were nine males and 13 females in the Love Songs group and 10 females and eight males in the Computer Music group (gender information for one participant was missing). Both groups had a mean age of 22 years.

An ANOVA revealed no significant differences between either group ($F(3,78) = 1.891$, $p = 0.14$). However, Figure 2 suggested some trends that may distinguish the two groups. Computer music/Electronica fans tended to score higher on both systemizing ($M = 2.42$, $SD = 6.51$) and empathizing ($M = 0$, $SD = 10.62$) scales relative to the Love Song fans ($M = -0.82$, $SD = 7.45$; $M = -4$, $SD = 9.778$, respectively). The possibility of an effect was supported by Cohen's-d statistics, which returned effect sizes of $.483$ for Music Systemizing and $.392$ for Music Empathizing.

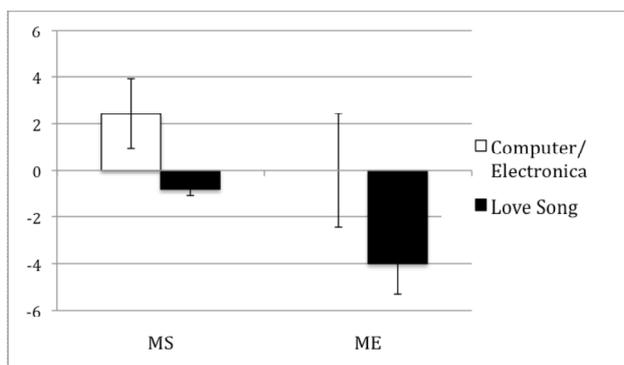


Figure 2. Results of Analysis 2, showing mean Music Systemizer (MS) and Music Empathizer (ME) scores by Computer/Electronica (n=19) vs. Love Song (n=22) groups, this time selected according to the ‘exclusiveness’ criterion, described in the text. Error bar is $\pm 1SE$.

8. DISCUSSION

Assuming our hypothesis was plausible, the possible reasons for non-significance were (1) not enough statistical power (only 18 computer music items were located among the cohort of participants in Analysis 1, and slightly more in Analysis 2), (2) we identified people who claimed to like listening to computer music, and not necessarily creating it (which is where the systemizing propensity may become important), and (3) we did not have the luxury of being able to identify music at the more experimental end of the computer music style spectrum (see Appendix), which would also more likely be distant from the presumably more sociable dance/pop music production styles. It is possible that many of the participants were attracted to electronica because of the other functions that it may serve, such as to support dancing.

Analysis 2 demonstrated that with more stringent criteria we could identify some possible effects of music cognitive style upon music preference. However, we did not predict that the Computer/Electronica group would have a greater mean Music Empathizing score than the Love Song group. The surprising difference could be explained by a residual confounding of groups: Despite our more stringent group membership criteria, since much of the music selected by the Computer/Electronica group was dance music, the dance function of this music may be a conduit for social interaction, and some empathizing in the traditional sense may have bled through to music empathizing. Both Cohen’s-d effect sizes are medium in Analysis 2, however, the music systemizing effect size was marginally larger than the music empathizing effect size.

While Music Systemizing and Music Empathizing may not be factors that determine or are correlated with music preference, we consider it important that such a null conclusion be subject to replication, particularly with larger sample sizes, alternate methods, and consideration of other covariates, such as age and gender. The research is

considered fruitful, despite being somewhat inconclusive because of the field’s interest in music and personality, and because the specifics of computer-based music production has not yet been tested in great detail. Importantly, we felt there was a theoretical perspective that could inform the relationship, in particular that music systemizers may enjoy interacting with machinery that produces music, more than other people, all things being equal.

9. CONCLUSION

Personality and music research is a relatively new area of study among music psychologists. None of the studies cited in the literature have attempted to identify whether personality characteristics might distinguish computer music lovers and creators from lovers of other music forms.

In particular we proposed that computer music creators would score high on a Music Systemizing scale because they were more likely to interact musically with an object (a computer) than people working with more conventional forms. We then conducted a study to see if this hypothesis might also generalize to *listeners* of computer music styles.

The study lent some support to our hypothesis – after a second analysis, a small trend was identified that demonstrated a higher mean music systemizing score for computer music lovers than a control group of Love Song (but not computer music) lovers. Our method avoided possible confounding effects due to participants being recruited because of their liking of a particular musical style (and therefore seeking to guess and try to support our hypothesis). Instead, we used data from a larger survey we conducted where the participants could not have been aware of the hypothesis under investigation. However, since such a method requires a large number of participants, we cannot be certain whether the lack of statistical significance in our finding was due to small numbers fitting into our test state criterion (computer/electronica music lovers) or because the hypothesis was not supported. Future work will also be required to determine whether our hypothesis can be sustained by, for example, comparing experimental computer music lovers and creators with more conventional music creators working with human ensembles.

Of course, we do not deny that computer music composers work with people nor that non-computer music composers could work with computers. The hypothesis simply suggests that people who tend to interact more with computers should be more ‘music systemizing’ than those who collaborate with other people, whether computer music composers or not.

10. REFERENCES

- [1] P. Litle and M. Zuckerman, "Sensation seeking and music preferences," in *Personality and Individual Differences*, 1986, pp. 575-578
- [2] S. J. Dollinger, "Research note: Personality and music preference: Extraversion and excitement seeking or openness to experience?" in *Psychology of Music*, 1993, pp. 73-77.
- [3] T. O. Robinson, J. B. Weaver, and D. Zillmann, "Exploring the relation between personality and the appreciation of rock music", in *Journal of Personality and Social Psychology*, 1996, pp. 259-269.
- [4] W. McCown, R. Keiser, M. Shea, and D. Williamson, "The role of personality and gender in preference for exaggerated bass in music," in *Personality and Individual Differences*, 1997, pp. 543-547.
- [5] K. D. Schwartz and G. T. Fouts, "Music preferences, personality style and developmental issues of adolescents," in *Journal of Youth and Adolescence*, 2003, pp. 205-213.
- [6] T. Chamorro-Premuzic and A. Furnham, "Personality and music: can traits explain how people use music in everyday life?" in *British Journal of Psychology*, 2007, pp. 175-185.
- [7] S. Garrido and E. Schubert, "Individual differences in the enjoyment of negative emotion in music: A literature review and experiment," in *Music Perception*, 2011, pp. 279-295.
- [8] G. Kreutz, U. Ott, D. Teichmann, P. Osawa, and D. Vaitl, "Using music to induce emotions; Influences of musical preference and absorption," in *Psychology of Music*, 2008, 101-126
- [9] E. Altenmuller, K. Schurmann, V. K. Lim, and D. Parlitz, "Hits to the left, flops to the right: Different emotions during listening to music are reflected in cortical laterisation patterns," in *Neuropsychologia*, 2002, pp. 2242-2256.
- [10] J. A. McNulty, B. Espiritu, M. Halsey, and M. Mendez, "Personality preference influences medical student use of specific computer-aided instruction (CAI)," [online] in *BMC Med Educ*, 2006, doi:10.1186/1472-6920-6-7.
- [11] C. L. M. Contreras, "Predicting computer self-confidence from demographic and personality variables and computer use," in *Quarterly Review of Distance Education*, 2004, pp. 173-181.
- [12] J. R. Slate, M. Manuel, and K. H. J. Brinson, "The "digital divide": Hispanic college students' view of educational uses of the Internet," in *Assessment & Evaluation in Higher Education*, 2002, pp. 75-93.
- [13] P. J. Rentfrow and S. D. Gosling, "The do re mi's of everyday life: the structure and personality correlates of music preferences," in *Journal of Personality and Social Psychology*, 2003, pp. 1236-1256.
- [14] W. B. Swann, and P. J. Rentfrow, "Blirtatiousness: Cognitive, behavioural, and physiological consequences of rapid responding," in *Journal of Personality and Social Psychology*, 2001, pp. 1160-1175.
- [15] M. Desling, T. Ter Bogt, R. Engels, and W. Meeus, "Adolescents' music preferences and personality characteristics", in *European Journal of Personality*, 2008, pp. 109-130.
- [16] S. Baron-Cohen, R. C. Knickmeyer, and M. K. Belmonte, "Sex differences in the brain: Implications for explaining autism," in *Science*, 2005, pp. 819-823.
- [17] D. Nettle, "Empathizing and systemizing: What are they, and what do they contribute to our understanding of psychological sex differences?" in *British Journal of Psychology*, 2007, pp. 237-255.
- [18] G. Kreutz, E. Schubert, and L. A. Mitchell, "Cognitive styles of music listening", in *Music Perception*, 2008, pp. 57-73.
- [19] A. Wakabayashi, S. Baron-Cohen, S. Wheelwrights, N. Goldenfeld, J. Delaney, and D. Fine, "Development of short forms of the Empathy Quotient (EQ-Short) and the Systemizing Quotient (SQ-Short)," in *Personality and Individual Differences*, 2006, pp. 929-940.

11. APPENDIX

Musical selections as reported by participants with regard to First Analysis reported in this study

Computer Music Selections

- “Deeply Disturbed” by Infected Mushrooms
- “Rocketeer”, by Far East Movement
- “Swoon” by The Chemical Brothers
- “You’ve Got the Love” by Florence and the Machine
- “Sa’eed” by Infected Mushrooms
- “Encoder” by Pendulum
- “Party in USA” by Miley Cyrus
- “Electric Feel” by MGMT
- “Bizarre Love Triangle” by New Order
- “Blackout” by Linkin Park
- “This Moment” (Original Mix) by Nic Chagall
- “Protection” by Massive Attack
- “Show Me Love” by Mobin Master
- “New Home” by Craving & Howe
- “Take Over Control” by Afrojack
- “Back Seat” by New Boyz
- “S & M” by Rihanna
- “2001 Spliff Odyssey” by Thievery Corporation

Romantic, Love Song Selections

- Love Theme from Romeo and Juliet, Henry Mancini.
- Nocturne, Opus 9 No. 2, Frederic Chopin
- “You are Beautiful”, James Blunt
- “If Love is the Providence”, Jin Guangxi
- “A Comme Amour”, Richard Clayderman
- “You Were Always on My Mind”, Elvis Presley
- “You Haven’t Seen the Last of Me”, Cher
- “Don’t Cry for Me Argentina”, performed by Julie Covington
- “We Belong Together”, Mariah Carey
- “How Do You Keep Love Alive”, Ryan Adams
- “Für Elise”, by Beethoven (chosen by two participants)
- “Try a Little Tenderness”, Otis Redding
- “The Heart Asks Pleasure First”, composed by Michael Nyman
- “Bella’s Lullaby”, Carter Burwell
- “Dream a Little Dream of Me”, performed by the Mamas and the Papas
- “Clair de Lune”, C. Debussy.
- “How am I Supposed to Live Without You”, Michael Bolton.

AUDITORY FEEDBACK IN A MULTIMODAL BALANCING TASK: WALKING ON A VIRTUAL PLANK

Stefania Serafin

Department of Architecture,
Design and Media Technology
Aalborg University Copenhagen
sts@create.aau.dk

Luca Turchet

Department of Architecture,
Design and Media Technology
Aalborg University Copenhagen
tur@create.aau.dk

Rolf Nordahl

Department of Architecture,
Design and Media Technology
Aalborg University Copenhagen
rn@create.aau.dk

ABSTRACT

We describe a multimodal system which exploits the use of footwear-based interaction in virtual environments. We developed a pair of shoes enhanced with pressure sensors, actuators, and markers. Such shoes control a multichannel surround sound system and drive a physically based sound synthesis engine which simulates the act of walking on different surfaces. We present the system in all its components, and explain its ability to simulate natural interactive walking in virtual environments.

The system was used in an experiment whose goal was to assess the ability of subjects to walk blindfolded on a virtual plank. Results show that subjects perform the task slightly better when they are exposed to haptic feedback as opposed to auditory feedback, although no significant differences are measured. The combination of auditory and haptic feedback does not significantly enhance the task performance.

1. INTRODUCTION

In the academic community, foot-based interactions have mostly been concerned with the engineering of locomotion interfaces for virtual environments [1]. A notable exception is the work of Paradiso and coworkers, who pioneered the development of shoes enhanced with sensors, able to capture 16 different parameters such as pressure, orientation, acceleration [2]. Such shoes were used for entertainment purpose as well as for rehabilitation studies [3]. The company Nike has also developed an accelerometer which can be attached to running shoes and connected to an iPod, in such a way that, when a person runs, the iPod tracks and reports different information.

In this paper we mostly focus on enhancing our awareness of auditory and haptic feedback in foot based devices, topic which is still rather unexplored.

We describe a multimodal interactive space which has been developed with the goal of creating audio-haptic-visual simulations of walking-based interactions. The system requires users to walk around a space wearing a pair of shoes enhanced with sensors and actuators. The position of such

shoes is tracked by a motion capture system, and the shoes drive a audio-visual-haptic synthesis engine based on physical models. The idea of enhancing shoes with sensors and actuators is similar to the ones we have been exploring in the context of the Natural Interactive Walking (NIW) FET-Open EU project¹ [7, 8]. The ultimate goal of this project is to provide closed-loop interaction paradigms enabling the transfer of skills that have been previously learned in everyday tasks associated to walking. In the NIW project, several walking scenarios are simulated in a multimodal context, where especially audition and haptic play an important role.

As a case study of the developed architecture, we describe an experiment where subjects were asked to walk straight on a virtual plank. The use of audio-haptic augmented footwear for navigation has not been extensively explored in the research community. An exception are the Cabboots [4], a pair of actuated boots which provide information concerning the shape of a path.

As another example, recently Takeuchi proposed Gilded Gait, a system which changes the perceived physical texture of the ground [5]. The Gilded Gait system is designed as a pair of insoles with vibrotactile feedback to simulate ground textures.

Recently, it has also been demonstrated that walking straight is a hard task also in the physical world. The research was justified by the common belief that people, when getting lost, tend to walk into circles. Subjects were asked to walk straight in two outdoor environments: a forest and a desert. When subjects were not able to see the sun, they walked in circles. It was suggested that veering from a straight course is the result of accumulating noise in the sensorimotor system, which, without an external directional reference to recalibrate the subjective straight ahead [6]. In this paper, we investigate the ability of subjects to walk straight on a narrow virtual plank, with the help of auditory and visual feedback. The results of this research can be applied to the fields of rehabilitation, navigation in virtual and physical worlds as well as entertainment.

2. THE OVERALL ARCHITECTURE

Figure 1 shows a schematic representation of the overall architecture developed. The architecture consists of a motion capture system (MoCap)(Optitrack by Naturalpoint),

Copyright: ©2011 Stefania Serafin et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

¹ <http://www.niwproject.eu/>



Figure 2. The developed haptic shoes used in this experiment.

the present configuration, the four actuators were driven by the same signal but could be activated separately to emphasize, for instance, the front or back activation, or to realize other effects such as modulating different, back-front signals during heel-toe movements.

A cable exits from each shoe, with the function of transporting the signals of the pressure sensors and for the actuators. Such cables were about 5 meters long, and they were connected⁷ to two 4TP (twisted pair) cables: one 4TP cable carries the sensor signals to a breakout board⁸, which then interfaces to an Arduino board; the other 4TP cable carries the actuator signals from a pair of Pyle Pro PCA1⁹ mini 2X15 W stereo amplifiers, driven by outputs from a FireFace 800 soundcard.¹⁰

Each stereo amplifier handles 4 actuators found on a single shoe, each output channel of the amplifier driving two actuators connected in parallel. The PC handles the Arduino through a USB connection, and the FireFace soundcard through a FireWire connection.

In our virtual environment the auditory feedback can be delivered by means of headphones (specifically, Sennheiser HD 650) or a set of 16 channels loudspeakers (Dynaudio BM5A speakers).

4. AUDIO-HAPTIC FEEDBACK

We developed a multimodal synthesis engine able to reproduce auditory and haptic feedback. Auditory feedback is obtained by the combination of a footstep and a soundscape sound synthesis engine. Haptic feedback is provided by means of the haptic shoes previously described. The haptic synthesis is driven by the same engine used for the synthesis of footstep sounds, and is able to simulate the haptic sensation of walking on different surfaces, as illustrated in [9]. The engine for footstep sounds, based on physical models, is able to render the sounds of footsteps both on solid and aggregate surfaces. Several different materials have been simulated, in particular wood, creaking wood, and metal as concerns the solid surfaces, and gravel, snow, sand, dirt, forest underbrush, dry leaves, and high grass as regards the aggregate surfaces. A complete description of such engine in terms of sound design, implementation and control systems is presented in [10]. Using such engine, we implemented a comprehensive collection of footstep sounds. As solid surfaces, we implemented metal, wood, and creaking wood. In these materials, the impact model

was used to simulate the act of walking, while the friction model was used to simulate the creaking sounds typical of creaking wood floors. As aggregate surfaces, we implemented gravel, sand, snow, forest underbrush, dry leaves, pebbles and high grass. The simulated metal, wood and creaking wood surfaces were furthermore enhanced by using some reverberation.

To control the audio-haptic footsteps synthesizer in our virtual environment, we use the haptic shoes: the audio-haptic synthesis is driven interactively during the locomotion of the subject wearing the shoes. The description of the control algorithms based on the analysis of the values of the pressure sensors coming from the haptic shoes can be found [9]. Such engine has been extensively tested by means of several audio and audio-haptic experiments and results can be found in [11] [12] [13] [14].

4.1 Movement to sound mapping

Figure 3 shows the dimensions of the path the users were asked to walk on. The mapping between feet movement and delivered auditory feedback was designed as follows:

- zone 1: a narrow band, 15 cm large and 120 cm long, corresponding to the straight direction to be covered. When both the feet stepped inside this zone, a creaking sound corresponding to the ecological sound of stepping on a creaking plank was provided.
- zones 2 and 3: two narrow bands, 10 cm large and 120 cm long, contiguous to zone 1 and placed at its left and right respectively. When one of the feet was stepping in this zone, no feedback was provided to it. However, subject was still able to continue the task by having the foot move to the correct zone 1. This is analogue to the situation of balancing with only one foot.
- zone 4 and 5: the areas contiguous to zone 2 and 3 and placed at its left and right respectively. When both the feet were inside such zone this was considered as failure for the task, and the recording of a long scream of a person falling down was triggered.
- zone 6: the area in front of zones 1, 2 and 3. When one or both the feet were inside this zone this was considered as success for the task, and the recording of a drum roll with an applause was triggered.
- zone 7: the area beyond of zones 1, 2 and 3. When both the feet were inside this zone, no sound was delivered.

⁷ through DB9 connectors

⁸ containing trimmers, that form voltage dividers with the FSRs

⁹ <http://www.pyleaudio.com/manuals/PCA1.pdf>

¹⁰ <http://www.rme-audio.com/english/firewire/ff800.htm>

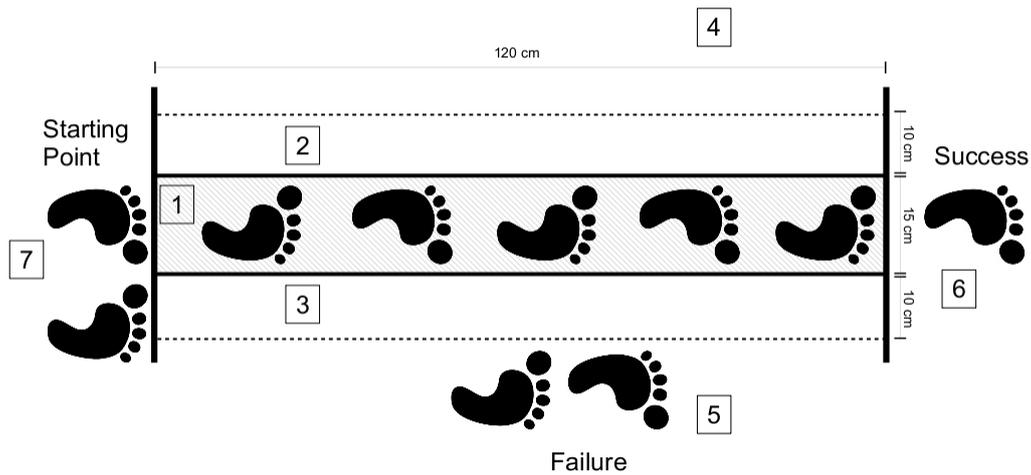


Figure 3. The zones in which the walking space was divided.

5. EXPERIMENT DESIGN

We designed an experiment in order to investigate the role of auditory and haptic feedback in facilitating a balancing task on a virtual plank. In such experiment, we asked subjects to walk straight in order not to virtually fall from the plank. Specifically, subjects were given the following instructions: "Imagine you are walking on a wooden plank. Your task is to walk from one side to the other. Walk slowly and pay attention to the feedback you receive in order to succeed on your task. If your feet are outside of the plank you will fall."

Figure 4 shows a subject performing the experiment. In this particular situation, no visual feedback was provided, and subjects were asked to walk being driven only by auditory and haptic feedback. The same stimuli were provided for the auditory and haptic simulation and designed as follows: when a user is walking on top of the virtual plank, the feet's position is detected by the motion capture system. In this case, the synthesis engine provides as a stimulus the sound and haptic feedback of a creaking wood.

5.1 Participants

The experiment was performed by 15 participants, 14 men and 1 woman, aged between 22 and 28 (mean=23.8, standard deviation=1.97). All participants reported normal hearing conditions. The participants took on average 6.8 minutes to complete the experiment.

Subjects were randomly exposed to the four following conditions: auditory feedback, haptic feedback, audio-haptic feedback and no feedback. Each condition was tried twice, given in total eight trials for each subject.

6. RESULTS OF THE EXPERIMENT

Table 1 shows the performance for each subject. The numbers in each row for each condition indicate whether the subject performed successfully the task ones, twice or never. The results show that feedback helps balance mostly when haptic stimuli are provided. In this case, 46.6% of the tasks

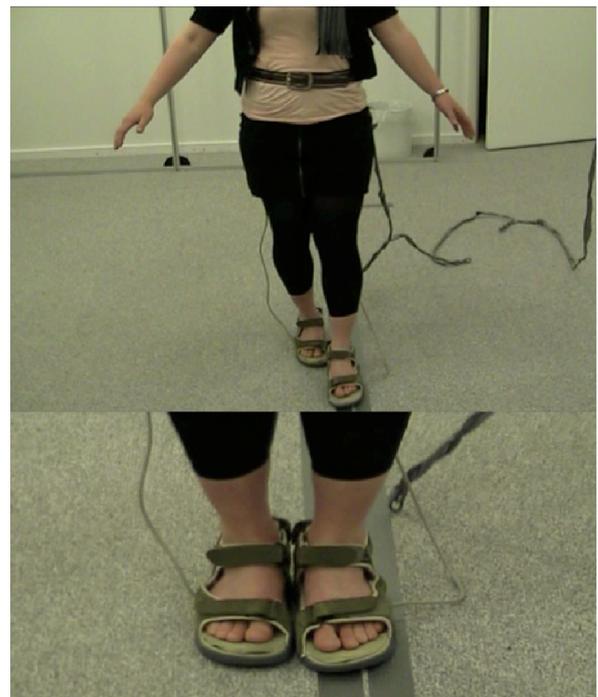


Figure 4. A subject performing the experiment of walking on a virtual plank. The tape on the floor represents the area where the plank is located.

Table 1. Summary of the results of the experiment. The number in each element of the matrix represents the times the task was successful (once, twice or never).

| Condition/ Subject number | Audio (A) | Haptic (H) | Audio-haptic (AH) | No-feedback (N) |
|------------------------------|-----------|------------|-------------------|-----------------|
| 1 | 2 | 1 | 2 | 1 |
| 2 | 2 | 1 | 1 | 1 |
| 3 | | 1 | | |
| 4 | 1 | 1 | | |
| 5 | 1 | | 2 | |
| 6 | 1 | 1 | 2 | 1 |
| 7 | | 2 | 1 | |
| 8 | 1 | | | |
| 9 | 1 | 1 | | |
| 10 | | | | |
| 11 | 1 | 1 | | |
| 12 | | 2 | 1 | 1 |
| 13 | | | | |
| 14 | | 1 | 2 | 1 |
| 15 | 2 | 2 | 1 | 2 |

were successfully completed. In the case where a combination of auditory and haptic feedback was provided, 43.3% of the tasks were completed. With only auditory feedback, 40% of the tasks were completed, while with no feedback only 26.6%. These results show how feedback slightly helps the balancing task. Haptic feedback performed better than the combination of auditory and haptic. This can be due to the fact that haptic feedback was provided directly to the feet, so the subjects had a closer spatial connection between the path they had to step on and the corresponding feedback.

A post-experimental questionnaire was also performed, where subjects were asked several questions on the ability to freely move in the environment, to adjust to the technology and to which feedback was the most helpful. Indeed, 7 subjects found the haptic feedback to be the most helpful, 6 subjects the auditory feedback and 2 subjects the combination of auditory and haptic feedback. One subject indeed commented that the most useful feedback was when there was background noise (the pink noise used to mask the auditory feedback) and only vibration was provided. All subjects claimed to notice the relationship between actions performed and feedback provided.

The subjects also commented that the feedback did not always match their expectations, since sometimes no feedback was provided. It is hard to assess if this was due to a technical fault of the system (for example, due to faulty tracking from the motion capture system), or to the fact that subjects were experiencing the condition with no feedback. Some subjects understood that the condition without any feedback was done on purpose, others confused it with a bug of the system.

Some subjects also commented on the fact that shoes were not fitting their size. Moreover, some felt disable without the visual feedback. One subject observed that he simply ignored the feedback and walked straight. This is an indi-

cation of his unwillingness of suspending his disbelief, and behave in a way similar to how they would behave when walking on a real narrow plank [15].

Overall, observations of most of the subjects showed that they were walking carefully listening and feeling the feedback in order to successfully complete the task. It is hard to assess whether the lack of feedback was the condition subjects were exposed to, the fact that they were outside the plank or a fault of the system. Afterall, previous mentioned research has shown that subjects do not walk straight even when they think they do. Some of the test subjects were noticeably not walking straight, although in the post-experimental questionnaire they commented on a faulty system. Very few understood that the lack of feedback was provided intentionally.

7. CONCLUSION

In this paper, we introduced a multimodal architecture whose goal is to simulate natural interactive walking in virtual environments. We present an experiment which assessed the role of auditory and haptic feedback, together with their combination, in helping subjects to complete the task of walking on a virtual rope.

The experiment provided some indications that haptic feedback at the feet level is more useful than auditory feedback when balancing on a virtual plank. Moreover, most subjects behaved in the virtual world as if they would have behaved in the real world, i.e., by walking slowly and carefully to try not to fall from the plank. More experiments, however, are needed to achieve a better understanding of the role of the different modalities in helping navigation and balance control.

8. ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Program under FET-Open grant agreement 222107 NIW - Natural Interactive Walking.¹¹ The authors would like to thank Vincent Hayward, Smilen Dimitrov and Amir Berrezag who built the sandals used in this experiment, and Jon Ram Pedersen and Kristina Daniliauskaite who collaborated in preliminary versions of the described experiment.

9. REFERENCES

- [1] A. Pelah and J. Koenderink, "Editorial: Walking in real and virtual environments," *ACM Transactions on Applied Perception (TAP)*, vol. 4, no. 1, p. 1, 2007.
- [2] J. Paradiso, K. Hsiao, and E. Hu, "Interactive music for instrumented dancing shoes," in *Proc. of the 1999 International Computer Music Conference*, 1999, pp. 453–456.
- [3] A. Benbasat, S. Morris, and J. Paradiso, "A wireless modular sensor architecture and its application in on-shoe gait analysis," in *Sensors, 2003. Proceedings of IEEE*, vol. 2, 2003.
- [4] M. Frey, "CabBoots: shoes with integrated guidance system," in *Proceedings of the 1st international conference on Tangible and embedded interaction*. ACM, 2007, pp. 245–246.
- [5] Y. Takeuchi, "Gilded gait: reshaping the urban experience with augmented footsteps," in *Proceedings of the 23rd annual ACM symposium on User interface software and technology*. ACM, 2010, pp. 185–188.
- [6] J. Souman, I. Frissen, M. Sreenivasa, and M. Ernst, "Walking straight into circles," *Current Biology*, vol. 19, no. 18, pp. 1538–1542, 2009.
- [7] Y. Visell, F. Fontana, B. Giordano, R. Nordahl, S. Serafin, and R. Bresin, "Sound design and perception in walking interactions," *International journal of human-computer studies*, vol. 67, no. 11, pp. 947–959, 2009.
- [8] R. Nordahl, S. Serafin, and L. Turchet, "Sound synthesis and evaluation of interactive footsteps for virtual reality applications," in *Virtual Reality Conference (VR), 2010 IEEE*. IEEE, 2010, pp. 147–153.
- [9] L. Turchet, R. Nordahl, A. Berrezag, S. Dimitrov, V. Hayward, and S. Serafin, "Audio-haptic physically based simulation of walking sounds," in *Proc. of IEEE International Workshop on Multimedia Signal Processing*, 2010.
- [10] L. Turchet, S. Serafin, S. Dimitrov, and R. Nordahl, "Physically based sound synthesis and control of footsteps sounds," in *Proceedings of Digital Audio Effects Conference*, 2010.
- [11] R. Nordahl, S. Serafin, and L. Turchet, "Sound synthesis and evaluation of interactive footsteps for virtual reality applications," in *Proc. IEEE VR 2010*, 2010.
- [12] R. Nordahl, A. Berrezag, S. Dimitrov, L. Turchet, V. Hayward, and S. Serafin, "Preliminary experiment combining virtual reality haptic shoes and audio synthesis," in *Proc. Eurohaptics*, 2010.
- [13] S. Serafin, L. Turchet, R. Nordahl, S. Dimitrov, A. Berrezag, and V. Hayward, "Identification of virtual grounds using virtual reality haptic shoes and sound synthesis," in *Proc. Eurohaptics symposium on Haptics and Audio-visual environments*, 2010.
- [14] L. Turchet, R. Nordahl, and S. Serafin, "Examining the role of context in the recognition of walking sound," in *Proc. of Sound and Music Computing Conference*, 2010.
- [15] M. Slater, "Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1535, p. 3549, 2009.

¹¹ Natural Interactive Walking Project: www.niwproject.eu

ANALYSIS OF SOCIAL INTERACTION IN MUSIC PERFORMANCE WITH SCORE-INDEPENDENT AUDIO FEATURES

Gualtiero Volpe, Giovanna Varni, Barbara Mazzarino, Silvia Pisano, Antonio Camurri

InfoMus - DIST - University of Genova

gualtiero.volpe@unige.it

ABSTRACT

Research on analysis of expressive music performance is recently moving its focus from a single player to small music ensembles, extending the analysis to the social interaction among the members of the ensemble. A step in this direction is the definition and the validation of a set of score-independent audio features that enable to characterize the social interaction in the ensemble, based on the analysis of the music performance. This paper focuses on the analysis of four different performances of a same music piece performed by a string quartet. The performances differ with respect to factors affecting the social interaction within the ensemble. The analysis aims at evaluating whether and to what extent a set of consolidated score-independent audio features, already employed for analysis of expressive music content and particularly suitable for string instruments, enable to distinguish among such different performances.

1. INTRODUCTION

Automatic analysis of music performance focused for long time on a single player. The aim was usually to identify a collection of audio and music features characterizing music performance in order to carry out various kinds of analysis and classification (e.g., genre, mood classification, and so on). Features concern several aspects: from low-level descriptions of the audio signal (e.g., energy-related features, spectral features, and so on), features derived from auditory models, features concerning higher-level music structures (e.g., articulation, phrasing). A typical example is the classification of the expressive content conveyed by the same music piece performed with different expressive intentions. In this case, this expressive content is represented and classified using different approaches, e.g., categorical approaches where it is characterized by expressive labels (e.g., the basic emotions) and dimensional approaches exploiting multidimensional spaces (e.g., the consolidated valence-arousal space), see for example [1][2][3][4][5].

A recent research trend shifts the focus to the *social dimension of music*. For example, Keller and colleagues analyzed piano duets and found that pianists were better

at synchronizing with their own than with others performances, and they were able to recognize their own recordings [6]. As referred by Trehub [7] music perception and performance are basic aspects of human being development, the study of which enables a better and a deeper understanding of the emotional development and group social interaction. Further, music group performances such as, for example, ensemble performance and choir performance involve coordination of movements and alignment of mental states, and require cooperation due to a shared goal, division of roles, and monitoring of progresses (e.g., [8][9]). In such a framework, automatic analysis of music performance needs to be improved with models and algorithms to probe and quantify the social interaction between the members of an ensemble or between the musicians and the audience.

The EU-ICT-FET Project SIEMPRE (2010-2013, see also: www.infomus.org/siempre) investigates non-verbal creative communication, in terms of entrainment, emotional contagion, co-creation and leadership, within groups of performers and audience. Some previous studies, forming the background of SIEMPRE and based on movement and gesture analysis, already explored the emergence of some of these phenomena [10][11]. However, a multimodal approach to music is needed and an important issue in this direction is to identify score-independent features that can describe ensemble performances characterized by differences in the social interaction of the members of the group.

This paper considers a set of score-independent audio features that already proved to be significant to distinguish between performances having different expressive and sensorial intentions [2]. The purpose of the work is to find out whether and to what extent the same features can also be used for analysis of social interaction in a small music ensemble, namely a string quartet. To this aim, the features were extracted and analyzed from four performances differing with respect to factors affecting the social interaction within the ensemble, with a particular focus on the functional roles of the players.

Section 2 briefly introduces the audio features that were taken into account. Section 3 describes the experiment that provided the data set for testing the features and the obtained results.

2. AUDIO FEATURES

Research on social interaction in music performance needs, as a first step, to identify a music ensemble which is suitable as a test-bed for investigation. For example, prelimi-

nary studies on violin duo performances provided encouraging results with respect to the possibility of developing techniques for the analysis of two relevant social signals: the level of synchronization established among the behaviors of each single member of a group and the emergence of functional roles (e.g., a leader) [12][10]. Analysis of the duo performance, however, suggested that synchronization and, especially, leadership may be better assessed with larger groups of players. In this direction, a quartet seemed an ideal ensemble, big enough to clearly display the phenomena under investigation, but not so big to become too complex for experimental set-up and data analysis. The focus of this work is thus on string quartets and required the identification of audio features particularly suited for string instruments.

Social interaction within a music ensemble (e.g., the emergence of a leader) may either follow the indications provided in a score, or may be the result of the application of specific performance techniques, or may arise from the internal organization of the ensemble, refined and tuned in many sessions and rehearsals where musicians play together. Social signals also emerge in performances that are not characterized by an exactly predefined score, such as improvisation. Score-independent audio features would allow to analyze social interaction within the ensemble without the need of the knowledge of the score.

Mion and De Poli [2] tested several score-independent audio features for their effectiveness in classifying performances which differ with respect to expressive and sensorial intentions. They asked three professional performers of violin, flute, and guitar to play in order to convey different expressive intentions, described by affective (happy, sad, angry, and calm) and sensorial (light, heavy, soft, and hard) adjectives. Using a sequential feature selection procedure followed by a Principal Component Analysis they identified 5 features yielding a high percentage of correct classification for the violin performances. These include both local features, computed on sliding time windows, and event features, computed on single events, segmented from the audio stream and identified by their onsets and their offsets.

Local features include:

- *Roughness* (R), or Sensory Dissonance, a feature characterizing the texture of a sound in terms of impure or unpleasant qualities. Such a sensation is associated with the physical presence of beating frequencies in the auditory stimulus. Leman and colleagues [13] developed a technique, based on auditory modeling, for computing roughness in terms of the energy provided by the neural synchronization to beating frequencies. As such, roughness is computed by applying a Synchronization Index Model to the output of an auditory peripheral model. The IPEM Toolbox for auditory-based musical analysis [14] provides a Matlab implementation of roughness on top of the auditory peripheral model of Van Immerseel and Martens [15].
- *Residual Energy* (RE) describes the stochastic residual of the audio signal, obtained by removing the

deterministic sinusoidal components [16]. Residual energy can be computed over different frequency regions, however, as Mion and De Poli show [2], the residual energy in the frequency range above 1805 Hz (*RE_h*) is particularly suited for the analysis of string instruments. *RE_h* is thus computed as the residual energy ratio in such a frequency band:

$$REh = \frac{\sum_{j \in H} |X_R(j)|^2}{\sum_{k=1}^{N/2-1} |X_R(k)|^2} \quad (1)$$

where *H* is the set of spectrum bins corresponding to frequencies higher than 1805 Hz and *X_R* is the spectrum of the residual component of the signal.

Event features are computed on single events in the audio stream. Onset detection is performed by using the algorithm proposed in [17]. Offsets are detected as suggested in [2] when the root-mean-square (RMS) of the temporal envelop of the audio signal falls by the 60% from its previous maximum value. Event features include:

- *Notes per second* (NPS), computed by dividing the number of onsets by the duration of the analysis window. In [2] analysis is performed with windows of 4-s duration and 3.5-s overlap, so that the window size allows to include a reasonable number of events, and it corresponds roughly to the size of the echoic memory.
- *Attack time* (A), computed as the time required to reach the RMS peak, starting from the onset instant.
- *Peak sound level* (PSL) computed as the maximum value of the RMS within the event, i.e., $PSL = \max(RMS(t))$.

3. ANALYZING SCORE-INDEPENDENT AUDIO FEATURES FROM A STRING QUARTET

3.1 Design and Material

The above-mentioned audio features were here applied to analyze the social interaction among players. The features were extracted from the multi-track recordings of a professional string quartet, Quartetto di Cremona. The recordings were carried out at Casa Paganini - InfoMus, in occasion of a concert of Quartetto di Cremona at the Opera House concert season, and they were performed in an environment very similar to a concert hall, with technology and scientists participating in the studies hidden in the upper level room. In such a way, an effective ecological environment with no perturbation on the investigated phenomena was set-up.

Players were asked to perform the first movement (*Allegro*) of the Streichquartet No. 14 by Schubert in four different conditions:

- *Regular condition*: players play as in a regular concert performance;
- *Switch condition*: the first violin plays the score of the second violin and viceversa;

- *Functional condition*: players are asked to follow a metronome and focus only on the gestures that are directly needed in the sound production process;
- *Over-expressive condition*: players emphasize gestures and affective intentions.

These four conditions were selected in order to emphasize variations in the social interaction. The regular condition is the reference condition, where the quartet plays as in a regular concert, thus applying all the usual mechanisms and techniques they learned and tuned. The switch condition introduces an explicit external action (the switch of the score of the first and second violin) to affect the normal social relationships between the members of the ensemble (e.g., a possible change of leadership). The functional (metronomic) condition operates on the social interaction in two different ways: on the one hand, it imposes a kind of external leader, the metronome; on the other hand it inhibits the expressive content conveyed by the piece. The over-expressive condition exaggerates the expressive content the music conveys, thus possibly requiring a higher degree of cohesion and synchronization in the group. Note that, even if the conditions also differ with respect to expressive content (in particular the functional and the over-expressive conditions), here the focus is not on the expressive content, but rather on social interaction. That is, the variation in the amount of expressive content to be conveyed is exploited as a way to affect the established social mechanisms of the ensemble in order to make the social variables emerge more evidently. Nevertheless, the presence of such variations of expressive content is a further motivation for choosing score-independent audio features that already proved to be significant in distinguishing between different expressive performances.

Each condition was repeated two times - two performances - with a short break in between them. Table 1 shows the experimental protocol. This resulted in the same piece repeated 8 times: 2 regular performances, 2 switch performances, 2 functional performances, and 2 over-expressive performances.

| No. | Condition | Performance |
|-----|-----------------|-------------|
| 1 | Regular | I |
| 2 | Regular | II |
| 3 | Switch | I |
| 4 | Switch | II |
| 5 | Functional | I |
| 6 | Functional | II |
| 7 | Over-expressive | I |
| 8 | Over-expressive | II |

Table 1: The order conditions were performed.

The recordings used in this work are part of a wider study aiming at measuring and evaluating social features in music ensemble performances, with particular reference to synchronization and leadership. Such a wider study also includes measures from physiological sensors and visual recordings. Initial results are discussed in [10].

3.2 Analysis and Results

The score was divided into 5 Parts based on salient points such as, for example, pauses, attacks, and changes in the dynamics. For each single Part and for each audio track of each player, the selected score-independent audio features were extracted using the same settings (e.g., window size, hop size, and so on) as indicated in [2]. These features were the dependent variables of the experiment, whereas the independent variables were *Condition* and *Performance*. The mean value of each feature in each Part was chosen as a synthetic descriptor for the Part. The effect of *Condition* and *Performance* and their combined effect were assessed with an RM two-way within subjects ANOVA on each of the features. The analysis was carried out on each of the five segmented Parts: 25 RM two-way within subjects ANOVA (5 audio features x 5 Parts).

Condition had no significant effect on any feature, whereas *Performance* had effect on RE Part III ($F = 9.7, p < 0.05$) and Part IV ($F = 25.94, p < 0.05$), NPS Part I ($F = 4.16, p < 0.05$), PSL Part II ($F = 9.24, p < 0.01$), and A Part II ($F = 4.72, p < 0.05$). A significant interaction *Condition* and *Performance* was found on NPS Part I ($F = 4.16, p < 0.05$), PSL Part II ($F = 6.13, p < 0.05$), A Part I ($F = 3.94, p < 0.01$), and Part II ($F = 4.19, p < 0.05$). Both the main factors and their interaction do not seem to affect the *R* feature. A further Bonferroni corrected post-hoc analyses could assess specific differences among the effects. Table 2 summarizes the number of Parts in which each feature was statistically significant with respect to the main factors and their interactions.

| Feature | Factor(s) | No. Parts |
|---------|-----------------------|-----------|
| NPS | Condition | 0 |
| NPS | Performance | 1 |
| NPS | Condition*Performance | 1 |
| R | Condition | 0 |
| R | Performance | 0 |
| R | Condition*Performance | 0 |
| A | Condition | 0 |
| A | Performance | 1 |
| A | Condition*Performance | 2 |
| RE | Condition | 0 |
| RE | Performance | 2 |
| RE | Condition*Performance | 0 |
| PSL | Condition | 0 |
| PSL | Performance | 1 |
| PSL | Condition*Performance | 1 |

Table 2: No. of Parts in which each feature was significant.

Figure 1 depicts the interaction plot between *Condition* and *Performance* for the NPS feature in Part I (upper panel), and for the PSL feature in Part II (lower panel), respectively. In both the panels the means change over *Condition* in different ways, resulting in lines having a different slope. NPS increases and decreases across *Condition*, whereas PSL slowly increases. This change of slope reveals that interaction between the variables is significant

as confirmed by the previous numeric analysis.

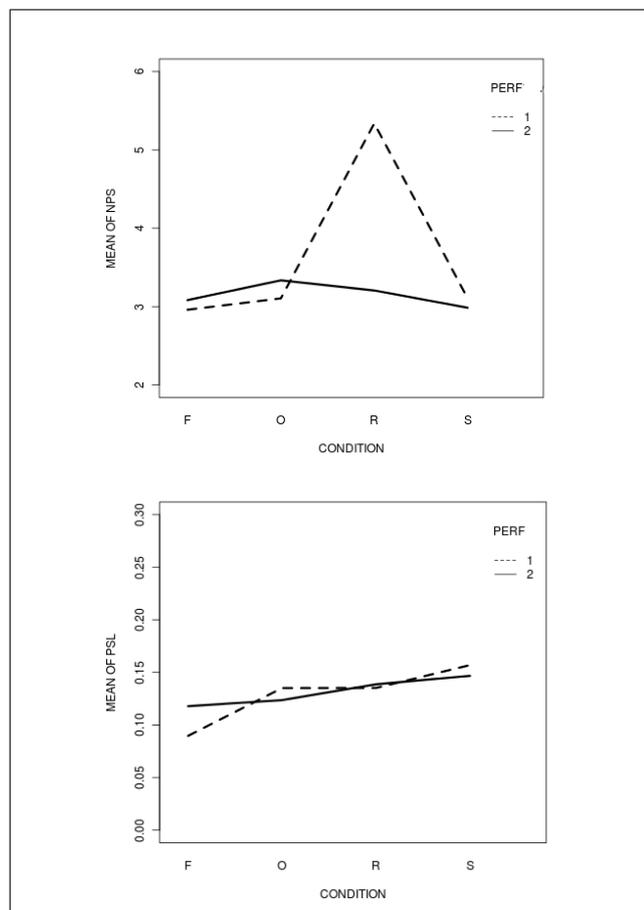


Figure 1: Upper panel: the interaction between *Condition* and *Performance* for NPS (Part I). Lower panel: the interaction between *Condition* and *Performance* for PSL (Part II). x-axis shows the *Condition* in alphabetic order: F: functional, O: over-expressive, R: regular, S: switch of score.

The analysis does not show significant between-subjects differences for each feature and for each Part. The interaction plot of Figure 2 exemplifies this in the case of feature A Part II. The variables *Instrument* and *Performance* are not significant: the lines are rather close together and there is no change over *Performance* (upper panel). In the lower panel the lines are also close together and they are parallel except for the line labeled as vio4 (vio4 is the cello player).

In conclusion, the selected features do not seem to provide enough information to distinguish among conditions. A motivation for this may be that the experiment involved a professional quartet, who is able to promptly react to possible changes and perturbations in the social interaction, so that the audio result does not fully reproduce such changes. In order to assess this hypothesis, experiments should be also performed with non-professional music players (e.g., a student quartet). Indeed, in a previous work on the same recordings, a significant effect was found on beat [10]. Results show that in some cases *Performance* is significant, i.e., there is a significant change in the audio features between the first and the second performance of the same

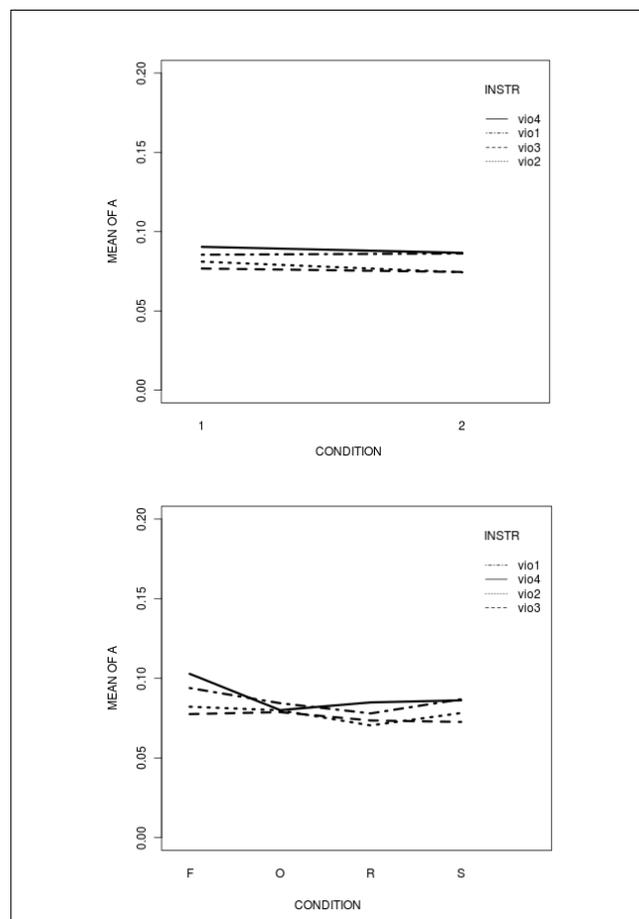


Figure 2: Upper panel: the interaction between player and performance for A; the x-axis shows the two performances. Lower panel: the interaction between players and condition for A; the x-axis shows the four conditions in alphabetic order: F: functional, O: over-expressive, R: regular, S: switch of score. The legend shows the four players: vio1 is the first violin, vio2 is the second violin, vio3 is the viola, and vio4 is the cello.

condition. This may be due to a lower stress in the players following the acquired knowledge of the specific condition (the quartet was not aware of each single condition before the experiment). However, such an effect still needs to be further investigated.

4. CONCLUSIONS

This paper addressed the identification and the analysis of score-independent audio features able to catch the social dimension of music. The selected features already proved able to distinguish performances acted with different expressive intentions. However, the carried out analysis revealed that such a set of features cannot provide sufficient information to distinguish among the different social conditions tested in this experiment. This result may either depend on the conditions, i.e., the introduced perturbations may have been too weak to produce an appreciable change in the music played by the quartet or on the features that may be unable to capture the possible variations induced

by the perturbations.

Nevertheless, the output of the work still suggests some possible perspectives for future research. One direction concerns the experiments to be performed and the conditions to be tested. A comparative analysis of professional and non-professional quartets could be carried out in order to better identify the perturbations that are likely to have a major impact on the social interaction within the ensemble. In this framework, it may be useful to have more quartets and to also use questionnaires for better assessing to what extent musical skills are related with and affect social interaction. Further possibilities include, for example, comparing musicians that usually play together and musician that do not, testing a condition where only one player is instructed to change her way to play during the performance, using less familiar music pieces making the performance more similar to improvisation.

Another direction is related to the features to be used for analysis. A deeper investigation on the possible dependence of the features from the music instrument is needed. If features are independent from the instruments, as the preliminary results obtained here seem to suggest, these may be used to improve e.g., analysis of leadership, making it less dependent on the music instrument. Moreover, as the previous work on beat analysis shows [10], the set of features can be changed/extended by including new ones that explicitly take into account the temporal dynamics and rhythmic aspects of music. Multimodal integration with motion capture data is also being investigated.

Some of the issues above are currently addressed in a study involving a string quartet of students of the Music Conservatory of Genova. New experiments are planned within the SIEMPRE Project in the near future.

Acknowledgments

This work has been partially supported by the SIEMPRE Project. The project SIEMPRE acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 250026-2. The authors thank Quartetto di Cremona and the colleagues at Casa Paganini - InfoMus.

5. REFERENCES

- [1] G. Widmer and W. Goebel, "Computational models of expressive music performance: The state of the art," *Journal of New Music Research*, vol. 33(3), 2004.
- [2] L. Mion and G. D. Poli, "Score-independent audio features for description of music expression," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16(2), 2008.
- [3] A. Camurri, G. D. Poli, A. Friberg, M. Leman, and G. Volpe, "The mega project: analysis and synthesis of multisensory expressive gesture in performing art applications," *Journal of New Music Research*, vol. 34(1), 2005.
- [4] W. Goebel, S. Dixon, G. D. Poli, A. Friberg, R. Bresin, and G. Widmer, "Sense in expressive music performance: Data acquisition, computational studies, and models," in *Sound to Sense, Sense to Sound - A State of the Art in Sound and Music Computing*, 2007.
- [5] G. Castellano, M. Mortillaro, A. Camurri, G. Volpe, and K. R. Scherer, "Automated analysis of body movement in emotionally expressive piano performances," *Music Perception*, vol. 26(2), 2008.
- [6] P. E. Keller, G. Knoblich, and B. H. Repp, "Pianists duet better when they play with themselves: On the possible role of action simulation in synchronization," *Consciousness and Cognition*, vol. 16, 2007.
- [7] S. Trehub, "The developmental origins of musicality," *Nature Neuroscience*, vol. 6(7), 2003.
- [8] M. Tomasello and M. Carpenter, "Shared intentionality," *Developmental Science*, vol. 10, 2007.
- [9] S. Koelsh, "Towards a neuronal basis of music-evoked emotions," *Trends Cogn Sci*, vol. 14(3), 2010.
- [10] G. Varni, G. Volpe, and A. Camurri, "A system for real-time multimodal analysis of nonverbal affective social interaction in user-centric media," *IEEE Transactions on Multimedia*, vol. 12(6), 2010.
- [11] D. Glowinski, P. Coletta, C. Chiorri, A. Camurri, G. Volpe, and A. Schenone, "Multi-scale entropy analysis of dominance in social creative activities," in *Proc. of the ACM Multimedia 2010 Conference (MM '10)*, 2010.
- [12] G. Varni, A. Camurri, P. Coletta, and G. Volpe, "Emotional entrainment in music performance," in *Proc. 8th IEEE International Conference on Automatic Face and Gesture Recognition (FG2008)*, 2008.
- [13] M. Leman, "Visualization and calculation of roughness of acoustical musical signals using the synchronization index model (sim)," in *Proc. of the 2000 COST G-6 Conference on Digital Audio Effects (DAFX-00)*, 2000.
- [14] M. Leman, M. Lesaffre, and K. Tanghe, "A toolbox for perception-based music analysis." Institute for Psychoacoustics and Electronic Music (IPEM), 2005.
- [15] L. V. Immerseel and J. Martens, "Pitch and voiced/unvoiced determination with an auditory model," *Journal of Acoustical Society of America*, vol. 91(6), 1992.
- [16] N. Laurenti and G. D. Poli, "A nonlinear method for stochastic spectrum estimation in the modeling of musical sounds," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15(2), 2007.
- [17] A. Klapuri, "Sound onset detection by applying psychoacoustic knowledge," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, 1999.

APPLICATIONS OF SYNCHRONIZATION IN SOUND SYNTHESIS

Martin Neukom

Institute for Computer Music and Sound Technology ICST, Zurich University of the Arts
 Baslerstrasse 30, 8048 Zurich, Switzerland
 martin.neukom@zhdk.ch

ABSTRACT

The synchronization of natural and technical periodic processes can be simulated with self-sustained oscillators. Under certain conditions, these oscillators adjust their frequency and their phase to a master oscillator or to other self-sustained oscillators. These processes can be used in sound synthesis for the tuning of non-linear oscillators, for the adjustment of the pitches of other oscillators, for the synchronization of periodic changes of any sound parameters and for the synchronization of rhythms. This paper gives a short introduction to the theory of synchronization [1, 2, 3, 4, 5], shows how to implement the differential equations which describe the self-sustained oscillators and gives some examples of musical applications. The examples are programmed as mxj~ externals for MaxMSP. The Java code samples are taken from the perform routine of these externals. The externals and Max patches can be downloaded from <http://www.icst.net/downloads>.

1. INTRODUCTION

Temporally coordinated processes are said to be synchronized. If the processes are periodic, their frequencies and phases can be coordinated. The spontaneous synchronization of machines was first described by Christiaan Huyghens (1629-1695), who observed that clocks affixed to the same support can synchronize themselves. Synchronization also plays an important role in laser technology, in neuronal nets, in chemical reactions, etc. Synchronization can be forced by any of a variety of impulse generators. More interesting is the synchronization of several non-hierarchically organized systems, which can happen with non-linear systems. Examples of synchronization by an external force are the control of cardiac activity by a pace maker and the synchronization of biological cycles through circadian rhythm. An example for the mutual synchronization of oscillating systems is the synchronization of the coordinated clapping of an audience. These systems have in common that they are not linear and that they oscillate without external excitation. They are called self-sustained oscillators. Nonlinear and chaotic oscillators have yet often been used in sound synthesis and control since the nineties [6, 7, 8] and some of them have been implemented as unit

generators (UGens) in sound synthesis languages [9] but only few literature exists on coupled nonlinear oscillators [10] and their synchronization [11].

2. SELF-SUSTAINED OSCILLATORS

Self-sustained oscillators have a natural frequency and compensate for energy loss by an inner energy source. The trajectory of the oscillation in phase space (x, \dot{x}) is a stable limit cycle. We first consider the behavior of a self-sustained oscillator from a purely qualitative point of view. Figure 1 shows the simplest limit cycle: a circle. The state of the unperturbed oscillator is described by a point rotating along the limit cycle. In a coordinate system rotating with the same angular velocity as the unperturbed oscillator, the system's state can be described as a stationary point. If the oscillator is perturbed, for example moved from state 1 to state 2, the influence of the attractor (here the circle) gradually dissipates the amplitude change, but the phase shift remains (state 3). The fact that very weak external forces can perturb the phase is one of the main reasons why self-sustaining oscillators can synchronize themselves.

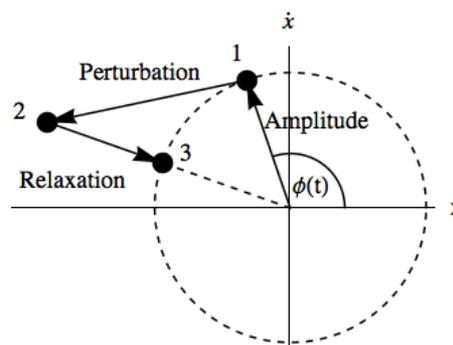


Figure 1. The influence of the attractor (here the circle) of a self-sustained oscillator gradually dissipates the amplitude change of a perturbation, but the phase shift remains.

2.1 The Van der Pol Oscillator

While attempting to explain the non-linear dynamics of vacuum tube circuits, the Dutch electrical engineer Balthasar van der Pol derived the equation

$$\ddot{x} = -\omega^2 x + \mu(1 - x^2)\dot{x} \quad (1)$$

The equation describes a linear oscillator $\ddot{x} = -\omega^2 x$ with an additional non-linear term $\mu(1 - x^2)v$. When $|x| > 1$, negative damping results, which means that energy is introduced into the system. For the following calculation we write the differential equation above as a system of two equations

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -\omega^2 x + \mu(1 - x^2)v \end{aligned} \quad (2)$$

Figure 2 shows solutions of the equation for given initial values. The limit cycle is quickly reached, regardless of the initial values. As μ increases, the limit cycle becomes more and more deformed.

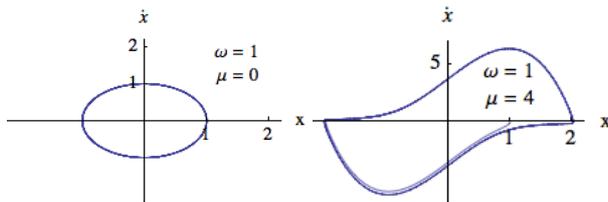


Figure 2. The limit cycle of the Van der Pol oscillator for different values of the parameter μ .

Figure 3 shows the spectrum of the oscillation for two values of μ .

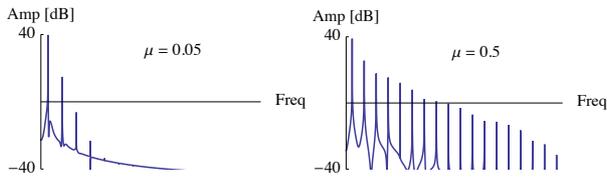


Figure 3. The spectrum of the Van der Pol oscillator becomes richer with growing parameter μ .

2.2 The Rössler Oscillator

The following homogeneous system of two linear and one non-linear differential equations can provide chaotic behavior and is named after its discoverer Otto E. Rössler:

$$\begin{aligned} \dot{x} &= -y - z \\ \dot{y} &= x + ay \\ \dot{z} &= b + xz - cz \end{aligned} \quad (3)$$

The phase space is three-dimensional. When z is small, the trajectory is close to the x - y plane and the approximations $\dot{x} = -y$ and $\dot{y} = x$ hold. Hence it follows that $\ddot{x} = a\dot{x} - x$. This equation describes an oscillation with negative damping which in the phase space is a spiral moving outward from the center. When x becomes larger than c , the third equation causes z to increase exponentially, causing the trajectory to rise quickly out of the x - y plane (see Figure 4). A large value for z makes \dot{x}

negative, so that x becomes smaller than c and the trajectory descends to the x - y plane again.

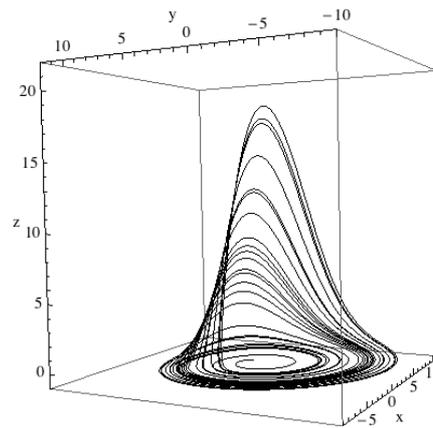


Figure 4. Chaotic trajectory of the Rössler system in state space (x, y, z) .

2.3 Implementation

There are various methods to get discrete systems, that is difference equations from differential equations, for example Euler's Method, the improved Euler's Method (or Heun's Method) or the Runge - Kutta Method. The last one is often used since it provides good results even for rather large time steps (see the SuperCollider implementations of nonlinear oscillators in [7]). With every method the results can be improved shortening the time step. In order to keep the code short in the presented MaxMSP externals we use Euler's Method and the sample period as time step. In order to implement the Van der Pol Oscillator (`mxj~ smc_v_d_pol_1`) we first calculate the acceleration a according to the differential equation above (eq. 2). Then we increment the velocity by the acceleration times dt and displacement x by velocity times dt . Taking the sample period (for sound) or frame period (for computer animations) as the time unit ($dt = 1$), we get:

```
a = (-c*x + mu*(1 - x*x)*v);
v += a;
x += (v+in[i]);
```

Where $c = \omega^2$ and $in[i]$ is the i -th sample of an excitation input buffer. Since in the Rössler system x , y and z depend on each other, we store copies of the actual values:

```
x = x0; y = y0; z = z0;
x += dt*(- y0 - z0);
y += dt*(x0 + a*y0);
z += dt*(b + (x0 - c)*z + in[i]);
```

3. SYNCHRONIZATION

3.1 Synchronization Using Periodic Excitation

First we describe a quasi-linear oscillator, that is, an oscillator with very little non-linearity. Let this non-linearity be an arbitrary function $n()$ of the state of the oscillator (x, \dot{x}) .

$$\ddot{x} = -\omega_0^2 x + n(x, \dot{x}) \quad (4)$$

The solution of the equation without the non-linear term is

$$x(t) = A \sin(\omega_0 t + \phi_0) \quad (5)$$

If we excite the oscillator with the periodic force $f(t) = \varepsilon \cos(\omega t + \phi_0^e)$, we obtain the equation

$$\ddot{x} = -\omega_0^2 x + n(x, \dot{x}) + f(t) \quad (6)$$

The instantaneous phase of the exciting force is $\phi_e = \omega t + \phi_0^e$. The frequency ω generally differs from the frequency ω_0 of the autonomous oscillator. The difference of the two frequencies $\omega - \omega_0$ is called detuning. Because a perturbation of amplitude decay rapidly, it suffices to consider the behavior of the phase. The limit cycle of the quasi-linear oscillator is a circle on which the point representing the phase rotates with the frequency ω_0 . In a coordinate system revolving with the frequency ω of the exciting force, the phase point of the oscillator has the angular velocity $\dot{\phi} - \dot{\phi}_e$ (Figure 5 left). The exciting force is represented as a vector of length ε (dotted arrows in Figure 5 right), acting at the angle $\phi^a = \phi_0^e + \pi/2$. The effect of the force depends on the phase difference $\phi - \phi_e$. At the points 1 and 2, the force acts perpendicularly to the trajectory and hence does not affect the phase. At all other points a force results that drives the points toward point 1. Hence point 1 is a stable fixed point, point 2 an unstable fixed point.

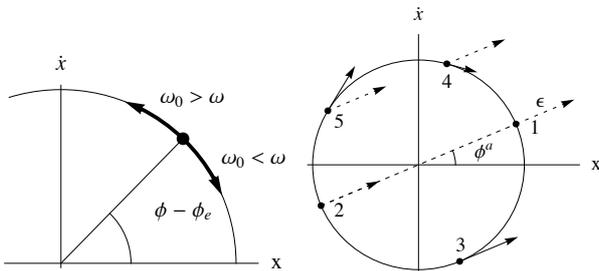


Figure 5. Coordinate system revolving with the frequency ω of the exciting force. The exciting force is represented as a constant vector. The effect of the force depends on the phase difference $\phi - \phi_e$.

If the detuning is zero, any initial phase difference between the excitation and the quasi-linear oscillator is reduced until $\phi = \phi_e - \phi^a$ and phase locking obtains

between the two. If the detuning increases (e.g. when $\omega_0 > \omega$), then two tendencies are in competition with each other: rotation (solid arrows in Figure 6 left) and the force of the excitation. The phase difference between excitation and oscillator levels off at a certain value $\Delta\phi$ (function 1 in Figure 6 right). Their movements are synchronous but not identical. If the detuning is large, the force offers insufficient resistance to the rotation. The result is a function $\Delta\phi(t)$ that remains constant for a certain time and then slips to make a quick full rotation (function 2 in Figure 6 right). The phase point starts to rotate with the so-called beat frequency Ω .

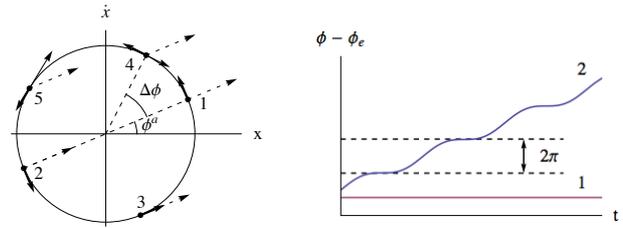


Figure 6. State space of a periodically excited quasi-linear oscillator (left figure). Phase difference $\phi - \phi_e$ versus time for an asynchronous (2) and for a synchronous (1) state (right figure).

3.2 Mutual Synchronization of Coupled Oscillators

Let us first consider two coupled limit-cycle oscillators. If we represent them as a system of first-order differential equations, we can write:

$$\begin{aligned} \dot{x}_1 &= F_1(x_1) + \varepsilon P_1(x_1, x_2) \\ \dot{x}_2 &= F_2(x_2) + \varepsilon P_2(x_1, x_2) \end{aligned} \quad (7)$$

Here the x_i are vectors, F_i and P_i are arbitrary functions and $|\varepsilon| \ll 1$. If the natural frequencies of the oscillators ω_i are approximately equal the behavior of the oscillators can be described by the following equations for their phases ϕ_i .

$$\begin{aligned} \dot{\phi}_1 &= \omega_1 + \varepsilon q_1(\phi_2 - \phi_1) \\ \dot{\phi}_2 &= \omega_2 + \varepsilon q_2(\phi_1 - \phi_2). \end{aligned} \quad (8)$$

The q_i are 2π -periodic functions. Then for the difference of the two phases $\theta = \phi_2 - \phi_1$ we have:

$$\dot{\theta} = \Delta - 2\varepsilon q(\theta) \quad (9)$$

where $\Delta = \omega_2 - \omega_1$ and $q(\theta) = q_2(\theta) - q_1(\theta)$. For there to be synchronization, the phase difference θ must be constant, that is $\dot{\theta} = 0$. Hence we have

$$q(\theta) = \Delta / 2\varepsilon. \quad (10)$$

For the simplest 2π -periodic function, the sine wave, we obtain the so-called Adler equation [1] of the first degree $\dot{\theta} = \Delta - 2\varepsilon\sin(\theta)$. $\dot{\theta}$ can only be made to disappear when $|\Delta| < 2\varepsilon$. If $|\Delta|$ becomes greater than 2ε , the coupled system begins to beat. The beat frequency Ω can be calculated from the equation $d\theta/dt = \Delta - 2\varepsilon q(\theta)$. By solving for dt and integrating dt over one period of θ , we obtain the period's duration and from it the beat frequency:

$$\Omega = 2\pi \left(\int_0^{2\pi} \frac{1}{2\varepsilon q(\theta) - \Delta} d\theta \right)^{-1}. \quad (11)$$

For $q(\theta) = \sin(\theta)$ we obtain

$$\Omega = \sqrt{\Delta^2 - 4\varepsilon^2}, \quad (12)$$

which is the same function as for the synchronization of an oscillator using periodic excitation.

4. APPLICATIONS

4.1 Single Oscillators

Sounds having rich and amplitude-dependent spectra can be produced with non-linear oscillators. The fundamental frequency of non-linear oscillators can depend on the amplitude. Figure 3 shows the spectrum of a Van der Pol oscillator for different values of the parameter μ . With increasing μ the spectrum becomes richer and simultaneously the frequency decreases. Within a broad range of the parameter μ , the frequency can be controlled by a periodic excitation. Such an oscillator can also be integrated into the Van der Pol oscillator.

The Max patch *smc_v_d_pol_1* represents a Van der Pol oscillator with a natural frequency of ω_0 and a non-linearity factor of μ . It can be excited by a sine wave of frequency ω and amplitude ε . The range of ω within which the oscillator is synchronized to the exciting frequency increases as μ and ε increase. The variation of the phase difference between excitation and oscillation, as well as the transitions between synchronous, beating and asynchronous behaviors, can be visualized by showing the sum of the excitation and the oscillation signals in a phase diagram. The screenshots of the Max patch in Figure 7 show to the upper left the waveform of the Van der Pol oscillator, to the lower left that of the excitation (amplified) and to the right the phase diagram of their sum. For these figures, the same values were always used for ω_0 , μ and ε . Comparing the figures a) and b), one sees that the oscillator adopts the exciting frequency ω within a large frequency range. When the frequency is low a), the phases of the two waves are nearly the same. Hence there is a large deflection along

the x -axis in the phase diagram showing the sum of the waveforms. When the frequency is high, the phases are nearly inverted (b) and the phase diagram shows only a small deflection. The figure c) shows the transition to asynchronous behavior. If the proportion between the natural frequency of the oscillator ω_0 and the excitation frequency ω is approximately simple, then within a certain range the frequency of the Van der Pol oscillator is synchronized so that $\omega/\omega_0 = m/n$ (with integral n and m) [5]. Here one speaks of higher order synchronization.

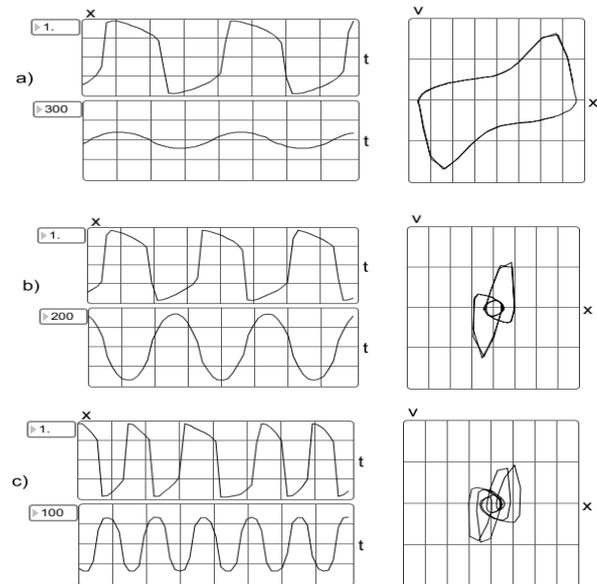


Figure 7. Waveforms of the Van der Pol oscillator (upper left) and of the excitation (lower left) and phase diagram of their sum (to the right) for different ratios between the natural frequency of the oscillator ω_0 and the excitation frequency ω .

Another possibility for controlling the frequency is to measure the frequency produced and to change the constant c until it matches the given frequency. For oscillators producing waveforms with only two zero crossings per period, we can easily measure the frequency by counting the sample periods between every second zero crossing. The following code sample is from the mxj external *smc_v_d_pol_2*, where *pcount* is a counter for the period, *dc* a correcting summand for the coefficient c , *p* the given period and *dcf* a factor for the speed with which the constant c is adjusted.

```
if(xd1*x >= 0) pcount += 1;
else{
    p = pcount + 1; pcount = 0;
    dc -= dcf*(per - 2.f*p/44100.f); }
```

Both methods result in a certain naturalness of the sound produced. Synchronization with an oscillator causes beatings and abrupt transitions to chaotic oscillations

when the frequency of the oscillator differs too much from the eigenfrequency of the Van der Pol oscillator. Controlling the frequency gives rise to continuous changes in pitch like portamento.

4.2 Mutual Synchronization of Coupled Oscillators

Only in the simplest cases can we treat the behavior of several coupled oscillators analytically [3]. Therefore, in what follows we will only describe the qualitative behavior of arrays of coupled oscillators and provide Max patches for experimentation. In the patch *smc_vdp_lin_array* N oscillators are generated by the mxj~ object *smc_vdp_lin_array* and arranged in a circle. The frequencies ω_0 and the non-linearity factors μ_0 of the individual uncoupled oscillators are uniformly randomly distributed within a range chosen by the user. An oscillator is coupled to its two neighbors by using a part of the sum of their velocities as its excitation. The following code from the mxj~ object shows how the velocity v and amplitude x are calculated for the oscillators 1 to $n-2$. The oscillators 0 and $n-1$ at the beginning and end of the array are treated separately.

```
for(int k = 1; k < n-1; k++){
    v[k] += (-c[k]*x[k]
            + mu[k]*(1 - x[k]*x[k])*v[k]
            + fb*([k-1] + v[k+1]));
    x[k] += (v[k] + in[i]);}
```

The behavior of the coupled oscillators is easy to describe for extreme values of ϵ . When ϵ is zero, the oscillators are independent and oscillate at their natural frequencies. When ϵ exceeds a certain value, the oscillators are in synchrony. The transition from complete synchrony to asynchrony as ϵ diminishes takes place through bifurcation. Figure 8 shows the frequencies of five Van der Pol oscillators as a function of their feedback factors. In this simulation over 25000 samples, the feedback is realized in the same way as in the Max patch. Over the duration of the simulation the feedback factor fb sinks from 0.0067 to 0.

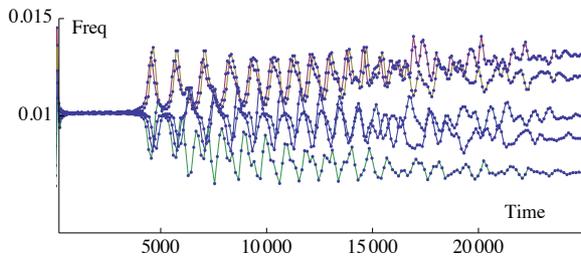


Figure 8. The transition from complete synchrony to asynchrony (for diminishing ϵ) takes place through bifurcation. The illustration shows the frequencies of five coupled Van der Pol oscillators.

4.3 Synchronizing of Chaotic Oscillators

Coupled chaotic oscillators like the Rössler oscillator (Section 2.2) can synchronize after just a few steps, even though the resulting time series is chaotic. This state is known as complete synchronization [4]. The code example below is taken from the mxj~ object *smc_roessler*, which is used in the following patches. The oscillation frequency depends essentially on the constant dt . In the examples which follow, $a = b = 0.2$ and the constant c is variable. When $c < 3$, periodic oscillation results, when $c > 3$, the periods are doubled, leading to chaotic behavior when $c \sim 4.3$. The oscillator is coupled with an external oscillator by adding the output of the latter to the velocity in z-direction.

```
x += dt*(-y0 - z0);
y += dt*(x0 - a*y0);
z += dt*(b + (x0 - c)*z + in[i]);
```

Figure 9 (Max patch *smc_roessler_1*) shows how the Rössler oscillator, within a certain range, takes on the frequency of an exciting oscillation. At the same time, it demonstrates how the influence of excitation can change originally chaotic behavior (a) into periodic oscillation (b). The figures show at the top left the oscillation of the Rössler oscillator, at the lower left the excitation (considerably enlarged) and on the right the trajectory in the phase space.

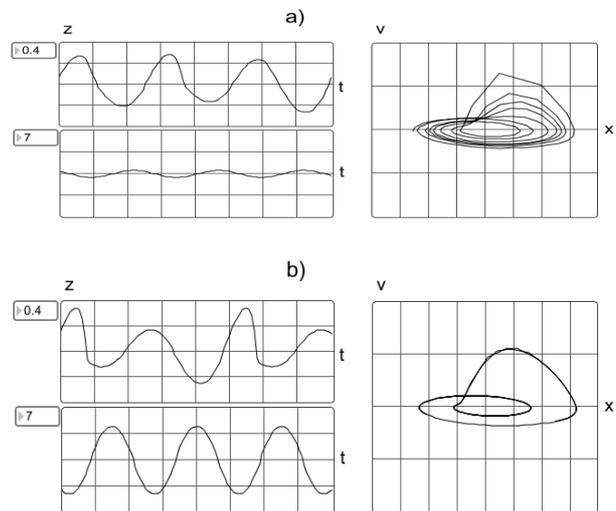


Figure 9. Waveform of the oscillation of the Rössler system (to the upper left), waveform of the excitation (at the lower left) and phase diagram of the Rössler system. Chaotic behavior a), periodic oscillation a).

The Max patch *smc_roessler_2* demonstrates how coupling two Rössler oscillators having nearly identical frequencies leads to synchronization. If the oscillators'

parameters are the same, the synchronization can be perfect.

Even the smallest differences in the initial conditions of chaotic systems lead rapidly to different trajectories. So it is astonishing that two identical uncoupled systems can be synchronized by being excited with noise. Chaotic systems can “forget”, as it were, their initial conditions (Max patch *smc_roessler_3*). Depending on the values of ω and μ , it can take a long time for the synchronization to become perfect. We can see the same behavior in non-chaotic non-linear systems. Figure 10 shows the time series of two uncoupled Van der Pol oscillators ($\omega = 0.1$, $\mu = 0.25$) excited by white noise.

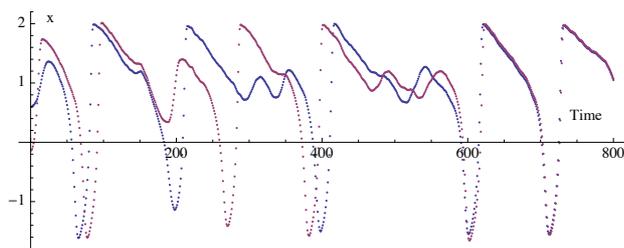


Figure 10. Time series of two uncoupled Van der Pol oscillators excited by noise.

4.4 Synchronizing Rhythms

Within one period of oscillation, many self-sustained oscillators go through a phase of slow variation and a phase of rapid variation. For example, neurons slowly build up tension and then discharge it rapidly. Oscillators of this type are called integrate-and-fire or accumulate-and-fire oscillators. Examples are the Van der Pol oscillator with a large nonlinearity or the so-called skew tent map. The `mxj~` object *smc_integrate_fire* realizes a simple oscillator by incrementing the variable x by a constant c and a random value until x is greater than 1. Then x is reset to zero, the feedback variable $sync$ is set to 1 and the discharge is indicated by a “bang”. The value of the variable $sync$ is quickly reduced.

```
x += (c + drand*Math.random());
if(x + exc > 1)
{ sync = 1.f; x = 0; outletBang(2); }
outlet(1, sync); sync*=0.9; outlet(0, x);
```

Several such objects are coupled together in the Max patch *smc_integrate_and_fire*. They produce more or less synchronous rhythms, depending on the parameter values used.

4.5 Synchronizing Any Parameter

Sine waves normally fuse to make a single sound if their frequencies are harmonic. But if they sound from different directions, do not begin at the same time or have different vibratos, they do not fuse. In order to synchronize their vibratos, not only the amplitudes and

frequencies of the vibratos must be identical, but also their relative (i.e. wrapped) phases. It is difficult to measure and control these parameters if independent oscillators are used. Even if we can adjust the frequencies, the phases normally differ. Using mutually coupled self-sustained oscillators to produce the frequencies of the vibratos we can let them synchronize just by adjust the parameter which controls the mutual coupling.

5. CONCLUSIONS

Self-sustained oscillators not only can be used to produce interesting sounds but also as a means to control parameters. Their capability to synchronize can be used to control processes which we cannot or do not want to control externally.

6. REFERENCES

- [1] A. Pikovsky, M. Rosenblum, and J. Kurths, “Synchronization”. Cambridge University Press, 2001.
- [2] G. V. Osipov, J. Kurths, and C. Zhou, “Synchronization in Oscillatory Networks”. Springer, 2007.
- [3] L. Junge, “Synchronisation interagierender komplexer Systeme”. Dissertation, Göttingen, 2000.
- [4] M. Rosenblum, A. Pikovsky, and Jürgen Kurths, “Phase Synchronization of Chaotic Oscillators”, in *Physical Review Letters*, 1996, pp. 1804–1807.
- [5] A. Balanov, N. Janson, D. Postnov, O. Sosnovtseva, “Synchronization - From Simple to Complex”. Springer, 2009.
- [6] R. Dobson, J. Fitch, “Experiments with Chaotic Oscillators”, in *Proc. of International Computer Music Conference, Banff (1995)*
- [7] X. Rodet, “Nonlinear Oscillations in Sustained Musical Instruments: Models and Control”, *Euromech, Hamburg (1993)*
- [8] M. Neukom, “Signale, Systeme und Klangsynthese”. Peter Lang, Bern, (2003)
- [9] N. Collins, “Errant Sound Synthesis”, in *Proc. of International Computer Music Conference, Belfast. (2008)*.
- [10] Georg Essl, “Circle Maps as Simple Oscillators for Complex Behavior”, in *Proc. of Conference on Digital Audio Effects, Montreal, (2006)*
- [11] A. Eldridge, “Collaborating with the behaving machine: simple adaptive dynamical systems for generative and interactive music”. PhD thesis, University of Sussex, 2007.

MELODY HARMONIZATION IN EVOLUTIONARY MUSIC USING MULTIOBJECTIVE GENETIC ALGORITHMS

Freitas, A. R. R.

Universidade Federal de Ouro Preto (UFOP)
Brazil
alandefreitas@gmail.com

Guimarães, F. G.

Universidade Federal de Minas Gerais (UFMG)
Brazil
fredericoguimaraes@ufmg.br

ABSTRACT

This paper describes a multiobjective approach for melody harmonization in evolutionary music. There are numerous methods and a myriad of results to a process of harmonization of a given melody. Some implicit rules can be extracted from musical theory, but some harmonic aspects can only be defined by preferences of a composer. Thus, a multiobjective approach may be useful to allow an evolutionary process to find a set of solutions that represent a trade-off between the rules in different objective functions. In this paper, a multiobjective evolutionary algorithm defines chord changes with differing degrees of simplicity and dissonance. While presenting such an algorithm, we discuss how to embed musical cognizance in Genetic Algorithms in a meta-level. Experiments were held and compared to human judgment of the results. The findings suggest that it is possible to devise a fitness function which reflects human intentions for harmonies.

1. INTRODUCTION

Genetic Algorithms (GAs) have a wide range of applications in Science and Engineering, in complex problems for which a specific solution is difficult to find. The search capabilities of GAs have drawn the interest from many scientific communities, including even applications in art and music [1, 2, 3, 4]. There have been many studies involving evolutionary computation and art trying to understand the possible influence of bioinspired systems on art, technology and aesthetic appreciation [3]. The use of GAs for evolving and creating art and music is a field known as evolutionary art and music [2].

Many successful applications of GAs for music analysis and synthesis can be found in the literature [5]. One of the main uses of GAs in this context is the formation of melodic lines. An algorithm for the creation of jazz solos in real-time has been proposed [6, 7], showing how the development of automated composition for a particular genre of music can be approached with GAs that consider many features of musical tasks, such as audition and improvisation. Although there have been many interesting results,

evolutionary music still faces many challenges [8] such as the evaluation of solutions.

The search for harmonies is another field of study altogether, which usually has the advantage of having fitness functions which are easier to specify. It has been shown that algorithms can create harmonies successfully [9]. Many of those works describe systems to generate four-part harmonies for a given melody (SATB harmonization) [10]. In these algorithms, a group of notes is usually generated for each original note in the melody and that is an easy problem when the chord changes are also given with the melody [11]. Analogously to methods for the generation of melodies, the development of harmonization methods focused on a particular style may be interesting [10].

When creating chord changes for pre-existing or earlier generated melodies, it is needed to define some factors that should reward or penalize an individual in its fitness computation. However, these factors are usually subjective and strongly rely on user preferences. The main idea in this paper is to present a multiobjective optimization approach for harmonization that is able to evolve harmonies while dealing with the trade-off between harmonies with or without tension¹. It is important to base composition methods on the formal aspects developed for tonal music. However, respecting some set of basic rules does not guarantee that the result will be musically meaningful or interesting. Creativity and new ideas may even follow from the violation of some rules, but the problem is knowing which set of rules to violate and when.

The results show that the multiobjective algorithm is able to find a set of solutions that represent a compromise between rules in different objective functions, leading to at least a subset of solutions that people would consider musically interesting. We also present a novel harmony representation scheme which defines a chord for each measure, or a fraction of a measure, which may generate results that differ from those of the traditional approach for SATB harmonization, in which a group of notes is defined for each note of the melody. Besides this, five notes are allowed to exist in a chord and some notes can be left unused.

In this paper, we initially present a brief introduction to GAs and give a few examples of how some of its aspects can be decided in the musical context. We then give more specific definitions of musical terms relative to harmonies

Copyright: ©2011 Freitas, A. R. R. et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

¹ Tension is created in a chord by including extra dissonant notes which create the need for relaxation or release to the listener

and how our experiments are outlined. We discuss how a multiobjective fitness function can help users with different or unknown preferences. Finally, we analyze the results of the experiments and present our conclusions.

2. GENETIC ALGORITHMS

Genetic Algorithms (GAs) are an evolutionary inspired [12] method initially proposed to solve optimization problems [13]. It is a very useful technique for hard search problems in which there is a manner to evaluate solutions but there is no known algorithm that is able to determine the optimal solution in polynomial time.

Those systems can achieve successful results for the task of generating harmonies [14] but some difficulties arise. The first of them is that there is no guarantee that the optimal solution will be found, which is understandable when a program is expected to solve such problems. The second complication is that GAs may become very computationally expensive systems when much knowledge is added to the system. This sort of knowledge is indispensable for any system expected to produce good results.

The main components of a GA are:

- A population of individuals (also chromosomes or genotypes) which represent possible solutions (phenotypes) for a problem. Those chromosomes must have some knowledge about the solution structure as the information in the chromosome maps to a phenotype which will undergo a process of natural selection. The chromosomes may represent musical information in the form of notes, frequencies or events.
- A selection process that judges the solutions according to an evaluation function (or fitness function). This operation selects which chromosomes shall have opportunity to reproduce and pass on part of its values (or genes) to the following generations. The computation of fitness for musical tasks are usually heuristic, interactive or based on rules [2]. Each of them have some assets and drawbacks.
- Genetic Operators to generate new potential solutions which will be latter tested. Those operators are structured considering information and premises about how the search for new solutions may take place. At this level, those operators must be able to combine previously selected information contained in current solutions in an effective way while having the potential to explore all possible solutions. Also, in musical systems, those operators can be guided for not generating solutions considered out of context.

Fundamental knowledge of the field must be given to the GA in order to distinguish between good and bad solutions, at least. Regarding the creation of harmonies, there must be enough knowledge embedded in the whole system to generate acceptable solutions in terms of pitches, durations and harmony itself. It is important to envisage how to

embed this sort of information not only in the fitness function but also in other features such as the representation of solutions, since the algorithm can not generate any music solution not foreseen by its representation format.

Difficulties relative to the generation of harmonies using GAs and its subsequent implications are analyzed in the following sections.

3. EVOLVING HARMONIES

3.1 Representing a solution

It is important to have solution (or harmony) representations which are convenient to map fundamental aspects of Western music, namely pitch and chords. Still considering implicit rules of Western composition, the group of notes forming a triad of a specific chord has been the basis of the process of harmonization over the last centuries. The other characteristics of chords are usually described by the degree of the scale a given note is at.

Since tuning systems of equal temperament are the mainstream in Western music, in most cases, it is handy to use the relative shift between notes for the representation of music rather than the absolute values of pitches. Thus, the final result can be normalized to represent that solution in any specific key convenient to the user. Also in the final result, the octave in which each note will be performed may be conveniently decided by the user.

There are twelve notes (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) and 7 degrees in a diatonic scale (Using C major as reference: C (1), D (2), E (3), F (4), G (5), A (6), B (7)). The degree of the remaining notes is defined in relation to the degree of its closest note (for instance, F# could be 5-, having a C major scale as reference). The interval between two consecutive notes is a semitone interval (for example, C and C#, or E and F). An interval of 2 semitones is a whole tone interval (for example, C and D, or E and F#). More information on intervals can be found in [15].

The representation must be meaningful to the specific domain, which is only the representation of harmonies in this paper. Thus, a given harmony is defined by a group of arrays, each of them defining a set of notes to be performed during that measure. This representation scheme considers a degree of granularity m of the chords in which a degree $m = 1$ means that one chord will be searched for each measure.

It is common to define a granularity for representation schemes in computer music [16] since the search space could be infinite otherwise. If the user wants more chords per measure, it is only necessary to increase the value of m . Therefore, m becomes a parameter of the algorithm or an aspect of the problem. Another solution possible for a variation in the granularity of the final result is to have a database with some chord progressions of different granularities that could be used among measures of the final result.

In this paper we propose a representation scheme for harmonies as the one in Table 1, where it is possible to perceive that the absolute frequencies of the notes are not represented. Those frequencies are defined in relation to the

| Measure | 1 | 2 | ... | n |
|----------|---|---|-----|-----|
| 5th Note | E | - | ... | D |
| 4th Note | C | D | ... | F |
| 3rd Note | G | A | ... | D |
| 2nd Note | E | F | ... | B |
| 1st Note | C | D | ... | G |

Table 1. Representation of a harmony

1st note, which is also the lowest pitched note. The octave of that note can be previously defined by the user in a convenient way for the composition. The use of 5 notes makes possible the use of a triad and two extra notes in the same chord. The 5th note is optional, as in Measure 2 of Table 1, because it could lead to many inevitable penalties during the evaluation of solutions in some specific contexts. Thus, a chord may have 4 or 5 notes.

This design avoids the generation of solutions with large vertical intervals because the 1st note is used as an anchor to decide the pitch of other notes. This codification makes the system more flexible than systems with only 4 notes (which would not be able to represent some dissonant chords) and avoids too large vertical intervals (because of the definition of the pitches in relation to each other), which would probably be treated by penalties in the fitness function if other representation approaches were used.

At the computer level, the notes can be represented by integer numbers from 1 to 12 and the codification does not have to define the classes of pitch of the notes as it does in Table 1, because a tuning system of equal temperament is used in the work. Therefore, the relation of intervals between the notes becomes more important than the absolute value of each note. The set of notes in each chord is represented by:

$$h_j = \langle \eta_{1j}, \eta_{2j}, \eta_{3j}, \eta_{4j}, \eta_{5j} \rangle, j = 1, \dots, n \quad (1)$$

with $\eta_{ij} \in \{0, \dots, 12\}$ and 0 meaning no note.

3.2 Genetic Operators

The creation of new chromosomes throughout the generations is made by carefully designed genetic operators, which actually allow an evolutionary process to occur. The most common operators are mutation and crossover, though more complex operators can be designed for a specific field of problems. These operators may be blind or guided, which means they do not always occur in a completely random manner. Guided operators can help musical applications since completely random changes in the chromosomes are not convenient in some situations. For instance, if a group of genes represents a note, the mutation can be guided to respect some harmonic or melodic rules.

Some common musical operators are used in our implementation to make the evolutionary process more efficient. Apart from the crossover, all other operators occur in the mutation phase. The probability to get into the mutation phase is 20% and each mutation operator has another prob-

ability to happen. This value is defined relatively high because some of the possible mutations that do not affect the solutions very much have greater chance of being chosen. The probability of getting into the crossover phase is 90% and only a musical crossover can happen. All probabilities of applying those operators were arbitrarily chosen by the authors based on the experience acquired after many executions of the algorithm. The musical operators used for evolving harmonies are described below.

Musical Crossover The crossover implemented here does not work at a bit level, since it could generate many random solutions. Similarly to the usual crossover, a point is chosen between the parents and they share information considering the information to the left or to the right of this point. Nevertheless, the information of a measure is inseparable and the crossover point must be between 2 measures. The selection of only 1 crossover point is convenient since it does not break many relations between measures. Using more crossover points would be very disruptive. The relation between measures is important because many fitness restrictions depend on the association between measures. The probability of a crossover is 90%.

Pitch Mutation Changes the pitch of one of the notes in a measure. The maximum change in pitch possible is 1 tone because even though some large vertical intervals may be musically acceptable, many of them lead to no musical result. A larger interval can be reached by the effect of more than one operator or even many pitch mutations but not with a single pitch mutation. Intervals of at most one tone also force the algorithm to explore more dissonant chords since a note from the triad will never be mutated to another note of the triad. Note that very large leaps are not even foreseen by the representation scheme. The probability of using this operator is 30%. If there is no note in the randomly selected position (only possible in the 5th note of a measure), the operator is not applied.

Swap between the same measure To exchange the position of notes in the same measure. That operator creates inversions in the given chord. The probability of using this operator is 50%. If a position with no note is selected, a new position is selected.

Reinitialize the chord It reinitializes all notes of a measure using a triad that includes a melody note from the corresponding measure. That makes new chords with a high possibility of being acceptable. All triads have the same probability of being chosen and the probability of using this operator is 15%.

Copy Copy the information from a measure to another measure, creating the repetition of some chords in the harmony. The probability of using this operator is 5%.

| Genetic Operators | Probability |
|----------------------|-------------|
| Crossover phase | 90% |
| 1: Musical Crossover | 100% |
| Mutation phase | 20% |
| 1: Pitch mutation | 30% |
| 2: Swap notes | 50% |
| 3: Reinitialize | 15% |
| 4: Measure Copy | 5% |

Table 2. Genetic Operators

| Measure | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|--------------|
| Notes | | | | | | | | | Total |
| C | 1 | 0 | 0 | 2 | 1 | 0 | 1 | 2 | 7 |
| C# | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 2 |
| D# | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 2 | 1 | 0 | 1 | 0 | 4 |
| F | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 2 |
| F# | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 3 | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 7 |
| G# | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 3 |
| A# | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 2 |

Table 3. Representing the problem

The probability of getting into the mutation phase was defined to 20% due to some mutations that do not alter expressively the fitness of a solution, such as the swapping of notes, which can only interfere in the condition of the position of the root note.

Table 2 summarizes the genetic operators used in this work with their respective probabilities of being applied.

3.3 Representing the problem

The problem (or the melody to be harmonized) is represented by a matrix in which a column represents a measure and twelve rows represent the possible notes in that given measure. Thus each element A_{ij} of the matrix represents the number of occurrences of the note i in the measure j . This representation scheme is exemplified in Table 3.

Summing up the values of all columns, as it is done in the column *total*, it is possible to have an idea of which scale is implicitly used in the melody. Otherwise, it is necessary to find in which measures the exceptions of a scale happen and which scales could be considered feasible to the problem.

In the example of Table 3, only notes of the C major scale are represented in column *total*, which means the implicit scale given by the melody is C major - or any relative scale, such as A minor. The scale used in a song as well as the present quantity of each note can be used later in the fitness function.

If the melody does not have enough notes to define a complete any of the predefined scales in column *total*, a scale

that contains all notes from column *total* will be chosen. On the other hand, if the melody has so many notes that every scale lacks some note to match column *total*, some measures of the melody must be considered exceptions until an exception pattern of a scale for each chord matches column *total*. In those cases, the user can also choose the scales used by the algorithm to harmonize the melody.

Only diatonic scales were considered in this experiment. In spite of that, it is easy to notice that any scale could be considered by the method.

3.4 Multiobjective harmonization

Two evaluation functions are defined in order to deal with the trade-off between dissonance and simplicity of the harmonies. That is done by considering that harmonies containing chords with only a simple triad (1st, 3rd and 5th) or a simple triad plus a 7th in some cases are good harmonies regarding the simplicity criterion. In those harmonies, a 7th is not considered a desirable note unless it is in a context in which it results in an appropriate note on the following chord, later on. The maximization of a fitness function defined as simplicity function f_1 will represent the evaluation of this sort of harmonies.

A second evaluation function to be maximized, named dissonance function f_2 , gives better fitness values to harmonies with dissonance on their chords, and measures with more notes than the triad are usually rewarded. Despite this consideration, notes besides the triad which do not fit the melody are penalized the same manner they would be in the first evaluation function.

With those functions, we define the harmonization stylistic contexts to be conciliated by the algorithm.

Perhaps, it would be possible to conciliate both objectives if we already knew the preferences of the user or if the designer becomes the user and use his own preferences to weight contradictory options. While some objectives are usually common to all harmonies, such as defining specific triads to guide the chord progression, some objectives are very contradictory according to the preferences of the user. For instance, the inclusion of dissonant notes will necessarily give different rewards and penalties to different functions.

With a multiobjective approach, the user can obtain a set of harmonies that fit that melody and represent a trade-off between the rules contained in the two objective functions. Then, the user can choose a harmony from the set of results and define the preferences while listening to the solutions.

As there are two fitness values for each solution now, we need to decide how to rank the solutions in relation to all functions. In order to do that, the solutions with the best values regarding both functions are grouped on a front which is numbered as front 1. The best solutions among the remaining ones are grouped on front 2 and so on. For solutions in the same front, the individuals are ranked accordingly to their groups and their distance from each other. This strategy is known as a Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [17] approach.

3.4.1 Evaluation of the solutions

In order to rate how good a solution is, it is necessary to define how knowledge about the subject will be built into the system. As it has been shown, much of this knowledge have already been implemented in the genetic operators but those are not able to distinguish which ones are good or bad solutions. The fitness of a solution is given depending on matters of note durations and vertical intervals in a harmony.

We need to determine some factors that may be desirable or penalized for solutions in different contexts. The relevant factors for evaluation of harmonies used in this work are described below.

- **Triads:** The triad has been the basic structure of a chord in Western Music over the last centuries. The absence of triads have a penalty of 40 in this work. If there is a 3rd but there is no 5th, the penalty reduces to 15 in the simplicity function f_1 and 5 in the dissonance function f_2 . With a triad, there are two notes left in the representation scheme that can be used to create tension.
- **Dissonant Chords:** They may be desirable depending on the context. That is what mainly justify the use of 2 evaluation functions in this work. Notes out of the triad are considered dissonant and they are penalized in the first fitness function by 10 while rewarded by 10 in the dissonance function. A 7th is not penalized in the simplicity function if it leads to another note on the following compass with a semitone interval. Two dissonant notes also lead to a penalty of 20 in both functions if they are semitone distant.
- **Invalid Pitches and Chords:** Notes which do not belong to the scales implicitly defined by the melody are penalized by 30. A note is considered invalid in the simplicity function if (i) it does not belong to the main implicit scale of the song and (ii) it does not exist in the respective measure of the melody. A note is considered invalid in the dissonance function if (i) it does not belong to the main implicit scale of the song, (ii) it does not exist in the respective measure of the melody, (iii) it does not chromatically lead to another note in the following measure and (iv) there is a note semitone distant from it in the respective measure of the melody. The penalty for this restriction is 30 as it has to be greater than the reward for dissonant chords in the dissonance function.
- **Avoid unisons:** It is not desirable to have unisons spending notes that could be used to form new chords. Unisons of the root note may be normal while unisons of the 3rd are not usually accepted. Most unisons are penalized by 5, unisons of 3rds are penalized by 10 and unisons with the root note are not penalized.
- **Sevenths:** Sevenths that lead to another note in the following chord are rewarded by 10 in the dissonance function. They are rewarded by 10 in the sim-

| Condition | Simplicity Function | Dissonance Function |
|-------------------------------------|---------------------|---------------------|
| 1: Absence of a triad | -40 | -40 |
| 1.1: Absence of 5ths | -15 | -5 |
| 2: Dissonant note | -10 | +10 |
| 2.1: Meaningful 7th | 0 | +10 |
| 2.2: Semitone dissonance | -20 | -20 |
| 3: Unisons | -5 | -5 |
| 3.1: Unison of 3rd | -10 | -10 |
| 3.2: Root note unison | 0 | 0 |
| 4: Invalid note ^a | -30 | -30 |
| 5: Meaningful sevenths ^b | +10 | +10 |
| 6: Root note position | +10 | +3 |

^a The conditions for an invalid note are different in each function.

^b The conditions for a meaningful 7th are different in each function.

Table 4. Parameters of the fitness evaluation

plicity function only if they lead to a 3rd in the following chord.

- **Tonic position:** Having some reward to chords with its root note in the 1st note, which is meant to be the lower note, may be interesting. Otherwise, there would be no difference of fitness between inverted chords and most chords would tend to be inverted. The value of the reward is 10 in the simplicity function and 3 in the dissonance function.

Other factors that could be used to influence the fitness of the harmonies in different contexts are (i) the distance of vertical intervals between the notes, (ii) the omission of specific notes, (iii) the range of the notes and (iv) stylistic predefined chord progressions. The latter would give implicit rewards to some genres of music as those sorts of rules cannot be considered in general even when considering only Western music.

Table 4 shows a brief description of the conditions of evaluation with their associated influence on the fitness function.

4. EXPERIMENTS AND RESULTS

Experiments with a population of 100 randomly initialized individuals were held with 100 executions of the algorithm. Random values between 1 and 12 are given to each note, apart from note 5 of each measure, which receives values from 0 to 12, where 0 means there is no note. The representation is made in the same way as in Table 1.

A known melody was used as the problem to give a better idea of the capabilities of the algorithm. This melody is represented in Figure 1.

To notice how the operators are influencing the execution of the GA, Figure 2 shows the influence of each condition of the fitness function in the whole population after 100 executions of the algorithm. Each group of bars shows the average influence of the restrictions for 20 generations.

The restrictions are numbered in the x axis in the same manner as they are numbered in Table 4. The values of influence consider the average influence in all individuals on



Figure 1. *Happy Birthday to You* and its harmony.



Figure 3. Solution 1.

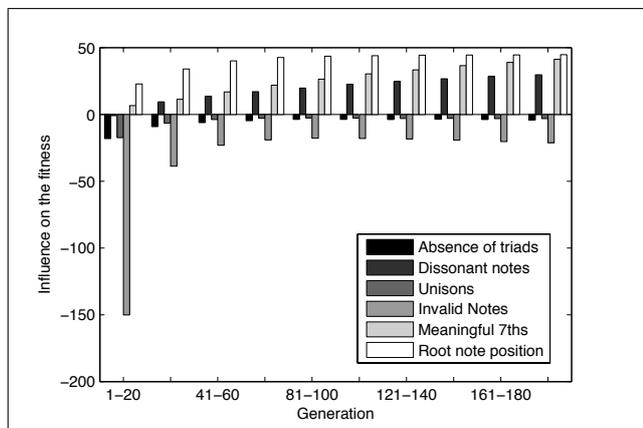


Figure 2. Fitness profile through the generations.

the respective generations, considering functions 1 and 2. It is clear that the presence of invalid notes has the greatest weight in the evaluation of the solutions. This happens mainly because of the nature of the problem and the penalties for invalid notes never get very low because two conflicting concepts of invalid notes are defined in the fitness functions. The graph was created considering 200 initial generations of this experiment.

Figure 3 shows one solution taken from the Pareto front with a good value in the simplicity function (50, 61). The generated harmony is very simple and has much similarity with the original harmony of the song represented in Figure 1, which the algorithm has no explicit access to. The only differences would be the minor chords, which are inexistent in the original song (Dm is the relative minor chord for F).

The possibility of getting harmonies similar to the original one is an evidence that the algorithm is resulting in musical solutions. However, that does not mean that a solution which does not look like the original one is not musical. Such harmonies may be found and copying or finding the original harmony of a melody is not the goal of this work. That is why “ground-truth” tests, based on the original harmony, do not apply to this algorithm as our intention is to create feasible harmonies as diverse and creative as possible.

Figure 4 shows one solution taken from the Pareto front with a good value in the dissonance function (-40, 101). It is possible to notice that the algorithm could manage to generate more complex harmonies even for a simple melody.

It is also interesting to note that this solution has a D, which does not belong to the implicit scale (Dm does). However, this D is in a measure in which there is no F,

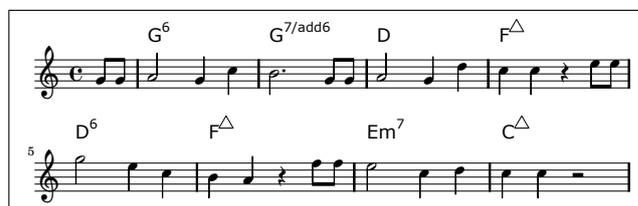


Figure 4. Solution 2.

which would be the 3rd of a Dm. Thus, the note F# in the D chord of the harmony is not penalized in the dissonance function because it chromatically leads to the following chord (F^Δ).

Also according to the determined rules, the algorithm can be considered successful since all individuals in the Pareto front of the last generations examined broke very few minor restrictions in the simplicity function or had positive values of fitness for the dissonance function.

An example of the evolution of the Pareto fronts in a single execution of the algorithm is represented of Figure 6. The lines represent the points dominated by the best solutions in a given generation while the dots represent the fitness of all solutions evaluated by the algorithm. In all executions of the algorithm, the extreme fitness values found as an answer for the simplicity function were (-310, 251) and (120, 121) while the extreme values found in the dissonance function were (70, 21) and (-60, 251). The final Pareto fronts have an average of 21.08 individuals, being the average fitness value of these individuals (4.36, 178.14).

The effectivity of the algorithm and its operators have to be measured accordingly to the quality of the set of best solutions. In order to do that, the hypervolume of the Pareto fronts can be used [18]. With this strategy, the area, volume or hypervolume dominated by the best set of solutions is used to measure the quality of the population in that respective generation. With the average hypervolume of all generations, it is possible to define the efficiency in the convergence of the algorithm.

The hypervolumes were measured with a reference point (-1200, -1200) for the dominated area. Figure 5 represents data about the hypervolumes generated throughout 200 generations (considering 100 executions). The axis x is in logarithmic scale.

The algorithm evaluated 20,000 solutions in 200 generations and the average hypervolume of the final Pareto front is 1.8197×10^6 (standard deviation 3.1243×10^4) while the initial Pareto front formed from 100 random solutions has an average hypervolume of 9.3421×10^5 . A Pareto

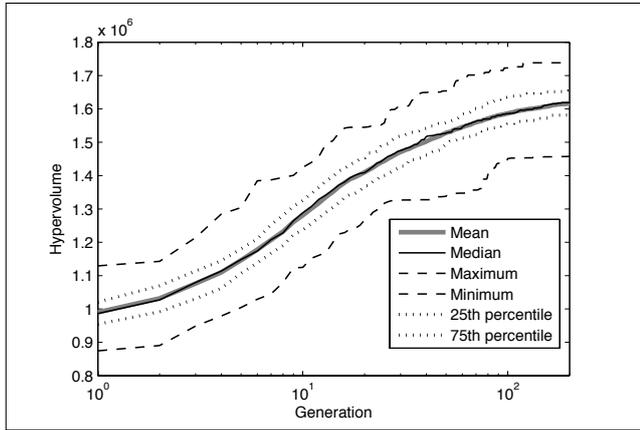


Figure 5. Evolution of the hypervolume.

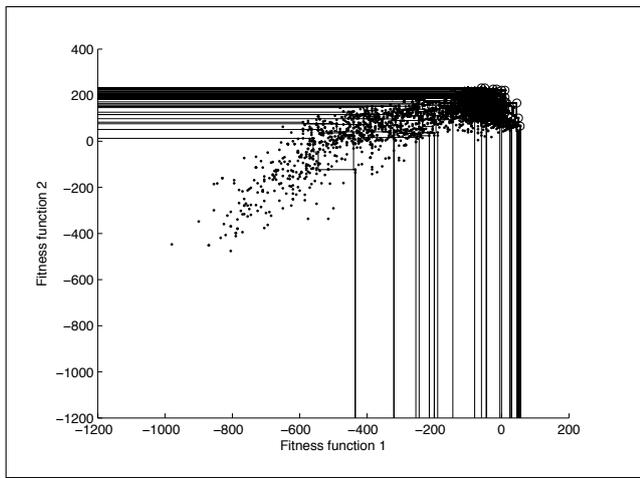


Figure 6. Evolution of the Pareto fronts.

front produced from 20,000 randomly generated solutions has an average hypervolume of 1.1833×10^6 (standard deviation 3.0844×10^4).

In order to give evidence of the reliability of the fitness functions, a survey was also done with 12 participants. They were asked to define the quality of 12 harmonies taken from the Pareto fronts of different generations.

The following idea was used to select the harmonies evaluated by the participants. The average hypervolume in the first generation is $g_0 = 9.3421 \times 10^5$ and in the last generation is $g_{200} = 1.8197 \times 10^6$. Therefore, $\Delta = g_{200} - g_0 = 8.8554 \times 10^5$. The generations with average hypervolume values closest to $g_0 + (i - 1) \frac{\Delta}{n-1}$ were used as samples in addition to generations 0 and 200, being n the number of samples and i the desired sample. This method was used because the evolution of the hypervolumes happens in logarithmic scale, as it can be seen in Figure 5. By using this method with $n = 4$, generations 0, 6, 16 and 200 (or 0, $10^{0.7782}$, $10^{1.2041}$, $10^{2.3010}$) were chosen. Three harmonies were taken from each of those generations, two of them from the extremes of the Pareto fronts, with good function 1 and 2 values, and the other one from its middle.

The participants were non-musicians and all scores given by them were normalized to a scale from 0 to 10. All harmonies were presented in a shuffled order. They were

| Generation | Harm.1 | Harm.2 | Harm.3 | Average |
|-----------------------|--------|--------|--------|---------|
| 0 | 2.2 | 1.8 | 5.0 | 3.0 |
| 6 ($10^{0.7782}$) | 4.3 | 5.1 | 3.6 | 4.3 |
| 16 ($10^{1.2041}$) | 5.1 | 5.7 | 3.0 | 4.6 |
| 200 ($10^{2.3010}$) | 8.7 | 7.7 | 7.2 | 7.9 |

Table 5. Survey about the quality of the solutions

asked to sort the harmonies by order of preference to define a score from 1 to 12 for those harmonies. The scores were defined by the rank of the respective harmony. The average results are shown in Table 5. Harmonies 1 and 3 are the best ones regarding functions 1 and 2, respectively. Harmony 2 represents a trade-off harmony.

The results indicate a small tendency of preference for the harmonies with good values in the simplicity function (Harmonies 1). However, it does not necessarily mean that any of the functions have the ability to represent the harmonization process more effectively than the other, because those numbers might be only showing preferences of the participants. In fact, this experiment can give only evidence of a well defined fitness function and not prove it as it could depend on the specific musical context.

Larger human scores for individuals in latter generations are a good sign that the fitness functions are in fact representing good harmonies. The gap between the average score in the last generation and other generations is also interesting. Harmonies which disrespected only few restrictions imposed by the fitness functions (generation 16) had a very low score (only 84% higher than completely random solutions) while the final solutions had significantly higher scores (320% higher than the initial solutions). Thus, solutions breaking few restrictions were considered almost as bad as solutions breaking many restrictions or even random solutions.

Either way, the higher the average human results are, the higher the fitness function values returned by the algorithm were. That gives some evidence that the codification of the restrictions described in this work represents at least in part successful harmonies while the system is able to work with different preferences of users. After all, these are the goals of the proposed algorithm.

5. CONCLUSION AND FUTURE WORK

The experiments with two functions to evaluate the solutions made clear important aspects of this sort of problem. The implicit knowledge of the system have great influence on the results. The way solutions which are good in relation to only one evaluation function differ from other solutions can demonstrate it very clearly. The multiobjective approach enables the algorithm to ignore particular preferences of the user and generate a set of feasible solutions.

The algorithm converged to many interesting solutions such as the one in Figure 4. This solution is the result of the flexibility built into the system which leads to another trade-off, one between diversity and obedience to certain rules. The last option is not usually useful to composers looking for new ideas. Yet systems with creativity can give

sets with many feasible solutions instead of giving always the same solution excessively based on rules.

Given the main idea of the method, readers can hopefully use their imagination to formulate their own harmonization rules and conciliate different musical contexts.

Methods that would allow the GA to have an idea of how the whole music is meant to be and get any sort of external influence would also probably lead to better contextualized approaches. Currently, most conditions of evaluation of a solution consider only the following chord, at most. It would be interesting to imagine systems that could analyze the measures as a whole before applying genetic operators.

Rule-based models can be more efficient in some specific cases of harmonization, specially when composing in a particular style. Nevertheless, evolutionary systems have clear advantages of flexibility and possibilities of creative new solutions. The multiobjective approach clearly demonstrates how these possibilities of new creative solutions can be utilized. In future work, a better definition of which aspects are preferences and which aspects are rules could lead to a creative evolutionary approach that could also implicitly include rule-based features.

Acknowledgments

This work was supported by the National Council for Research and Development (CNPq) and Coordination for the Improvement of Higher Level Personnel (CAPES, Brazil).

6. REFERENCES

- [1] D. W. Corne and P. J. Bentley, Eds., *Creative Evolutionary Systems*, ser. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann, 2001.
- [2] E. R. Miranda and J. A. Biles, *Evolutionary computer music*. Springer Verlag, 2007.
- [3] J. J. Romero and P. Machado, *The art of artificial evolution: A handbook on evolutionary art and music*, ser. Natural Computing Series. Springer, 2007.
- [4] S. Todd and W. Latham, *Evolutionary art and computers*. Orlando, FL, USA: Academic Press, 1992.
- [5] A. R. Brown, "Opportunities for evolutionary music composition," in *Proceedings of the Australasian Computer Music Conference*, Melbourne, 2002, pp. 27–34.
- [6] J. A. Biles, "GenJam: A genetic algorithm for generating jazz solos," in *Proceedings of the International Computer Music Conference*. Citeseer, 1994, pp. 131–131.
- [7] —, "GenJam: Evolution of a jazz improviser," in *Creative Evolutionary Systems*, D. W. Corne and P. J. Bentley, Eds. Morgan Kaufmann, 2001, pp. 165–188.
- [8] J. McCormack, "Open problems in evolutionary music and art," in *Lecture Notes in Computer Science, Applications on Evolutionary Computing, EvoWorkshops 2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, and EvoSTOC*, vol. 3449. Springer, 2005, pp. 428–436.
- [9] G. Papadopoulos and G. Wiggins, "Ai methods for algorithmic composition: A survey, a critical view and future prospects," in *AISB Symposium on Musical Creativity*. Citeseer, 1999, pp. 110–117.
- [10] R. McIntyre, "Bach in a box: The evolution of four part baroque harmony using the genetic algorithm," in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*. IEEE, 1994, pp. 852–857.
- [11] A. Horner and L. Ayers, "Harmonisation of musical progression with genetic algorithms," in *ICMC Proceedings 1995*, 1995, pp. 483–484.
- [12] C. Darwin, *The origin of species*. Signet Classic, 2003.
- [13] C. Reeves, "Genetic algorithms," *Handbook of Metaheuristics*, pp. 109–139, 2010.
- [14] S. Phon-Amnuaisuk, A. Tuson, and G. Wiggins, "Evolving musical harmonisation," in *Artificial neural nets and genetic algorithms: proceedings of the international conference in Portorož, Slovenia, 1999*. Springer Verlag Wien, 1999, p. 229.
- [15] M. Kennedy and J. Bourne, *The concise Oxford dictionary of music*. Oxford University Press, USA, 2004.
- [16] J. A. Biles, "Autonomous GenJam: eliminating the fitness bottleneck by eliminating fitness," in *GECCO-2001 Workshop on Non-routine Design with Evolutionary Systems*, 2001. [Online]. Available: http://sydney.edu.au/engineering/it/~josiah/gecco_workshop_biles.pdf
- [17] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Parallel Problem Solving from Nature PPSN VI*. Springer, 2000, pp. 849–858.
- [18] C. Fonseca, J. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," in *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, vol. 216, 2005.

AN ADAPTIVE CLASSIFICATION ALGORITHM FOR SEMIOTIC MUSICAL GESTURES

Nicholas Gillian

R. Benjamin Knapp

Sile O'Modhrain

Sonic Arts Research Centre

Queen's University Belfast

United Kingdom

{ngillian01,b.knapp,sile}@qub.ac.uk

ABSTRACT

This paper presents a novel machine learning algorithm that has been specifically developed for the classification of semiotic musical gestures. We demonstrate how our algorithm, called the Adaptive Naïve Bayes Classifier, can be quickly trained with a small number of training examples and then classify a set of musical gestures in a continuous stream of data that also contains non-gestural data. The algorithm also features an adaptive function that enables a trained model to slowly adapt itself as a performer refines and modifies their own gestures over, for example, the course of a rehearsal period. The paper is concluded with a study that shows a significant overall improvement in the classification abilities of the algorithm when the adaptive function is used.

1. INTRODUCTION

Musicians frequently use communicative gestures to interact with other performers live on stage when other forms of communication, such as verbal, are inappropriate. These gestures could consist of subtle looks between players in an improvisation trio or the more obvious movements of a conductor in front of an ensemble. Rime and Schiaratura [1] refer to such communicative movements as *semiotic gestures*; including symbolic hand postures such as the "OK" sign or deictic pointing gestures within this definition. This natural method of interaction is still difficult however between a musician and a computer and the objective of this work has therefore been to improve this.

To enable a computer to recognise a performer's semiotic gestures we adopted a machine learning approach in which a large data set, consisting of the recorded sensor data - or features derived from the data - from each gesture for example, are used to tune the adaptive parameters of a model or function. As outlined in the previous work by the authors [2], a key aspect in the design of the machine learning algorithms used for the recognition of musical gestures is that they need to be quickly trained with the performer's own gestural vocabulary, i.e. the relationship between a gesture and its corresponding action, using

whatever sensor best suits the user. The algorithms should not, therefore, be constrained to work with just one type of sensor, such as a mouse or camera, but should instead work with any N -dimensional signal. Further, the recognition algorithms should be designed to be rapidly trained with a low number of training examples. This would result in a fast data collection/training phase facilitating a musician to rapidly prototype a gesture-sound relationship; enabling a performer or composer to quickly validate whether such a relationship works both aesthetically and practically. For real-time musician-computer interaction, particularly in a live performance scenario, it may not be practical for a performer to be able to inform a recognition algorithm that they are currently performing a gesture (by pressing a trigger key for example). Therefore a recognition algorithm should be able to automatically calculate a classification threshold for each gesture in the model to enable real-time continuous recognition, without the user having to first train a null-model, such as a noise model in speech recognition. For the recognition of semiotic musical gestures it is also beneficial for an algorithm to be able to, after being initially trained by the performer, automatically adapt its model to provide the best classification results if the user adapts their own gestures. This is particularly useful for a musician as they might define a set of gestures to use at the start of a rehearsal session, for example, and then slowly modify and refine these gestures over the course of the rehearsal period.

To the authors knowledge, there are only a small number of examples of machine learning algorithms that are suitable for gesture recognition and can automatically adapt their own models online. Licsar and Sziranyi [3], for example, developed a vision-based hand gesture recognition system with interactive training aimed to achieve a user-independent application by on-line supervised training. Babu et. al. [4] also created an online adaptive radial basis function neural network for robust object tracking. However, both these algorithms did not fulfill the design constraints for a semiotic musical gesture classification algorithm, as the algorithm was either restricted to use just a video camera as input to the recognition system or a large number of training examples were required because of the complexity of the model being used. A novel algorithm, called the **Adaptive Naïve Bayes Classifier**, was therefore specifically developed for the recognition of semiotic musical gestures.

2. ADAPTIVE NAÏVE BAYES CLASSIFIER

The Adaptive Naïve Bayes Classifier (ANBC) is a supervised machine learning algorithm based on a simple probabilistic classifier called Naïve Bayes that itself is based on Bayes' theory and is particularly apt for the classification of musical gestures. Like a Naïve Bayes Classifier, ANBC makes a number of basic assumptions with regard to the data it is attempting to classify, most significantly that all the variables in the data are independent. However, despite these naïve assumptions, Naïve Bayes Classifiers have proved successful in many real-world classification problems [5] [6] [7] [8]. It has also been shown in an empirical study that the Naïve Bayes Classifier not only performs well with completely independent features, but also with functionally dependent features, which is surprising given the algorithm's naïve assumptions [9]. One major advantage of the ANBC algorithm for the recognition of musical gestures is that it requires a small amount of training data to estimate the parameters of each model. This is mainly due to the naïve assumption that each variable in the data is independent, as the parameters for each dimension can be computed independently and it therefore does not suffer from the 'curse of dimensionality' [10]. We have specifically updated the Naïve Bayes Classifier with an adaptive online training function along with the automatic computation of a classification threshold for each gesture in the model, making the algorithm particularly suited for the recognition of semiotic musical gestures. Prior to explaining these modifications, we first describe the algorithm's foundations.

2.1 The Naïve Bayes Classifier

The ANBC algorithm is based on the Naïve Bayes Classifier, which itself is based on Bayes' theory and gives the likelihood of event A occurring given the observation of event B:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

where $P(A)$ represents the prior probability of event A occurring and $P(B)$ is a normalising factor to ensure that all the posterior probabilities sum to 1. Using Bayes' theorem, the Naïve Bayes Classifier predicts the likelihood of gesture g_k occurring given the observation of sensor value x :

$$P(g_k|x) = \frac{P(x|g_k)P(g_k)}{\sum_{i=1}^G P(x|g_i)P(g_i)} \quad (2)$$

Note that $P(B)$, the normalising factor, has now become the summation of the likelihood of all the G gestures in the model occurring given the observation of sensor value x . In most real-world applications, $P(g_k)$, the prior probability of observing gesture k , will be equally likely for all the gestures and given by $1/G$ (in which case it could simply be ignored). Because a Naïve Bayes Classifier makes the naïve assumption that each dimension of data is independent, equation (2) can easily be extended to calculate the posterior probability of gesture g_k occurring given the

observation of the N -dimensional vector \mathbf{x} :

$$P(g_k|\mathbf{x}) = \frac{P(\mathbf{x}|g_k)P(g_k)}{\sum_{i=1}^G P(\mathbf{x}|g_i)P(g_i)} \quad (3)$$

where $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$. As each dimension is assumed to be independent, $P(\mathbf{x}|g_k)P(g_k)$, becomes:

$$P(\mathbf{x}|g_k)P(g_k) = \prod_{n=1}^N P(x_n|g_k)P(g_k) \quad (4)$$

2.2 The Gaussian Density Function

The structure of a Naïve Bayes classifier is determined by the conditional densities $P(\mathbf{x}|g_k)$ along with the prior probabilities $P(g_k)$. For the classification of musical gestures, the multivariate Gaussian density is a suitable density function to use, particularly in the instance where the feature vector \mathbf{x} for a given gesture g_k is a continuous-valued, randomly corrupted version of a single prototype vector $\boldsymbol{\mu}_k$ [8]. This is commonly the case for a static musical gesture, which will feature a specific body pose that will be slightly corrupted by both human and sensor variability, hence why the Gaussian is a good model for the actual probability distribution. Other density functions such as the Radial Basis Function, Cauchy distribution [5] or Dirichlet distribution [11] would also be suitable.

The univariate Gaussian density is specified by two parameters, its mean μ and its variance σ^2 :

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (5)$$

The multivariate Gaussian density function in N dimensions is given as:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \quad (6)$$

where \mathbf{x} is an N -dimensional column vector, $\boldsymbol{\mu}$ is an N -dimensional mean vector, $\boldsymbol{\Sigma}$ is a N -by- N covariance matrix, and $|\boldsymbol{\Sigma}|$ and $\boldsymbol{\Sigma}^{-1}$ are its determinant and inverse respectively. Using the multivariate Gaussian, $P(\mathbf{x}|g_k)$ can be replaced by:

$$P(\mathbf{x}|g_k) \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (7)$$

Instead of having to compute the determinant and inverse for each $\boldsymbol{\Sigma}_k$, the multivariate Gaussian density function can be calculated by taking the product of N independent univariate Gaussians, each with their own mean and variance values:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^2) = \prod_{n=1}^N \frac{1}{\sigma_n\sqrt{2\pi}} \exp\left(-\frac{(x_n-\mu_n)^2}{2\sigma_n^2}\right) \quad (8)$$

2.3 Adding a Weighting Coefficient

For the recognition of musical gestures, it is beneficial to add an additional weighting coefficient (ϕ_{kn}) for the n th dimension of the k th gesture. This weighting coefficient adds an important feature for the ANBC algorithm as it

enables one general classifier to be trained with a high number of multi-dimensional signals, even if a number of signals are only relevant for one particular gesture. For example, if the ANBC algorithm was used to recognise hand gestures, the weighting coefficients would enable one general classifier to recognise both left and right hand gestures independently, without the position of the left hand affecting the classification of a right handed gesture. By setting the left handed sensor dimension's weighting coefficients to 0 for any right handed gesture and the right handed sensor dimension's weighting coefficients to 1, any left handed movements will be ignored for a right handed gesture. The opposite weighting coefficient values could also be set for any left handed gesture, or for a gesture that required both hands, all the weighting coefficients could be set to 1. This simple addition of a weighting coefficient enables one general classifier to be trained for left handed, right handed and two handed gestures, rather than creating and training three individual classifiers. This weighting coefficient can either be set manually by the user or could even be set by computing the overall significance of each dimension for each particular gesture. A Gaussian model (Φ) for the k th gesture therefore consists of:

$$\Phi_k = \{\mu_k, \sigma_k^2, \phi_k\} \quad (9)$$

Equation (8) can therefore be updated with a weighting coefficient to give:

$$\mathcal{N}(\mathbf{x}|\Phi_k) = \prod_{n=1}^N \begin{cases} \text{if } \phi_n > 0, & \frac{1}{\sigma_n \sqrt{2\pi}} \exp\left(-\frac{(x_n - \mu_n)^2}{2\sigma_n^2}\right) \phi_n \\ \text{otherwise,} & 1 \end{cases} \quad (10)$$

To stop a weighting coefficient value of 0 setting the product over all dimensions to 0, regardless of the other values or weights, the current product will only be multiplied by the n th dimensional Gaussian value if the n th dimensional weight coefficient is greater than 0. If the n th dimensional weighting coefficient is equal to 0 then that dimension should be ignored and therefore 1.0 is used instead.

2.4 Real-World Computational Concerns

As the product of a large number of small probabilities can easily underflow the numerical precision of a computer, it is more practical to take the sum of the log of each weighted Gaussian rather than the product:

$$\ln \mathcal{N}(\mathbf{x}|\Phi_k) = \sum_{n=1}^N \ln \begin{cases} \text{if } \phi_n > 0, & \frac{1}{\sigma_n \sqrt{2\pi}} \exp\left(-\frac{(x_n - \mu_n)^2}{2\sigma_n^2}\right) \phi_n \\ \text{otherwise,} & 1 \end{cases} \quad (11)$$

Taking the log of the function not only stops numerically underflow, it also simplifies the subsequent mathematical analysis. Because the logarithm is a monotonically increasing function of its argument, maximization of the log function is equivalent to maximization of the function itself [10]. Like the case in equation (10), the log of the weighted Gaussian is only taken if the n th dimensional weighting coefficient is greater than 0, otherwise the log of 1 is used instead which gives 0 and therefore achieves the desired result.

2.5 Training The Gaussian Model

Using the weighted Gaussian model, the ANBC algorithm requires $G(3N)$ parameters, assuming that each of the G gestures require specific values for the N -dimensional μ_k , σ_k^2 and ϕ_k vectors. Assuming that ϕ_k is set by the user, the μ_k and σ_k^2 values can easily be calculated in a supervised learning scenario by grouping the input training data \mathbf{X} , a matrix containing M training examples each with N dimensions, into their corresponding classes. The values for μ and σ^2 of each dimension (n) for each class (k) can then be estimated by computing the mean and variance of the grouped training data for each of the respective classes:

$$\mu_{kn} = \frac{1}{M_k} \sum_{i=1}^M \mathbf{1}\{\mathbf{X}_{in}\} \quad 1 \leq k \leq G, \quad 1 \leq n \leq N \quad (12)$$

$$\sigma_{kn} = \sqrt{\frac{1}{M_k - 1} \sum_{i=1}^M \mathbf{1}\{(\mathbf{X}_{in} - \mu_{kn})^2\}} \quad 1 \leq k \leq G, \quad 1 \leq n \leq N \quad (13)$$

where M_k is the number of training examples in the k th class and $\mathbf{1}\{\cdot\}$ is the indicator bracket that gives 1 when the training label of example i equals k and 0 otherwise.

2.6 Preventing Over-Fitting

Although the Gaussian distribution is a suitable function to use when the number of training examples is small, compared with more complex distributions with a high number of parameters, it is still prone to the problem of bias. In particular, it can be shown that the maximum likelihood solution given by taking the sample mean and sample variance will commonly underestimate the true variance of a distribution [10]. This is a key example of over-fitting when a limited number of training examples are presented to the learning algorithm. The bias of the maximum likelihood solution will, however, become significantly less as the number of M_k training points increases and in the limit $M_k \rightarrow \infty$ the maximum likelihood solution for the variance equals the true variance of the distribution that generated it. A performer should therefore ensure that they do not attempt to train the ANBC algorithm with a very limited number of training example as this would cause the algorithm to severely over fit its model.

2.7 Classification Using The Gaussian Model

After the Gaussian models have been trained for each of the G classes, an unknown N -dimensional vector \mathbf{x} can be classified as one of the G classes using the *maximum a posterior probability* estimate (**MAP**). The MAP estimate classifies \mathbf{x} as the k th class that results in the maximum a posterior probability given by:

$$\arg \max_k P(g_k|\mathbf{x}) = \frac{P(\mathbf{x}|g_k)P(g_k)}{\sum_{i=1}^G P(\mathbf{x}|g_i)P(g_i)} \quad 1 \leq k \leq G \quad (14)$$

As the denominator in equation (14) is common across all gestures it can therefore be ignored without effecting the

results. If $P(g_k)$ is a constant scalar that is equal across all of the G gestures then it can also be ignored, leaving the maximum likelihood which, when using the logarithm of the weighted Gaussian model, is equivalent to:

$$\arg \max_k \ln \mathcal{N}(\mathbf{x}|\Phi_k) \quad 1 \leq k \leq G \quad (15)$$

Using equation (15), an unknown N -dimensional vector \mathbf{x} can be classified as one of the G classes from a trained ANBC model. If \mathbf{x} actually comes from an unknown distribution that has not been modeled by one of the trained classes (i.e. if it is not any of the gestures in the model) then, unfortunately, it will be incorrectly classified against the k th gesture that gives the maximum log-likelihood value. A rejection threshold, τ_k , must therefore be calculated for each of the G gestures to enable the algorithm to classify any of the G gestures from a continuous stream of data that also contains non-gestural data.

2.8 Computing a Rejection Threshold

For the rejection threshold, we desire a value that indicates how confident the classifier is in predicting that \mathbf{x} actually came from the k th distribution. In some applications it would be possible to use the normalised value resulting from Bayes' theorem and classify \mathbf{x} as class k if its prediction value was above some pre-defined value, such as 0.5. Unfortunately though, this approach will not work for the classification of a semiotic gesture in a continuous stream of data which may also contain segments of non-gestural data. Bayes' theorem cannot be used in this instance because, as $P(B|A)P(A)$ is normalised by $P(B)$, a poor prediction value when normalised may unfortunately yield a very confident prediction value, resulting in a false-positive classification error if \mathbf{x} is not a gesture.

This error can easily be mitigated however by using the log-likelihood value of the k th predicted gesture as a measure of how confident the algorithm is that \mathbf{x} is in fact gesture k . Using the log of the weighted Gaussian function as a confidence measure, a suitable rejection threshold can therefore be computed during the algorithms training phase to enable the rejection of non-gestural data in the real-time classification phase. The rejection threshold, τ_k , can be computed for each of the G gestures by taking the average confidence level of all the training data for class k minus γ standard deviations:

$$\tau_k = \mu_k^* - (\sigma_k^* \gamma) \quad (16)$$

where μ_k^* and σ_k^* are the average confidence values and standard deviation of the confidence levels respectively for the k th gesture given by:

$$\mu_k^* = \frac{1}{M_k} \sum_{i=1}^M \mathbf{1} \{ \ln \mathcal{N}(\mathbf{X}_i | \Phi_k) \} \quad (17)$$

$$\sigma_k^* = \sqrt{\frac{1}{M_k - 1} \sum_{i=1}^M \mathbf{1} \{ (\ln \mathcal{N}(\mathbf{X}_i | \Phi_k) - \mu_k^*)^2 \}} \quad (18)$$

Here γ is a constant scalar value that can be adjusted by the user until a suitable level of classification has been

achieved. The γ parameter enables the performer to further mitigate the effects of over-fitting, as by setting γ to a value greater than 1.0 will lower the threshold value and enable 'noisier' data than that in the training data set to be classified as gesture k . Using the rejection threshold, a gesture will only be classified as k if its log-likelihood estimation is greater than or equal to that classes' threshold value. Otherwise, \mathbf{x} will be classified as a null gesture, usually with an I.D. value of 0:

$$\hat{k} = \begin{cases} k & \text{if } (\ln \mathcal{N}(\mathbf{x} | \Phi_k) \geq \tau_k) \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

2.9 Adaptive Online Training

One key element of the Naïve Bayes Classifier, is that it can easily be made adaptive. Adding an adaptive online training phase to the common two-phase (training and prediction) ethos provides some significant advantages for the recognition of semiotic musical gestures. During the adaptive online training phase the algorithm will not only perform real-time predictions on the continuous stream of input data; it will also continue to train and refine the models for each gesture. This enables the performer to initially train the algorithm with a low number of training examples after which, during the adaptive online training phase, the algorithm can continue to train and refine the initial models, creating a more robust model as the number of training examples increases. The adaptive online training phase also importantly facilitates the algorithm to adapt its initial model as the performer themselves adapts and refines their own gestures; as may happen over the course of a rehearsal period for example. The adaptive online training works as follows:

After the musician has initially trained the algorithm, they can use it in real-time to classify their musical gestures. During this real-time prediction, the musician can choose to turn on the adaptive online training mode. In this mode the algorithm will slowly refine μ_k , σ_k and τ_k for each of the G gestures, overwriting the previous models that have been computed earlier. For the adaptive online training phase, the user must first decide on three parameters, the maximum training buffer size, the update rate and γ the scalar on the number of standard deviations (see equation (16)). These parameters control the maximum number of training examples to save for each class in the model, how fast the algorithm retrains the model and the number of standard deviations to use when calculating the classification threshold in the model respectively. If \mathbf{x} is classified as g_k and is greater than or equal to τ_k as determined by equation (19) then:

- Add \mathbf{x} to the training buffer, popping out the oldest training example if the buffer is full and increment the update counter by 1.
- If the update counter is equal to the update rate then recompute μ_k , σ_k and τ_k using the data in the training buffer. These are calculated using equations (12), (13), (17) and (18). Reset the update counter to 0.

Using a limited size first-in, first-out (FIFO) buffer, set by the maximum training buffer size parameter, ensures that only the most recent training examples are used to refine the models allowing the μ and σ vectors to slowly change as the user refines their own movements. Setting a fixed buffer size also ensures that an unfeasible amount of memory is not consumed by thousands of training examples over the course of a long rehearsal session. An individual FIFO buffer must be used for each of the G gestures to ensure that a large amount of new training data for one class does not ‘pop-out’ the original training data in any of the other classes. The speed at which the algorithm adapts can be controlled by the update rate parameter, allowing the performer to control how sensitive the adaption algorithm will be to their latest gestures. The overall sensitivity of the system, both for the adaptive online training phase and for the standard real-time prediction can be controlled by the performer using the γ parameter.

2.10 Real Time Implementation

The ANBC algorithm has been fully integrated into the SEC, a machine learning toolbox that has been specifically developed for musician-computer interaction [2]. The SEC is a third party toolbox consisting of a large number of machine learning algorithms that have been added to EyesWeb¹, a free open software platform that was established to support the development of real-time multimodal distributed interactive applications.

2.11 Strengths and weaknesses of the ANBC algorithm

The greatest strength of the ANBC algorithm is also, perhaps, its greatest weakness. This is the algorithm’s ability to automatically adapt its model by adding the latest classified input vector to the data that will then be used to recompute the model. In the best case this self-labelled data will help to create a more robust model, however, in the worst case a small number of incorrectly labelled training examples could create a ‘run-away’ model that becomes less effective at each update step. To mitigate this problem we have added a parameter in the EyesWeb implementation of the algorithm that enables the user to reload the original ANBC model if the real-time classification abilities of an updated model starts to perform poorly. The user can also ensure that they have set the buffer size, update rate and γ parameters to the most appropriate values. We have found through the real-time application of using the ANBC algorithm to classify semiotic musical gestures, that one of its key strengths is the algorithm’s ability to automatically compute τ_k , the classification threshold for the k th gesture. This classification threshold enables the ANBC algorithm to classify a gesture from a continuous stream of data that also contains null-gestures without having to explicitly train a null-class or tell the algorithm that one of the gestures has just been performed.

3. EVALUATING THE ANBC ALGORITHM

The adaptive classification abilities of the ANBC algorithm were tested using a simple ‘free-space’ pointing based experimental task. Participants were asked to define a number of target areas within a fixed region of space that they then had to return to when prompted. The ANBC algorithm was then used to classify if the participant’s hands were in the correct area of space when prompted; and if the classification results would improve when the adaptive function of the algorithm was used. To constrain this task as much as possible we chose not to use a musical scenario and instead used a rudimentary game orientated task. To achieve this we created a virtual boxing game called ‘Air Makoto’ in which participants were asked to strike a number of virtual targets when prompted. The ANBC algorithm, combined with a punch detection algorithm, were used to recognise if the participant was able to successfully hit the correct virtual target within a limited time scale.

3.1 Air Makoto

Air Makoto is a virtual boxing game loosely based on the martial arts training game Makoto². In Makoto, a player stands in the center of an equilateral triangle, with a six-foot tall metal column situated on each of the three corners of the triangle. Each column features ten clear panels containing lights, pressure sensors and a speaker and represents an ‘opponent’ for the player to battle with. The player uses one piece of equipment, consisting of a four foot fiber-glass pole with lightly padded ends. The objective of Makoto is for the player to continually strike the randomly appearing lights on each of the columns as fast as possible using the pole without missing any, as the computer controlling the lights monitors the player’s reaction time. As the game progresses, the interval between each new light and the amount of time it is lit decreases, with the overall objective of the game to make it to the end of the final level without missing a single panel.

Air Makoto uses a similar game design, with the exception that only two columns are used, both of which are imaginary. The player must therefore define where in space they want the columns’ target panels to be located. For simplicity, we used three target panels for each column and asked the player to ‘punch’ the air targets when prompted, rather than hitting them with a pole. Using Air Makoto, we were able to test the classification abilities of the ANBC algorithm by using it to recognise whether a player had successfully hit the corresponding target panel when prompted. A Polhemus Liberty 6-degrees of freedom magnetic tracker was used to track the participants’ movements, using custom built capturing software. The Polhemus was sampled at 120Hz using two tracking sensors, one of each mounted on the top of a small glove that each participant was asked to wear on their left and right hands. The Polhemus data was streamed directly into EyesWeb via OSC, after which the position data from both sensors was sent to the ANBC block for training/prediction, along with being sent to a hit detection algorithm to recognise the punch

¹ <http://musart.dist.unige.it/EywMain.html>

² <http://www.makoto-usa.com/new/index.html>

gestures. EyesWeb then sent the ID's of any punch gestures that were recognised via OSC to Processing³ which contained a game engine, to keep track of the participant's progress during a game, and a visual engine, that provided the participant with a 3D virtual game environment for visual feedback, as illustrated in Figure 1.

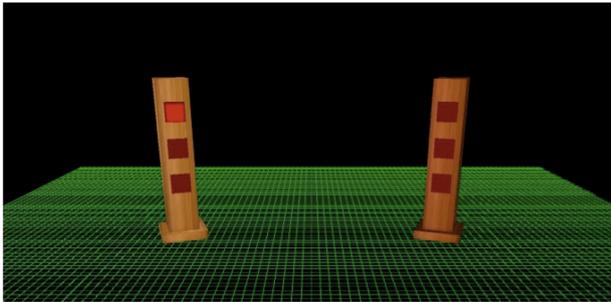


Figure 1. The Air Makoto game screen

3.2 Hit Detection

In Air Makoto, a participant was evaluated as being able to correctly hit a target panel if they made a punching gesture at the imaginary location of the correct target panel before the target panels light went out. The ANBC algorithm was used to detect whether the player's hand was in the correct target area, however, the game also required a way of detecting whether a punch gesture was made. A punch gesture was detected by taking the first derivative of the position data from the X, Y and Z axis of both sensors on the left and right hands. The position data was first low pass filtered using a moving average filter with a buffer size of 5 prior to differentiation. The differentiated signal was then passed through a dead zone block which zeroed any value between the range of -1.0 to 1.0, offsetting any value either above or below this range by -1. The output of the dead zone block was passed through a threshold crossing block that was triggered with an upwards threshold crossing above the value of 0.1. Using these signal processing techniques, illustrated in Figure 2, a robust punch detection algorithm was created as the thresholding block would only trigger an output if a negative-positive change of direction occurred in any of the three axes of either hand. If a threshold crossing was detected then EyesWeb would check to ensure that the ANBC algorithm was predicting that one of the corresponding target areas was active, sending a message to the Air Makoto game engine running in Processing to inform it of the punch.

3.3 Subjects And Setup

Twelve participants were recruited from the SARC research community via email. The sample group consisted of 9 males and 3 females with an average age of 29.3 ($\sigma = 2.96$). Six of the participants were right handed and none of the participants had any conditions that would have affected them in performing any of the movements required in this experiment. Each participant was asked to stand

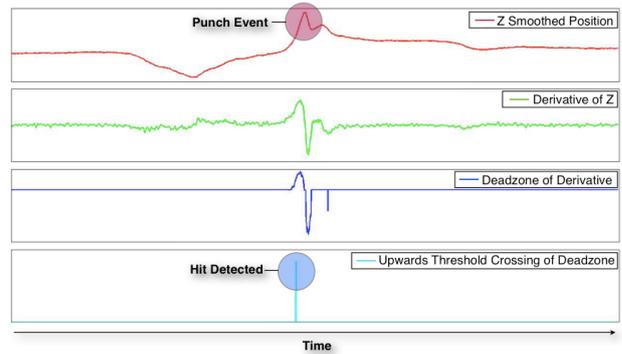


Figure 2. An example of the four main signal processing steps of the hit detection algorithm used to detect punches in the Air Makoto game. Moving from top down the four images show: z position smoothed data, first derivative of z, dead zone of the derivative signal and finally the upwards threshold detection on the dead zone signal.

on a marked location in the room and face a large projection screen situated three meters in front of them and two meters to their right. A pair of speakers was placed on either side of the screen to provide audio feedback. The projection screen displayed the Air Makoto virtual game scene, which consisted of two wooden columns placed on the left and right of the main view (as illustrated in Figure 1). Each column featured three dark red panels, which would change to bright red when the participant needed to hit them.

3.4 Method

We used a within-subject experimental design, in which each participant was asked to play the Air Makoto game in two conditions. Condition A used the ANBC algorithm without the adaptive online training mode and condition B used the ANBC algorithm with the adaptive online training mode. Prior to playing the game in either condition, each participant was given specific instructions about how to play the game and what they needed to do to train the system to recognise the location of their target panels. None of the participants were told that the ANBC algorithm was being used to recognise their gestures. The experiment was divided into three phases, with an initial data collection phase followed by a practice phase and a game phase. The practice and game phases were repeated for each of the two conditions. The order in which each participant completed the two conditions was randomised to account for any learning effects that might have occurred over the previous practice and game phases.

3.4.1 Data Collection Phase

To gather the initial training data required to train the ANBC algorithm for both conditions, each participant was asked to move their hand around the location of where they wanted to place each of the three target panels for each column. The participants were asked to only use their left hand to train and hit the three target panels on the left-most column and to only use their right hand to train and hit the three target panels on the right-most column. For the actual training stage, each target panel on the screen would light up yellow

³ <http://processing.org/>

indicating for the participant to move their respective hand to the location they wanted that target panel to be placed. The target panel would then light up red, indicating that the training data was being recorded, at which point the participant was instructed to move their hand around the location of the target panel covering a sphere with a diameter of approximately 12-inches. The size of each target area was constrained to approximately 12-inches to ensure the game would be challenging enough for the participants to play. After five seconds the training data for that target panel would stop being recorded and the next panel would light up yellow indicating that the training data for that target panel was about to be recorded. This was repeated until the training data for all of the target panels was recorded.

3.4.2 Practice Phase

The participant then entered a practice phase which lasted for one minute. In the practice phase all audio and visual feedback was turned on. For condition *A*, the original training data was simply reloaded and the adaptive training mode was turned off. For condition *B* however, the adaptive training mode was turned on during the practice phase. At no stage in the experiment could the participants see a representation of the position of their hands in the virtual world as this would have made the game too easy. However, during the practice phase an additional piece of visual feedback was provided in the form of a white square that would light up around any target panel if the participant had their hand in the location of that target panel. This visual feedback gave the participants valuable information in terms of whether they had their hands in the correct location or not. The participants also received audio feedback in the form of a punching noise if they were able to successfully ‘hit’ an illuminated target panel in the time allotted. This audio feedback informed the participants whether they were punching in the correct location and also making the correct punching gesture to trigger a hit. All of the twelve participants were able to perform the correct punching gestures, if they could remember where they had placed the target locations. The fact that each participant could correctly trigger a hit showed that there was no influence in the participants training the ANBC algorithm by moving their hand around the hit location; even though they then triggered a hit by punching this location in the practice and game phases.

3.4.3 Game Phase

After the participants had completed their one minute practice phase, they were then asked to play the main game during which their successful hit scores would be recorded. The main game lasted a total of two minutes, during which time the participants had to hit fifty randomly selected virtual target panels. To ensure the game was not too easy for the participants, each panel was only illuminated for 1.5 seconds, resulting that a participant had to react very quickly to hit the correct target panel. During the main game the participants only received the visual feedback informing them of which target panel they needed to hit. The ‘correct target area’ visual feedback and ‘correct punch

noise’ audio feedback were both turned off, resulting that the participants were unsure whether they were hitting the correct target panel in time or whether they were even punching the correct area of space at all. At the end of the two minute game the correct hit accuracy score was displayed on the screen, informing the participant how well they had performed overall during that main game. The correct hit accuracy score was calculated by awarding the participant a point for each of the 50 randomly selected virtual target panels if the participant successfully ‘hit’ the correct illuminated target panel within the 1.5 second time frame. Each participant was then given a small amount of time to rest before starting the practice phase again, only this time with a different condition being used. After the second practice phase the participant then played the main game one final time after which their scores were recorded.

3.4.4 Algorithm Settings

For this experiment, we set the maximum training buffer size parameter to 600 to ensure that the number of training examples in the initial training data set would be equal to the number of training examples used to retrain the ANBC algorithm during the practice phase in condition *B*. The update rate was set to 240, resulting in the ANBC model being recomputed every two seconds during the practice phase in condition *B*. The γ parameter was set to 5 for all conditions.

3.5 Results & Discussion

Table 1 contains the results for all twelve participants across both conditions. All of the participants, with the exception of participant eight, achieved a higher score in condition *B* which used the adaptive function compared with condition *A* which just used the training data collected in the initial data collection phase. A paired *t-test* analysis on these results showed that there was a significant overall improvement between the participants’ scores in condition *A* with that of the participants’ scores in condition *B* ($h = 1, p = 0.0028$). But why? One observation noted during the course of the study may explain these results, in that the majority of participants found it difficult to remember exactly where they had placed some or all of their target zones, even thirty seconds after they had just specified their locations. Because of this inability to locate the target zones, many of the participants had to spend the first thirty seconds of the practice phase just locating one or several of the target zones. In condition *B*, a difficult target zone slowly adapted itself until the participant found it easy to locate, with many of the participants remarking “*ah, now I remember where it is*”, unaware that the algorithm was adapting the location and size of the target zone as the participant was exploring its possible location in space. The outcome of this adaptive training resulted that, for the majority of participants, they were consistently successful at hitting the flashing target panels by the time the practice mode ended. In condition *A* however, many of the participants were still unsure of exactly where they needed to punch for one or more of the target panels by the time the practice mode ended. This observation highlights two im-

| Participant # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---------------|----|----|----|----|----|----|----|----|----|----|----|----|
| Condition A | 25 | 24 | 28 | 13 | 13 | 16 | 34 | 38 | 18 | 17 | 40 | 21 |
| Condition B | 29 | 42 | 31 | 24 | 17 | 31 | 42 | 36 | 19 | 19 | 45 | 36 |

Table 1. The results for all twelve participants for conditions A and B, with the maximum score possible in either condition of 50. The adaptive training was only used in condition B.

portant points for the application of such ‘free space’ gestures for both music and the wider HCI community. The first is a performer’s ability to remember the precise location of a point in space and the second is the importance of some form of visual or audio feedback to inform the user how far they are from any target location. For this experiment we used a world-centered frame of reference (FoR) [12], in which the user’s movements were tracked relative to the 3D space in which they were moving. The participants may have found it easier to locate the target areas if a body-centered FoR, in which the target areas were always relative to the user’s body, was used instead. A body-centered FoR may have helped the user, as a target area placed at eye-level and arms reach at the user’s right, for example, would always be at this body-centered-location irrespective of where in the room the participant moved. A body-centered FoR could have been achieved using a third tracking sensor placed on the participant’s chest, for example, from which the position coordinates of all the other sensors could be translated.

Participant eight, the only participant to achieve a better score in condition A over condition B, achieved an above average score ($\mu_A = 23.92$, $\mu_B = 30.92$) of 38 and 36 for conditions A and B respectively. A possible reason of this participant achieving a better score in condition A over condition B is that his ‘target zones’ were already optimally trained from the initial training data and the difference in score simply resulted from a better performance in condition A over condition B. Obviously, all of the participants would have achieved a higher score if they were allowed to move their hands around a much larger area of space in the initial data collection phase as this would have created a much larger ‘target zone’, enabling the participant to be less accurate. To mitigate this, we deliberately constrained the participants to only move their hands around a spherically volume with an approximate diameter of twelve inches. This constraint, combined with the speed at which the random panels appeared in the main game phase ensured that the game was difficult enough to prove a challenge to the participants. This is confirmed by the results over all participants and across both conditions as none of the participants were able to successfully hit all fifty of the targets.

4. CONCLUSION

This paper has presented the Adaptive Naïve Bayes Classifier, a novel machine learning algorithm that has been developed for the classification of semiotic musical gestures. We have shown how the algorithm can classify a set of gestures in a continuous stream of data and how the algorithm can slowly adapt itself once initially trained. The paper was concluded with a study that showed a significant overall improvement in the classification abilities of

the algorithm when the adaptive function was used.

5. REFERENCES

- [1] B. Rimé and L. Schiaratura, “Gesture and speech,” 1991.
- [2] N. Gillian, R. B. Knapp, and S. O’Modhrain, “A machine learning toolbox for musician computer interaction,” in *Proceedings of the 2011 International Conference on New Interfaces for Musical Expression (NIME11)*, 2011.
- [3] A. Licsar and T. Sziranyi, “User-adaptive hand gesture recognition system with interactive training,” *Image and Vision Computing*, vol. 23, no. 12, pp. 1102 – 1114, 2005.
- [4] R. V. Babu, S. Suresh, and A. Makur, “Online adaptive radial basis function networks for robust object tracking,” *Computer Vision and Image Understanding*, vol. 114, no. 3, pp. 297 – 310, 2010.
- [5] N. Sebe, M. S. Lew, I. Cohen, A. Garg, and T. S. Huang, “Emotion recognition using a cauchy naive bayes classifier,” *Pattern Recognition, International Conference on*, vol. 1, p. 10017, 2002.
- [6] Y. Li and R. Anderson-Sprecher, “Facies identification from well logs: A comparison of discriminant analysis and naive bayes classifier,” *Journal of Petroleum Science and Engineering*, vol. 53, no. 3-4, pp. 149 – 157, 2006.
- [7] S. Lu, D. Chiang, H. Keh, and H. Huang, “Chinese text classification by the naïve bayes classifier and the associative classifier with multiple confidence threshold values,” *Knowledge-Based Systems*, 2010.
- [8] R. Duda, P. Hart, and D. Stork, *Pattern classification*. Citeseer, 2001.
- [9] I. Rish, “An empirical study of the naive bayes classifier,” in *IJCAI-01 workshop on ”Empirical Methods in AI”*, 2001.
- [10] C. M. Bishop, *Pattern Recognition and Machine Learning*. Science and Business Media, Springer, 2006.
- [11] T.-T. Wong and L.-H. Chang, “Individual attribute prior setting methods for naive bayesian classifiers,” *Pattern Recognition*, vol. 44, no. 5, pp. 1041 – 1047, 2011.
- [12] S. O’Modhrain, “Touch and godesigning haptic feedback for a hand-held mobile device,” *BT technology journal*, vol. 22, no. 4, pp. 139–145, 2004.

AN INTERACTIVE MUSIC COMPOSITION SYSTEM BASED ON AUTONOMOUS MAINTENANCE OF MUSICAL CONSISTENCY

Tetsuro Kitahara
Nihon University
kitahara@chs.
nihon-u.ac.jp

Satoru Fukayama, Shigeki Sagayama
The University of Tokyo
{fukayama, sagayama}@
hil.t.u-tokyo.ac.jp

Haruhiro Katayose, Noriko Nagata
Kwansei Gakuin University
{katayose, nagata}@
kwansei.ac.jp

ABSTRACT

Various attempts at automatic music composition systems have been made, but they have not addressed the issue of how the user can edit a composed piece. In this paper, we propose a *human-in-the-loop* music composition system, in which the manual editing stage is integrated into the composition process. This system first generates a musical piece based on the lyric input by the user. Then, the user can edit the melody and/or chord progression. The advantage of this system is that once the user edits the melody or chord progression of the generated piece, the system can regenerate the remaining part so that this part musically matches the edited part. With this feature, users can create various melodies and arrangements and avoid the musical inconsistency between the melody and the chord progression. We confirmed that this feature facilitates the trial and error process of users who edit music.

1. INTRODUCTION

Automatic music composition (AMC) is an important task in sound and music computing, from both an academic and an industrial point of view. From an academic point of view, AMC involves constructing a computational model of human creative activities. From an industrial point of view, AMC provides a means for musically unskilled people to obtain original songs. Therefore, various researchers have developed AMC systems [1, 2, 3, 4, 5, 6].

There are two major approaches used in the existing AMC systems. The first is the fully automatic approach, in which AMC systems generate musical pieces based on the user's input, such as lyrics and styles [1, 2, 3, 4]. Because the main focus in those studies is the exploration of new models and/or algorithms for creating musically superior or novel melodies, they do not address the issue of what the system should do when the generated melody does not match that desired or expected by the user.

The second is a semi-automatic approach based on interactive evolutionary computation [5, 6]. The systems based on this approach run iterations of automatic generation of a musical piece and a user's evaluation of the generated piece. The merit of this approach is that it does not require the users to have musical skills because all they need to do is to judge whether the generated piece is good. In prac-

tice, however, this approach can impose an excessive burden on users because they have to repeatedly listen to and evaluate system-generated pieces (sometimes thousands of times). In addition, if they want to partly modify the generated piece, they cannot specify which part of the generated piece should be regenerated and how.

The common problem with these studies is that, even if the generated piece is different from what the user wants, the user cannot specify to the system what should differ in the generated piece and how, so that the system can regenerate it¹. This is an important problem because it is almost impossible for AMC systems to generate pieces that perfectly match users' desires at the first attempt. When the generated piece is different from what the user wants, the most common solution is for the users to edit the piece themselves using commercial software such as music sequencers or digital audio workstations. It is, however, not easy for unskilled people to appropriately edit a generated piece of music using such software because musical pieces in general consist of multiple voices, each of which could produce inharmonic tones if inappropriately edited. We attribute this problem to the unidirectional nature of the composition process: from automatic generation to manual editing.

In this paper, we propose an AMC system, called *OrpheusBB*, in which the manual editing stage is integrated into the iterative composition process. OrpheusBB allows users to edit the melody and chord progression of a generated piece after the first automatic generation. Once the user edits part of the melody or chord progression, the system immediately regenerates the rest of the piece. By repeating such editing, users can elaborate upon the piece without considering the possibility that the melody and the chord progression may become musically inconsistent (typically inharmonic). This approach of iterative composition involving a manual editing stage is called the *human-in-the-loop* approach. The technical issue in achieving this system is how to estimate a melody and chord progression that are musically consistent with the edited part in real time. We call this *autonomous maintenance of musical consistency* (AMMC) and achieve it using a Bayesian network.

2. SYSTEM DESIGN

In the human-in-the-loop approach, music composition is regarded as an iterative process of automatic music gener-

¹ Roads also discussed human-system interactions in music composition from a similar point of view, stating that "totally automated composition programs demand little in the way of creativity" [7].

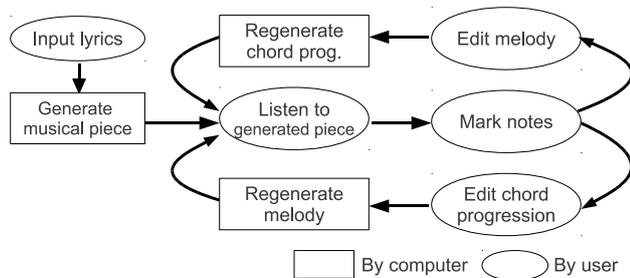


Figure 1. Iterative process in OrpheusBB.

ation and manual music editing (Figure 1). This approach was inspired by the collaboration between a creator and his assistants. A creator (such as a musician) sometimes creates his works in collaboration with assistants. In this case, the creator determines the key concept of the work and creates the important parts alone, but may leave the remaining parts to the assistants. Then, the assistants create the remaining parts using their professional knowledge and experience so that they are consistent with the creator-made parts. The ideal goal of the human-in-the-loop approach is to develop such a sophisticated music composition assistant.

Human-in-the-loop AMC systems should satisfy the following three requirements:

1. *Direct edit*

If users want to edit a specific part of the melody and/or chord progression, they should be allowed to directly edit the part that they want to edit.

2. *Priority on users' editing*

Users' editing should have priority. In other words, a mechanism should be provided to avoid overwriting the part edited by the user.

3. *Autonomous maintenance of musical consistency*

If the user's editing causes musical inconsistency, the system should correct the remaining part to maintain musical consistency. For example, if a user's melody editing introduces a disharmony with the chord progression, the system should correct the chord progression.

OrpheusBB first generates a musical piece based on the lyrics input by the user. After listening to the generated piece, the user can edit any note in the melody and/or any chord in the chord progression (satisfying Requirement 1). When the user edits the melody, the system immediately updates the chord progression to best match the edited melody (satisfying Requirement 3). Similarly, the melody is updated if the chord progression is edited. The user can mark certain manually edited or automatically (re)generated notes if they like those notes. The marked notes will not be overwritten by the update caused by subsequent editing, even if the marked and edited notes cause musical inconsistency (satisfying Requirement 2).

2.1 Initial Input as the Seed of Composition

The initial input of the user, as a seed of the composition, is an important issue because there are an infinite number of musically appropriate melodies. This initial input should efficiently constrain the variety of melodies available but should not require musical skill for the input. If the target music is limited to Japanese songs, lyrics can be used as the initial input, because Japanese is a pitch-accent language [4]. Given a Japanese sentence, its natural prosody in the standard Japanese pronunciation style is uniquely determined, and the pitch motion in the prosody can be used as a constraint for creating a melody. In fact, making a melody following the prosody of the lyrics is well known as a method for Japanese song composition. For this reason, lyrics are adopted as initial inputs in OrpheusBB.

2.2 Achieving Autonomous Maintenance of Musical Consistency

The main technical issue in achieving the above-mentioned three requirements is how to achieve autonomous maintenance of musical consistency (AMMC). The key point in achieving AMMC is determining what type of knowledge representation framework should be used for describing the mutual dependency of the melodies and chord progressions. The knowledge representation framework to be used here should provide a mechanism for inferring the optimal values for the remaining part when part of the framework is updated (a new value is observed). One possible framework for such inference may be the truth maintenance system (TMS) [8]. It is, however, unsuitable for music because TMSs are based on logical reasoning. Another available inference framework is the Bayesian network. A Bayesian network is a probabilistic graphical model in which the dependencies between random events are represented via a directed acyclic graph. Because the dependency between melodies and chord progression can be probabilistically described (e.g., the C# note is rarely used in melodies under the C chord), it is suitable for domains, such as music, for which appropriateness cannot be logically determined.

In this study, we describe the dependencies between the prosody of lyrics, a melody, a chord progression, and chord voicings (each is called a *layer*) in a dynamic Bayesian network (DBN), as shown in Figure 2. This network represents two types of dependencies: the sequential dependency within each layer and the simultaneous dependency between different layers. The network should ideally have arcs between the melody nodes and the voicing nodes [9], but these are omitted to reduce the computational cost. To use this network, the system has to know the temporal correspondence between the nodes of the melody layer and those of the chord progression layer. We therefore put the following restrictions on the melody and chord progression:

- The rhythm of the melody is determined in the initial generation stage and cannot be modified in the subsequent manual editing stage.
- The chords change every two beats (half measure).

Once music knowledge has been represented in this

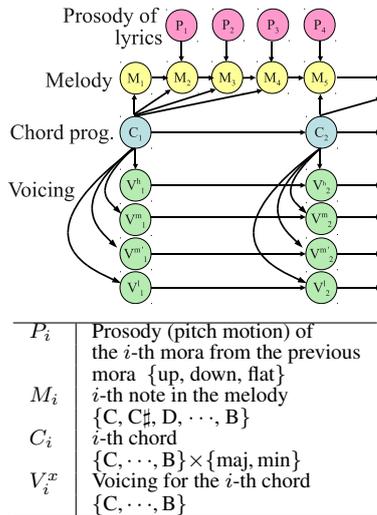


Figure 2. Dynamic Bayesian network used in our system.



Figure 3. Initial input window.

DBN, AMMC is reduced to the problem of inferring the most likely values for the remaining nodes after updating the value for the node corresponding to the edited note/chord. Marking a note is reduced to giving a probability of 1.0 to the current value for the node corresponding to the marked note. AMMC is therefore achieved through the following steps:

1. If the user marks a note, the probability of the current value for the corresponding node is set to 1.0.
2. If the user edits a note or chord, the value for the corresponding node is updated, and then, the probability inference is determined.
3. After determining the probability inference, the music data (melody, chord progression, and chord voicings) are updated to the values with the highest likelihood, and the editing window is updated for further editing.

3. IMPLEMENTATION

We implemented OrpheusBB based on the design described above.

3.1 Graphical User Interface

3.1.1 Initial Input Window (Figure 3)

Once OrpheusBB is launched, the initial input window appears. The user inputs Japanese lyrics with both *kanji* and *kana*. If necessary, the user can change the chord progression and/or the rhythm pattern of the melody. Once

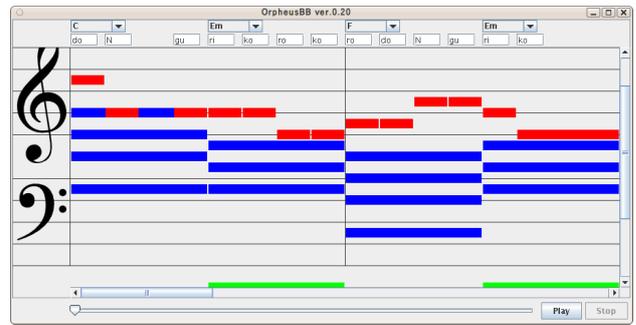


Figure 4. Music editing window.

the “Generate” button is pressed, the system generates a melody that has the same pitch motion as the prosody of the input lyrics, and proceeds to the music editing stage.

3.1.2 Music Editing Window (Figure 4)

Once it has generated a musical piece, OrpheusBB displays a pianoroll-like window, where the user can edit the melody, chord progression, and chord voicings in the generated piece. If one or more components are edited by the user, the remaining elements are immediately updated based on the aforementioned AMMC function to avoid producing inharmonic tones in the edited melody or chord progression. Specifically, the chord progression and its voicings are updated when the melody is edited, and the melody and chord voicings are updated when chords are changed.

The user can mark notes in the melody and/or chord voicings to prevent those notes from being changed by AMMC in subsequent editing against the user’s will (Figure 5 Left). In the typical situation for this note marking function, the number of marked notes gradually increases with every iteration of the manual editing and automatic regeneration process. As the number of marked notes increases, the musical piece is expected to become closer to the user’s desires. Thus, the user elaborates upon the piece through the iterative action of (1) editing notes/chords, (2) listening to the edited and automatically updated notes, and (3) marking them if desired.

The chord candidates are represented by chord names, such as *C* or *Dm*; however, the selection of an appropriate chord is not easy for people who are not familiar with this notation. We therefore implemented a function for assisting such people in selecting chords, where the likelihoods of the chord candidates are represented in grayscale (Figure 5 Right). If a chord has a high likelihood under the current context (the melody and neighboring chords), the chord name appears in dark gray (or almost black) in the dropdown list. If a chord has a low likelihood, on the other hand, the chord name appears in light gray (or almost white). Using this function, the user can select a chord that is expected to match the current melody from the dark gray candidates. In addition, the user can select a light gray chord to change the impression because the light gray chords are expected to have largely different impressions. When a light gray chord is selected, the melody is regenerated in most cases.

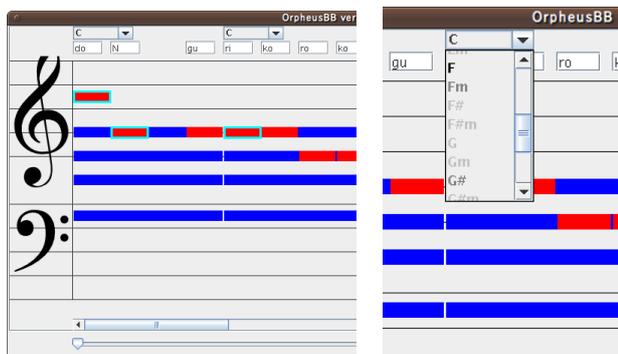


Figure 5. Left: marked notes. Right: a dropdown list for selecting a chord where the chords with high likelihoods are displayed in dark gray and the chords with low likelihoods are displayed in light gray.

3.2 Generation of a Melody from Lyrics

Given a set of lyrics, a melody is generated using the method adopted in Orpheus [4]. The input lyrics are first analyzed by the front-end module of a Japanese text-to-speech engine to identify the pronunciation (*yomi*) and prosody (pitch motion) of the lyrics. Then, the rhythm of the melody is determined. Under the constraint that the melody must have the same number of notes as the number of morae of the lyrics, the system generates the rhythm by dividing (e.g., from one quarter note to two eighth notes) or merging (e.g., from two quarter notes to one half note) notes in the rhythm pattern selected by the user as needed. The notes in the pattern that should be preferentially divided or merged are defined by a tree structure called a *rhythm tree* [4].

After constructing the rhythm, the pitch (note number) of each note in the melody is determined so that the pitch motion matches the prosody of the lyrics and is also musically appropriate. Based on the emission probability of each pitch and the transition probability from each pitch to each pitch, which are manually defined in advance, the note sequence that has the highest probability is searched for using the dynamic programming search method [4].

3.3 Autonomous Maintenance of Musical Consistency

The autonomous maintenance of musical consistency (AMMC) is achieved based on the DBN described in Section 2. Because implementation of the DBN shown in Figure 2 is not practical due to the computational cost and the amount of training data required, we simplify the DBN as follows:

- When the melody is edited, the system infers only the chord nodes, considering all melody nodes to be given. In this case, the DBN is equivalent to a hidden Markov model, so the Viterbi algorithm can be used to infer the chord nodes.
- When a chord is changed, the system infers only the melody nodes, considering all chord and lyric nodes to be given. In this case, inference of the melody nodes is equivalent to an optimal path search performed in Orpheus for melody generation.



Figure 6. The generated piece given the lyrics “*Donguri korokoro donguriko*.” The latter part “*korokoro donguriko*” was repeated three times because the input lyrics are too short for a eight-measure piece.

- If a chord is changed by the user or system, only the voicing of the changed chord is inferred.

The conditional probabilities for all nodes are experimentally defined based on conventional music theory.

4. EXAMPLE OF USE

In this section, we present an example of the composition of a musical piece using this system as a proof of concept. In particular, we will confirm here that:

- (1) chords are automatically changed when notes in the melody are changed to notes producing disharmony with the current chord progression,
- (2) the melody is automatically regenerated when edited chords produce disharmony with the current melody,
- (3) the user can prevent the system from overwriting the notes that they like by marking such notes, and
- (4) the coloring of the chord names in the chord selection lists helps the user select chords.

A user executed music generation with the lyrics “*Donguri korokoro donguriko*”, obtaining the piece shown in Figure 6. He then changed “E–E–G–G” in the melody in the second measure to “F–F–A–A”, which results in disharmony with the current chord “Em”. Then, the chord was automatically changed to “F”, which matches the edited melody (Figure 7). The neighboring chords were also changed. He marked the edited and automatically updated notes to prevent those notes from being overwritten.

Next, he changed the chord progression of the last two measures (“F–G–C–C”) to “F–Ab–G–C”) to change the impression of the ending. He selected “Ab” because this chord may give listeners the feeling of oddness as it is rarely used in this context, according to the color (light gray) of the chord name in the dropdown list. Accordingly, the melody was regenerated (Figure 8), and he marked those notes.

The unmarked notes of the first two measures in the melody were, however, automatically changed. This is be-

Figure 7. The result of changing “E–E–G–G” in the second measure of Figure 6 to “F–F–A–A”. The corresponding chord was automatically changed to “F”.

Figure 8. The result of changing the last two-measure chord progression in Figure 7 to “F–Ab–G–C”. The melody was automatically regenerated.

cause, in the current implementation, any unmarked notes in the melody may be updated when a chord is changed. Of the updated melody, he changed the notes “B” in the first measure and “A” in the second measure to “C” and “G”, respectively. Accordingly, the second chord in the first measure was automatically changed from “Em” to “Am”. The chords in the second measure were not changed because they did not cause disharmony with the changed melody.

Finally, he changed the last “G” note in the last measure to “A”. The voicing notes for the corresponding chord were not changed because these notes were marked. The complete piece is shown in Figure 9.

Thus we confirmed that the four aims mentioned in the beginning of this section were achieved.

5. DISCUSSION

5.1 Results of Trial Use

Through the trial use reported in the previous section, we confirmed that users can compose musical pieces based

Figure 9. The complete musical piece.

on our composition model, where a musical piece gradually becomes closer to the user’s desires by repeating the four steps of (1) listening to the system-generated piece, (2) marking the notes that the user likes, (3) editing the notes or chords that the user does not like, and (4) having the system update the rest according to the user’s editing.

In addition, the coloring of the chord names in the chord selection lists was effective at assisting the user in selecting chords. It is well known that the use of non-diatonic chords is effective in making an impressive chord progression, but it is not easy to use these chords with a conventional music sequencer if the user does not have a knowledge of harmony theory. With the chord coloring of our system, users can simply select a chord from the light gray chord names. Because the melody is automatically regenerated, the users do not have to consider the mismatch between the melody and chord progression when selecting a chord.

However, the following areas for improvement were revealed in the trial:

- **Conditional probability table (CPT)**

To make the system behavior clearly understandable, we adopted an extreme CPT: the probabilities of non-diatonic chords and the conditional probabilities of non-chord tones are very close to zero. For this reason, all melody notes under the Ab chord were C. The exploration of more appropriate CPTs is an important future issue. The training of CPTs from existing music data should also be investigated.

- **Over-updating**

Whereas AMMC worked well overall, some notes were automatically changed, even though they did not produce inharmonic tones. Because this phenomenon may negatively affect the user’s trial and error process, it should be avoided by, for example, establishing a threshold for overwriting: if the likelihood for the current value is higher than the threshold, the current value should not be overwritten to the most likely value, even if it is not the most likely value.

- **Undo function**

The user wanted to revert his editing several times but

unable to do so because the current implementation does not have an undo function. Implementing an undo function to facilitate further users' trial and error will be important in our future work.

5.2 Directability

The recent development of machine learning technologies and music corpora has facilitate great advances in the automation of music generation such as music composition, music arrangement, and musical performance rendering. Needless to say, such automation technologies are important in achieving an environment that enables musically unskilled people to create music. Most existing studies, however, have neglected the issue of allowing users to modify the automatically generated content.

Automation technologies should be developed by considering how the technologies are (or should be) used by users. From this point of view, we recently introduced the concept of *directability* [10], which indicates the controllability of content at an appropriate abstraction level. With conventional tools, such as music sequencers, creators have to directly edit all of the components of the content (e.g., all of the notes in the music). The goal of directability is to achieve intuitive editing by editing a structure-level representation.

Although users edit musical content at the note level in OrpheusBB, the editing is immediately reflected in the music structure described in a DBN. OrpheusBB can therefore be considered to be a directable user interface for editing music.

5.3 Use for Further Advanced Arrangement

AMMC would be more effective if it were applied to more advanced arrangements. When a walking bass line is used in the bass part, appropriate passing notes are carefully determined to smoothly connect chord to chord. If a chord is changed, the passing notes around the chord (especially before the chord) in the bass line should also be appropriately changed. By adding the bass line layer to our DBN, we can achieve autonomous maintenance of the walking bass line: once a chord is changed, the passing notes around the chord in the bass line are automatically modified. In actual performance situations, the detailed arrangement, such as passing notes in a bass line, is often left to each player, whereas the overall arrangement, such as the chord progression, is determined by the arranger. Similarly, users can focus on the overall arrangement, being freed from the detailed arrangement, by using AMMC.

6. CONCLUSION

Automatic music composition (AMC) systems, rather than simply generating a musical piece, should provide an environment that enables users to elaborate upon music by repeated manual editing with the assistance of AMC technology. Based on this belief, we developed a human-in-the-loop AMC system, *OrpheusBB*. The main advantage of this system is that, when the melody or chord progression is edited by the user, it can automatically regenerate the remaining part to maintain musical consistency between

the edited part and the remaining part. We call this feature *autonomous maintenance of musical consistency*, and achieved it by using probabilistic inference based on a dynamic Bayesian network.

We have some future plan. First, we will conduct quantitative evaluations to more thoroughly explore the effectiveness of this system. Second, we plan to improve the graphical user interface for editing music data. The current user interface is not easily used by people who are unfamiliar with notewise editing in a pianoroll display. We are therefore currently developing a user interface that enables both notewise and non-notewise editing. Third, we plan to extend our DBN to achieve a more advanced arrangement, as discussed in Section 5.3.

Acknowledgments

This research was partially supported by CREST, JST, Japan. OrpheusBB was implemented in collaboration with Mr. Naoyuki Totani and Mr. Ryosuke Tokunami (Kwansei Gakuin University).

7. REFERENCES

- [1] L. Hiller and L. Isaacson, "Musical composition with a high-speed digital computer," *Journal of Audio Engineering Society*, 1958.
- [2] C. Ames and M. Domino, "Cybernetic composer: An overview," in *Understanding Music with AI*, M. Balaban, K. Ebcioglu, and O. Laske, Eds. AAAI Press, 1992, pp. 186–205.
- [3] D. Cope, *Computers and Musical Style*. Oxford University Press, 1991.
- [4] S. Fukayama, K. Nakatsuma, S. Sako, T. Nishimoto, and S. Sagayama, "Automatic song composition from the lyrics exploiting prosody of the japanese language," in *Proc. Sound and Music Computing*, 2010.
- [5] D. Ando, P. Dahlstedt, M. G. Nordaxhl, and H. Iba, "Computer aided composition by means of interactive gp," in *ICMC 2006*, 2006, pp. 254–257.
- [6] J. A. Biles, "Genjam: A genetic algorithm for generating jazz solos," in *Proc. ICMC*, 1994.
- [7] C. Roads, *The Computer Music Tutorial*. MIT Press, 1996.
- [8] J. Doyle, "A truth maintenance system," *Artificial Intelligence*, vol. 12, no. 3, pp. 251–272, 1979.
- [9] T. Kitahara, M. Katsura, H. Katayose, and N. Nagata, "Computational model for automatic chord voicing based on bayesian network," in *Proc. ICMPC 2008*, 2008, pp. 395–398.
- [10] M. Hashida and H. Katayose, "Mixtract: A directable musical expression system," in *Proc. ACII 2009*, 2009.

A LEARNING INTERFACE FOR NOVICE GUITAR PLAYERS USING VIBROTACTILE STIMULATION

Marcello Giordano and Marcelo M. Wanderley

Input Devices and Music Interaction Laboratory,
Centre for Interdisciplinary Research in Music Media and Technology,
McGill University,
Montréal, QC, Canada

marcello.giordano@mail.mcgill.ca, marcelo.wanderley@mcgill.ca

ABSTRACT

This paper presents a full-body vibrotactile display that can be used as a tool to help learning music performance. The system is composed of 10 vibrotactile actuators placed on different positions of the body as well as an extended and modified version of a software tool for generating tactile events, the Fast Afferent/Slow Afferent (FA/SA) application. We carried out initial tests of the system in the context of enhancing the learning process of novice guitar players. In these tests we asked the performers to play the guitar part over a drum and bass-line base track, either heard or felt by the performers through headphones and the tactile display they were wearing. Results show that it is possible to accurately render the representation of the audio track in the tactile channel only, therefore reducing the cognitive load in the auditory channel.

1. INTRODUCTION

It has been shown in many experiments during the last twenty years (which we will present and discuss in detail in Section 2) that tactile sensation is a crucial component in the exploitation of the haptic channel and in the knowledge of our surrounding environment. Many of these experiments had music and musical interaction as their main focus, studying the role of tactile sensation and vibrotactile feedback, trying to understand to which extent the tactile sensation is crucial in the *embodiment* of the instrument by the performer or in the perception of music by an audience (see [1] and [2] respectively for example).

In both directions, what lies behind the development of this field of research is the understanding of the physiological and neurological mechanisms which guide the tactile perception. None of the projects we have described would have been realized without the fundamental work of scientist like R.T. Verrillo, who at Syracuse University brought on a series of experiments which spanned through

thirty years, trying to identify the processes and the receptors that generate tactile sensation. Verrillo gave, for the first time, a complete picture of how this complex system works, and a large amount of technical data which is essential to design any kind of prototype in this field ([3]).

Others pioneering works in this sense are those by P. Bachy-Rita, who in the '60s developed one of the first tactile displays to perform sensory substitution. His *Tactile Vision Sensory Substitution* device consisted of a chair embedded with moving rows of pins to draw on the back of the subjects images received by an external camera ([4]).

2. PREVIOUS WORKS ON VIBROTACTILE STIMULATION

The existing work about vibrotactile stimulation in musical interaction can be loosely divided into two categories according to which side of the instrument we decide to take in consideration : the performer's side or the audience's side.

The first category of works is mainly focuses in understanding how tactile vibration is involved in the creation of a relation of intimacy between the performer and its instrument, and how crucial this kind of feedback is in controlling and mastering the instrument itself. The first who analysed in depth this kind of problem is probably Chafe who, in a paper which dates back to 1993 ([1]) showed that introducing vibrotactile feedback in a controller led to a significant improvement in the performance of the subject testing the apparatus when controlling a realtime physical model of a brass.

Other experiences (such as [5], [6] or [7]) were performed in the following years: a recent one was designed by J.Rovan and V.Hayward ([7]) and studied the possibilities of adding vibrotactile feedback to open air wireless controllers, with the aim of improving their playability. This is an important aspect which has been stressed by several authors who remarked how the vibrotactile information is essential when speaking about professional performances ([8]). The tactile channel seems to be the only one fast enough to convey the huge amount of information needed to the performer for proficiently controlling the instrument.

In this category finds his place also D.Birnbaum ([9, 10]) whose work involved the realization of a digital musical

instrument, embedded with tactile actuators, all controlled by a synthesizer of vibrotactile events. This projects will be described in Section 4 since it is the base for the realization of our prototype.

The other category of research investigates the role of vibrotactile information in the perception of sounds and music. The applications are many : spatialization tools, multi-modal displays and sensory substitution. An application of this latter one is the prototype developed at Ryerson University by M. Karam and D. Fels for their *Model Human Cochlea* project ([11]) which uses the tactile channel to help deaf or hard-of-hearing people experience music through a special chair embedded with speakers. Each speaker received only a part of the spectral content of the music, although the criteria by which the frequency bands were chosen and mapped to a particular speaker does not seem to be evident.

Another project is *Cutaneous Grooves* ([2]) which was developed at the MIT Media Lab : the objective was to create a real "tactile composition language" and accompany a musical performance with a series of vibrotactile events expressly designed for the specific composition. The audience had to wear a special suit embedded with tactors and other kinds of vibrating actuators which received the signals from the tactile composition environment. This multi-modal experience aimed to expand the musical experience with extra tactile stimulation and allowed the composer to "highlight different parts of the music [...] and focus the audience's attention on different aspects of the music" using tactile stimulation.

3. THE TACTILE SENSE

3.1 Physiology of the tactile sense

Tactile perception operates through a network of cutaneous receptors present in human skin and is responsible of sensations like pressure, temperature, texture, orientation, vibration and many others. Its role is crucial in motor control and in the execution of many simple and complex tasks.

In the glabrous skin we can identify three different layers: *epidermis*, *dermis* and *subcutis*; each of these layers contains different kinds of receptors of which, those responsible for the perception of vibrotactile events are the corpuscles of *Meissner*, *Merkel*, *Pacini* and *Ruffini* ([9, chap. 3] and [3]). Each is associated to a different sensory channel which analyses different features of the stimulus.

Two families of corpuscles exists, the Fast Afferent (FA) family and the Slow Afferent (SA) one; the first one is characterized by the fact that the response of these corpuscles to a given stimulus ceases very rapidly, while the SA ones keep responding longer after the stimulation began. This behaviour is called *adaptation property*.

Meissner and Pacini's corpuscles belong to the FA family, while Merkel and Ruffini's belong to SA family; here is a brief description of each receptor and its main properties (see [9] and [3] for more details):

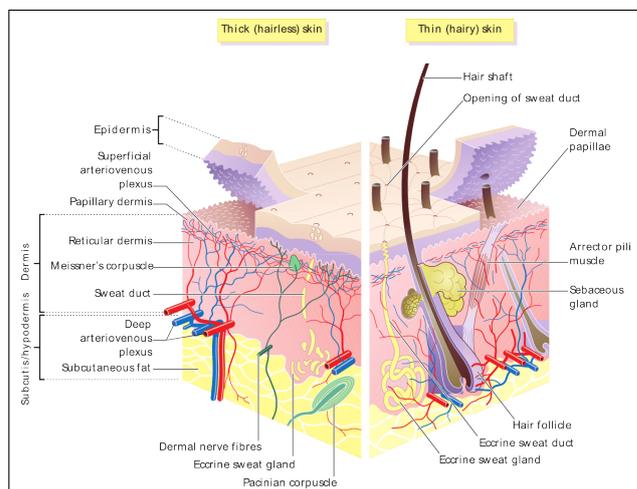


Figure 1. The structure of the skin with the different corpuscles in evidence (Image from en.wikipedia.org, released under the Creative Commons Attribution-Share Alike licence).

1. **Pacini (FA)** : Pacini's corpuscles are present in the deeper layers of the skin and their primary role is to protect the nerves from the vibrations given by the manipulations of objects in everyday life. They operate as a high pass filter, not allowing the low-frequency but energy-rich vibrations to reach the nerves. On the other hand they are very sensitive to the frequencies higher than 40Hz. The way Pacinians receptors behave when firing neural signals after the reception of the stimulus, is very similar to the way the auditory system reacts. This seems to be an evidence to the fact that the Pacinian channel is the one to be mainly exploited for the mediation of musical vibrotactile events ([9, Chap. 3]).
2. **Meissner (FA)** : these receptors are very sensitive to lower frequencies and are disposed in the superficial part of the dermis. Their spatial resolution is not very high, meaning that they cannot accurately sense where the stimulation is taking place; their principal function is to guide the grip control on the objects we manipulate.
3. **Merkel (SA)** : These cells have a very good spatial resolution and are very sensitive to the changes in texture of the objects manipulated.
4. **Ruffini (SA)** : The functionality of these cells is the most difficult to identify. They are sensitive to lateral stretching and this gives rise to two hypothesis about their role. They could be involved in the control of moving objects or in the determination of the position of the body parts, according to the superficial tension of the skin.

A very important remark we have to make is that there exists a difference in the mechanisms which control vibrotactile perception in glabrous skin (which we have just described in the previous section) and in hairy skin. For hairy

skin the data we can find in the literature is not so accurate as for glabrous skin, but it is anyway sufficient to form a solid base on which design a project involving a whole-body vibrating experience.

The most important differences characterizing hairy skin is a higher threshold and lower peak sensitivity values compared to glabrous skin. Verrillo and others made a series of experiments on different parts of the body trying to identify this values ([14, 12, 13]), and other possible important features. When developing our prototype we tried to take into account this differences as much as possible, according to the available data.

3.2 Properties of the tactile sensation

A fundamental step in understanding how the perception of vibrotactile stimuli works is the individuation of frequency thresholds and frequency ranges to which the skin is responsive. Verrillo ([3]) in the following years, made intensive studies on this subject, producing very detailed data about threshold, frequency ranges of sensitivity depending from amplitude and size of the actuator used.

The best data we have concerns the skin in the fingertip area, which has probably been the one studied more in depth; less accurate but still valuable data is available for other parts of the body. Also other important parameters have been identified by these studies like a set of equal sensation magnitude curves, very similar to the ones we have for the auditory system.

The frequency response of the skin has a characteristic inverted U-shaped form which spans from 40 Hz to 1000 Hz with a peak sensitivity around 250 Hz for every kind of factor used in the tests and at any amplitude. Another important aspect to consider is the discrimination of different frequency values; it has been shown that the tactile system has a very poor resolution compared to the auditory system when taking into account this particular feature. Rovin and Hayward ([7]) have suggested that the glabrous skin is capable of identifying from 3 to 5 different values for a continuous change in frequency from 2 Hz to 300 Hz and from 8 to 10 values when going from 70 Hz to 1000 Hz . This shows that, even if far away from the accuracy of the auditory system (for which the just noticeable difference goes down to 0.3%), the tactile channel is still able to determine gross frequency changes. Speaking about spectral content, Rovin and Hayward ([7]) have also showed how a sinusoidal wave is normally associated to a smooth vibration, while rougher vibrations are connected to signals with richer spectra, such as a square wave. This property can be used to give very interesting effects to the kind of signals used for vibrotactile feedback, allowing to stress some properties of the signal more than others. For example, a sine wave could be used for lower frequencies and a square wave could be suitable for increasingly higher frequencies, producing an effect of perceived *brightness* in the tactile stimulation ([9, Chap. 3]).

Another aspect to take into account is the appearance of *masking* and *adaptation* phenomena, which can have a sig-

nificant impact on the overall perception of the stimulus : *Masking* occurs when the presence of a background stimulus makes the primary one go undetected; it's an important factor to consider since it can dramatically increase the threshold value for the given signal. *Adaptation* is instead present when the extended exposition to a given stimulus decreases the sensitivity to the following ones, also increasing threshold or decreasing the magnitude perceived. When designing an experience involving vibrotactile feedback, those two aspect have to be seriously taken into account to avoid any possible interference with the final results.

Beside that, there are also two other phenomena which can be important to consider : *Enhancement* and *Summation*. The first one works essentially in the opposite way than *Masking*, in fact a second signal can be used to amplify the perceived magnitude of the first one, and this can be done in two different ways; spatially, by presenting simultaneous stimuli in different loci, or temporally, by presenting the two stimuli one immediately after the other. *Summation* occurs when the second stimulus is integrated with the first one, without changing its perceived magnitude.

4. THE PROTOTYPE

The most important thing we had in mind while conceiving this project was to reunite in one testing prototype the two main categories we describe in Section 2. As in the second category we described, we designed an experience where a performer would play on top of an existing base track and using a whole body display we created a representation of the sound of the base track *on the skin* of the performers. The aim was to design a multi-sensory environment involving the normal auditive stimulation and the added vibrotactile stimulation to test if this added information could be useful to increase the performer's degree of control of the instrument (problem usually addressed in the first category of experiments seen in Section 2).

The conception of our prototype started from the comparison of the works cited in Section 2, [2] and [11] in particular; after the analysis of the technology and the methods used for conducting those experiences we realized that most of them were based on some arbitrary decisions in the designing phase, meaning that the choice of the type of signal or of the actuators for example, seemed to be led more by an a priori decision than by a clear and systematic use of the data on vibrotactile perception. Most of the times, the prototypes and the testing environment we considered did not have any counterpart in the literature about physiology of the tactile sense to justify why they were designed in a certain way more than in another.

As we said, D. Birnbaum's work was the only one between those we considered to approach the problem of exploring vibrotactile feedback using the physiological basis as guidelines to develop his prototype.

He investigated the role of each of the four channels involved in tactile perception and developed a synthesizer of vibrotactile events, the *FA/SA* application, expressly tuned to stimulate the skin with the right frequency range, avoid-

ing as much as possible masking phenomena, balancing the effect of the augmented sensitivity to lower frequencies ([9, chap 4]). The application receives in input the sound coming out from his *BreakFlute* instrument (a flute-like controller embedded with little actuators on the holes, to stimulate fingertips upon pressure) and applies the series of processes we will describe in Section 4.2 to output the signal to send to the vibrating actuators. In this way the final signal still preserves the features of the original sound, but rescaled and tuned to the different sensitivity of the tactile channel. The feedback sent to the fingertip is a "projection" on the tactile space of the original sound in the auditory space.

The test Birnbaum conducted on its prototype gave very interesting results : all the performers who tried the prototype experienced a bigger connection with the instrument, and an improved ability to control the output sound when the vibrotactile feedback was activated.

The design of the display we built was inspired in particular by two already cited works we already spoke about : the *Cutaneous Grooves* ([2]) and the *Model Human Cochlea* ([11]). What we wanted to do was to go in the same directions given by those two experiences (whole body display, mapping different bands of the audio signal to different parts of the body) but keeping in mind as much as possible the physiological limits and the peculiarities of the tactile sense. With these considerations in mind we tried to identify the best distribution of the actuators on the body, ending up with the scheme in Figure 2.

The distribution is symmetrical on the limbs and has as center the two actuators placed on the back; it is also specular on the right and left side of the body. The actuators on the limb marked with the same number receive the same signal from a single output channel. The two actuators on the back, even with a different numbering also receive the same signal, but they have each one their own output channel to increase the intensity of the signal and also to allow different, non symmetrical dispositions in future tests.

This set-up seemed to suit the idea of creating a tactile, spatial representation of sound onto the body of the performer, allowing to map different features of the audio channel on different loci. What we did was to map the lowest frequencies to the back, and increasing them symmetrically going further on the limbs, towards the wrists and ankles. The center of the frequency bands selected via the band pass filtering changed according with the spectral distribution of the different base tracks.

We decided to double the channel on the back with two independent actuators driven by the same signal, since the frequencies mapped to this area would likely correspond to the drum-kick of the base tracks in our catalogue. This is very important for the tempo and we wanted to make this parameter as evident as possible for the players.

For the software part of the project we implemented a synthesizer of vibrotactile events, following the directions given by Birnbaum in [9].

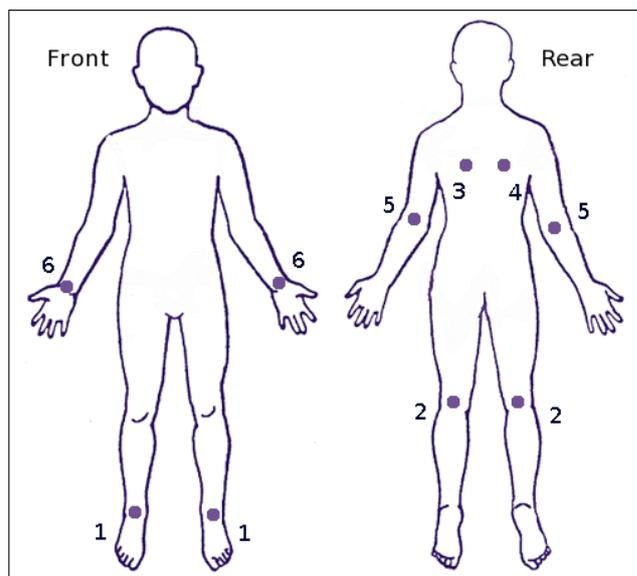


Figure 2. The distribution of the actuators on the body.

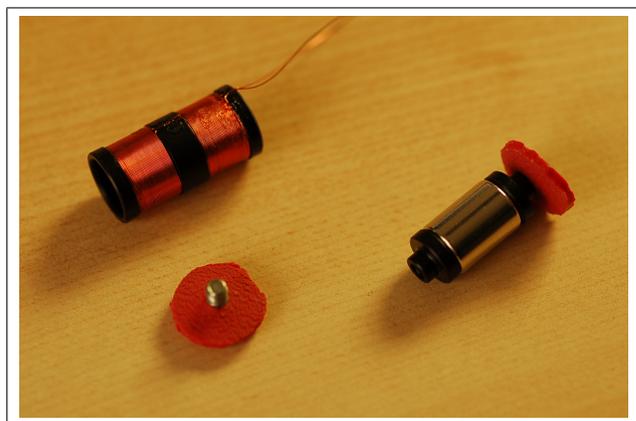


Figure 3. One of the actuators as used in [15] (Photo by courtesy of J. Malloch).

4.1 Actuators

The hardware prototype we built up consisted of ten vibrating actuators, based on the design of Hsin-Yun Yao ([16]). Similar actuators were used by J. Malloch in his vibrating version of the T-Stick instrument ([15]) (these actuators can be seen in Figure 3).

Each of the actuators consisted of a small plastic pipe of 5 cm length and 1.5 cm diameter, wrapped in a very thin conductive cable. The two coils formed on the two halves of the pipe are wrapped in opposite directions so to create a varying magnetic field in the interior of the pipe when an electric current flows in the cable. A magnet of the same size of the internal part of the pipe is then enclosed into it by two plastic caps with an added foam layer which allows the magnet to rapidly move with the change of the magnetic field, causing the vibration (see [16, Appendix A]). The choice of this particular type of actuators among the many types available on the market (see [17, 18] for a comprehensive list) was made because of their wide amplitude and frequency ranges, an independent control of these two parameters, a cylindrical form which made easy to apply

them in the designed loci on the body.

The actuators were connected to six 1 W mono amplifier designed by M.T. Marshall and based on the Philips TDA 7052 integrated circuit powered by two 9 V batteries connected in parallel to give a consistent current intensity. Each amplifier controlled two actuators for channels 1, 2, 5 and 6 (Figure 2) while the two actuators placed on the back were connected to one amplifier each (channels 3 and 4). Medical elastic gauzes were used to apply them on the body, allowing the performer to easily move and, at the same time, assuring a constant contact with the skin.

A 6.35mm Jack connector was used to plug the amplifier in the external sound card used, a firewire Digi 002 produced by Digidesign. This sound card provided a sufficient number of input and output channels and an easy configuration on both Microsoft Windows and Mac OS X.

4.2 Control Application

The Control Application (CA) was written in Max/MSP, and performed the necessary operations on the input audio to finally synthesize the signal fed into each of the 6 channels used by the actuators.

As mentioned this was essentially an extension of the FA/SA application written by D. Birnbaum ([9, chap.4]) with modifications and improvements needed for the specificity of this project. The input sound is preliminary converted to mono and band-pass filtered into 6 different signals centred on a given frequency to take only the desired part of the spectrum.

For each of the 6 signals we then perform a spectral analysis, producing the psychoacoustic descriptors necessary to generate the signals sent to the actuators; this parameters are used to generate the final tactile signal so that the main perceptual features of the signal will remain in the final *tactile signal*. An adjustable envelope triggered is applied to this signal to give it a pulse effect so to reduce the presence of masking and adaptation, which will be unavoidable with a continuous signal (see Section 3.2). The spectral centroid is used as base frequency to generate a sinusoidal and a square wave. This two waves are then added one to the other in a ratio which also can be defined by the user; This implementation allows us to chose the amount of "smoothness" or "roughness" to give to the output, so to enhance the perception and distinction of low and high frequency at the tactile level. The signal so generated is modulated in amplitude by the value of the loudness parameter.

After the signal has been generated, another filter is applied to eliminate any frequency out of the maximum and minimum skin receptivity range. Furthermore an equalization is made, to compensate the increased sensitivity of the skin to higher frequencies (Verrillo ([3]) found that equal sensation magnitude curves, similar to Fletcher-Munson curves for the auditory sense, exist for the tactile sense. We tried to take this into account with this last treatment of the signal).

After this final treatment each signal is sent to the specific actuators chosen to represent that predetermined frequency band on a specific location.



Figure 4. One of the players testing the display.

5. PRELIMINARY TESTS

We decided to perform the first test on five beginner guitar players, even if the prototype and the software can be adjusted to work in different configurations, suitable for other instruments. The choice of taking guitar players was mainly based on the availability of the subjects and on the fact that we wanted to concentrate only on one class of players to have more comparable results in this preliminary phase.

We asked the subjects to participate to a simple experiment and recording their overall impression about the display. After placing the actuators on the body, we asked them to chose in a catalogue a song they knew or felt comfortable to play (the 20 songs listed in the catalogue were "classics" of pop and rock music, no longer than five minutes; the most chosen one was "Smells like teen spirit" by Nirvana). We then played the part of the track consisting only of base and drums through the headphones they were wearing and asked them to play the missing guitar parts. We repeated the experience feeding the track to the FA/SA driven application which subsequently activated the tactile display, and asked the player to perform again on the same track.

When we asked about their impressions between the two modalities, all the novice players agreed that the presence of the vibration helped them to keep "more focused on the instrument", freeing them from paying too much attention to the accompaniment coming from the headphones; some of them stressed in particular the benefit of the presence of the actuators on the back, more sensitive to the low fre-

quency of the kick-drum, which helped them to stick on the tempo. They also remarked how the experience felt more "immersive" and how they felt more involved in the task.

These results seem to suggest that the extra tactile information could be useful for beginner players, since it seems to be capable of augmenting their ability to focus on the instrument, therefore allowing them to be more in control when performing the the given task. This factor could be useful to develop new strategies for the learning of the instrument, in which the player can *feel* the presence of other instruments playing, rather than just listening to them. The remark our subjects made about the more "immersive" connotation of the experience with the use of vibrotactile display, seem to suggest that they felt the tactile stimulation as something correlated to what they were listening, and not as a distinct stimulation. They felt to interact more with the base track, rather than passively listening it trough the headphones.

6. CONCLUSIONS

The use of vibrotactile stimulation in musical applications has been an important topic of research in the last twenty years, but the role of this kind of sensation and the ways to exploit it for conveying musical content are yet to be fully understood.

With this project we tried to proficiently use our knowledge of the physiology of the tactile sense to build a tactile display capable of translating acoustic properties on the tactile dimension. The preliminary tests we performed indicate that we are probably following a good path in this sense; the subjects seemed to be able to catch the connection between the track they were listening and the tactile stimulation built from that. Our goal was to build a display capable to help beginner guitar players in the control of the instrument and also the preliminary results show that the vibrotactile stimulation could be useful in this sense.

For our future work we plan to perform more accurate tests on guitar players, designing a formal and measurable framework by which precisely evaluate the effectiveness of the tactile display. We will start testing also on other instruments and we will develop new tasks and configurations for the placement of the actuators to find the most efficient ones according to the kind of instrument we will be testing on.

7. ACKNOWLEDGEMENTS

This project initiated as part of the first author's Master Thesis at the "Association pour la Création et la Recherche sur les Outils d'Expression" (ACROE), Grenoble, France during a research internship at the IDMIL in 2010. We would like to thank Hsin-Yun Yao and Vincent Hayward for developing the actuators. Partial funding for this research was provided by NSERC Special Research Opportunity and Discovery Grants to the second author.

8. REFERENCES

- [1] C. Chafe, "Tactile audio feedback," *Proceedings of the International Computer Music Conference (ICMC)*, pp. 76–79, 1993.
- [2] E. Gunther and S. O'Modhrain, "Cutaneous grooves: Composing for the sense of touch," *Journal of New Music Research*, pp. 369–381, 2003.
- [3] R. T. Verrillo, "Vibration sensation in humans," *Music Perception*, pp. 281–302, 1992.
- [4] P. B. y Rita, "Tactile sensory substitution studies," *Annals of the New York Academy of Sciences*, pp. 83–91, 2004.
- [5] A. Bongers, "Tactical display of sound properties in electronic musical instruments," *Displays vol. 18*, pp. 129–133, 1998.
- [6] M. T. Marshall and M. M. Wanderley, "Examining the effects of embedded vibrotactile feedback on the feel of a digital musical instrument," in *Proceedings of the 11th International Conference on New Interfaces for Musical Expression (NIME11)*, 2011.
- [7] J. Rován and V. Hayward, "Typology of tactile sounds and their synthesis in gesture-driven computer music performance," *Trends in Gestural Control of Music*, pp. 297–320, 2000.
- [8] L. L. Chu, "Haptic feedback in computer music performance," *Proceedings of the International Computer Music Conference (ICMC)*, pp. 57–58, 1996.
- [9] D. M. Birnbaum, "Musical vibrotactile feedback," Master's thesis, Schulich School of Music - McGill University, 2007.
- [10] D. M. Birnbaum and M. M. Wanderley, "A systematic approach to musical vibrotactile feedback," *Proceedings of the International Computer Music Conference (ICMC)*, pp. 397–404, 2007.
- [11] M. Karam and D. I. Fels, "Designing a model human cochlea : Issues and challenges in crossmodal audio-haptic displays," *Proceedings of the Ambi-Sys workshop on Haptic user interfaces in ambient media systems*, p. Article number 8, 2008.
- [12] R. T. Verrillo, "Vibrotactile thresholds for hairy skin," *Journal of Experimental Psychology*, pp. 47–50, 1966.
- [13] M. Morioka, D. J. Whitehouse, and M. J. Griffin, "Vibrotactile thresholds at the fingertip, volar forearm, large toe and heel," *Somatosensory and Motor Research*, pp. 101–112, 2008.
- [14] S. J. Bolanowski, G. A. Gescheider, and R. T. Verrillo, "Hairy skin: psychophysical channels and their physiological substrates," *Somatosensory and Motor Research*, pp. 279–290, 1994.

- [15] J. Malloch, "Adding vibrotactile feedback to the t-stick digital musical instrument," IDMIL - McGill University, Tech. Rep., 2007.
- [16] H.-Y. Yao, "Touch magnifying instrument applied to minimally invasive surgery," Master's thesis, Faculty of Engineering - McGill University, 2004.
- [17] M. T. Marshall and M. M. Wanderley, "Vibrotactile feedback in digital musical instruments," *Nime Proceedings*, pp. 226–229, 2006.
- [18] M. T. Marshall, "Physical interface design for digital musical instruments," Ph.D. dissertation, Schulich School of Music - McGill University, 2009.

FUNCTIONAL SIGNAL PROCESSING WITH PURE AND FAUST USING THE LLVM TOOLKIT

Albert Gräf

Dept. of Computer Music, Institute of Musicology
Johannes Gutenberg University Mainz
Dr.Graef@t-online.de

ABSTRACT

Pure and Faust are two functional programming languages useful for programming computer music and other multimedia applications. Faust is a domain-specific language specifically designed for synchronous signal processing, while Pure is a general-purpose language which aims to facilitate symbolic processing of complicated data structures in a variety of application areas. Pure is based on the LLVM compiler framework which supports both static and dynamic compilation and linking.

This paper discusses a new LLVM bitcode interface between Faust and Pure which allows direct linkage of Pure code with Faust programs, as well as inlining of Faust code in Pure scripts. The interface makes it much easier to integrate signal processing components written in Faust with the symbolic processing and metaprogramming capabilities provided by the Pure language. It also opens new possibilities to leverage Pure and its JIT (just-in-time) compiler as an interactive frontend for Faust programming.

1. INTRODUCTION

Programming signal processing applications in imperative programming languages can be a difficult and error-prone task. Functional programming provides an alternative way to tackle these problems. It allows signals to be modelled either as discrete functions of time or, equivalently, as streams (potentially infinite lists) of samples or control messages. The “patching” of signal processing components can then be expressed very conveniently through the combination of higher-order functions. An important benefit of this approach is that functional programs typically have simpler semantics and can thus serve as formal and platform-independent specifications of signal processing components.

Pure and Faust are two functional programming languages which are useful in this context and which nicely complement each other. Faust is a statically typed language based on the lambda calculus which has been designed specifically for synchronous processing of numeric signals at the sample level [1]. It enables you to build complicated signal processors from simple components such as

pointwise arithmetic operations and delays by means of its built-in block diagram algebra. Faust has an optimizing compiler with which Faust programs can be compiled to efficient native code.

In contrast, Pure is a dynamically typed general-purpose language tailored for symbolic processing, which can be used to tackle the higher-level components of computer music and other multimedia applications [2]. Pure is based on term rewriting, thus Pure programs are essentially collections of symbolic equations which are used to evaluate expressions by reducing them to their simplest form. Pure compiles the term rewriting rules to efficient native code so that they are executed very efficiently. While Pure doesn't specifically target numeric signal processing, it provides a matrix data structure akin to MATLAB/Octave which can be used to handle copious amounts of sample data and interface to numeric signal processing code written in other languages in an efficient way.

While Faust is batch-compiled, Pure has a just-in-time (JIT) compiler and is typically used in an interactive fashion, either as a standalone programming environment or as an embedded scripting language in other environments such as Miller Puckette's Pd. Pure is based on the LLVM compiler toolkit, which opens some interesting possibilities to interface it with other LLVM-capable languages.

LLVM, the “Low-Level Virtual Machine”, is an open-source cross-platform compiler backend available under a BSD-style license, which forms the backbone of a number of important compiler projects, including Apple's llvm-gcc and the new clang compiler [3]. It is also used in Google's “UnladenSwallow” Python compiler [4] and the latest versions of the Glasgow Haskell compiler [5], as well as in OpenCL implementations by Apple, AMD and NVIDIA [6]. Besides static compilation, LLVM also offers JIT compilation which makes it usable in dynamic environments where bits of source code are compiled at runtime as needed before being executed. This is also the way it is typically used in Pure, although the Pure interpreter can also be invoked as a static compiler in order to produce native executables and libraries.

LLVM exposes a fairly low-level code model (somewhere between real assembler and C) to client frontends. This makes it a useful target for signal processing languages where the generation of efficient output code is very important. Thus an LLVM backend has been on the wish-list of Faust developers and users alike for some time. Such a backend is now available [7]. This paper reports on sub-

sequent work by the author to build an LLVM-based bridge between Faust and Pure.

Note that Faust requires a considerable amount of “glue code” to work in different environments, which is usually provided in the form of special C++ modules (known as “architectures” in Faust parlance). The Pure-Faust bridge described here allows this glue code to be written in Pure instead, which has the advantage that Faust modules can be tested and run interactively in a dynamic, interpreter-like environment without sacrificing execution speed.

2. USING FAUST WITH LLVM

To take advantage of Faust’s new LLVM backend, you currently need a fairly recent snapshot of the “faust2” branch of the compiler in the Faust git repository [7].

We’ll use the following little Faust module as a running example throughout this paper. It implements a simple additive synthesizer with control variables `gain` (volume), `gate` (note on/off) and `freq` (fundamental frequency in Hz). The output is the sum of three sine oscillators for the fundamental and the first and second overtone.¹

```
import("music.lib");

freq = nentry("freq", 440, 20, 20000, 1);
gain = nentry("gain", 0.3, 0, 10, 0.01);
gate = button("gate");

amp(1) = 1.0; amp(2) = 0.5; amp(3) = 0.25;
partial(i) = amp(i+1)*osc((i+1)*freq);

process = sum(i, 3, partial(i)) * gain
  * (gate : adsr(0.01, 0.3, 0.5, 0.2));
```

The `-lang llvm` option instructs the Faust compiler to output LLVM bitcode (instead of the usual C++ code). Also, for using Faust-generated code with Pure, you want to add the `-double` option to make the compiled Faust module use double precision floating point values for samples and control values.² So, if you saved the above Faust code in a source file `organ.dsp`, say, you’d compile this module as follows:

```
faust -double -lang llvm organ.dsp -o organ.bc
```

If you did everything right, you should now have the LLVM bitcode for our little Faust organ in the `organ.bc` file which is ready to be loaded by the Pure interpreter, as described in the following section. If you want, you can also have the Faust compiler print the code in a human-readable format (LLVM assembler) by omitting the `-o organ.bc` option. A description of this format can be found on the LLVM website [3].

3. LOADING A BITCODE MODULE IN PURE

The Pure interpreter has the capability to load arbitrary LLVM bitcode modules and make the external functions

¹ To keep things simple, we have hardcoded the relative amplitudes of the partials and the parameters of the ADSR envelop here. In a real application, you’d probably want to turn these into additional control variables.

² The `-double` option isn’t strictly necessary, but it makes interfacing between Pure and Faust easier and more efficient, since the Pure interpreter uses `double` as its native floating point format. This option is also added automatically when inlining Faust code (see Section 4).

defined in that code callable from Pure. It is worth mentioning here that the ability to load Faust modules is in fact just a special instance of this facility. Pure can import and inline code written in a number of different programming languages supported by LLVM-capable compilers (C, C++ and Fortran at present), but in the following we concentrate on the Faust bitcode loader which has special knowledge about the Faust language built into it.

Loading a Faust bitcode module in Pure is done with a special kind of import clause which looks as follows (assuming that you have compiled the `organ.dsp` example from the previous section beforehand):

```
using "dsp:organ";
```

It’s not necessary to supply the `.bc` bitcode extension, it will be added automatically. You can repeat this statement as often as you want; the bitcode loader then checks whether the module has changed (i.e., was recompiled since it was last loaded) and reloads it if necessary. On the Pure side, the callable functions of the Faust module look as shown in Figure 1. (This uses pretty much the same syntax as C extern declarations; you can obtain this listing yourself by typing `show -g organ::*` in the Pure interpreter after loading the module.) Also note that the interpreter automatically places the interface functions of the Faust module in their own `organ` namespace in order to avoid name clashes if several different Faust modules are loaded in the same Pure program.

The most important interface routines are `new`, `init` and `delete` (used to create, initialize and destroy an instance of the dsp) and `compute` (used to apply the dsp to a given block of samples). Two useful convenience functions are added by the Pure compiler: `newinit` (which combines `new` and `init`) and `info`, which yields pertinent information about the dsp as a Pure tuple containing the number of input and output channels and the Faust control descriptions. The latter are provided in a symbolic format ready to be used in Pure; more about that in the Section 5. Also note that there’s usually no need to explicitly invoke the `delete` routine in Pure programs; the Pure compiler makes sure that this routine is added automatically as a finalizer to all dsp pointers created through the `new` and `newinit` routines so that dsp instances are destroyed automatically when the corresponding Pure objects are garbage-collected.

4. INLINING FAUST CODE

Instead of compiling a Faust module manually and loading the resulting bitcode module in Pure, you can also just inline Faust programs directly in Pure. The necessary steps to compile and load the module will then be handled automatically by the Pure interpreter. To do this, you just enclose the Faust code in Pure’s inline code brackets. Behind the opening bracket, there’s a special tag identifying the contents as Faust source, which takes the form `‘-*- dsp:name -*-’`. The `‘dsp’` tag tells the compiler that what follows is Faust code, while the given `name` indicates the name of the Faust module (which, as we’ve seen, becomes the namespace into which the Pure compiler places

```
extern void buildUserInterface(struct_dsp_llvm*, struct_UIGlue*) = organ::buildUserInterface;
extern void classInit(int) = organ::classInit;
extern void compute(struct_dsp_llvm*, int, double**, double**) = organ::compute;
extern void delete(struct_dsp_llvm*) = organ::delete;
extern void destroy(struct_dsp_llvm*) = organ::destroy;
extern int getNumInputs(struct_dsp_llvm*) = organ::getNumInputs;
extern int getNumOutputs(struct_dsp_llvm*) = organ::getNumOutputs;
extern expr* info(struct_dsp_llvm*) = organ::info;
extern void init(struct_dsp_llvm*, int) = organ::init;
extern void instanceInit(struct_dsp_llvm*, int) = organ::instanceInit;
extern struct_dsp_llvm* new() = organ::new;
extern struct_dsp_llvm* newinit(int) = organ::newinit;
```

Figure 1. Call interfaces for the sample Faust module on the Pure side.

the Faust interface routines). The inline code section for our previous example would thus look as follows:³

```
%< -*- dsp:organ -*-
import("music.lib");

freq = nentry("freq", 440, 20, 20000, 1);
gain = nentry("gain", 0.3, 0, 10, 0.01);
gate = button("gate");

amp(1) = 1.0; amp(2) = 0.5; amp(3) = 0.25;
partial(i) = amp(i+1)*osc((i+1)*freq);

process = sum(i, 3, partial(i)) * gain
  * (gate : adsr(0.01, 0.3, 0.5, 0.2));
%>
```

You can insert these lines into a Pure script, or just type them directly at the prompt of the Pure interpreter. This method is particularly convenient when experimenting with small Faust modules interactively in the Pure interpreter, as it eliminates the edit-compile-link cycle needed when compiling the Faust modules separately.

5. RUNNING A FAUST DSP IN PURE

Let us now take a look at how we can run the Faust organ in Pure to generate some samples. This process generally involves the steps sketched out below. After loading (or inlining) the Faust module, you can type these commands at the command prompt ('> ') of the Pure interpreter.

Step 1. We first create an instance of the Faust signal processor using the `newinit` routine, and assign it to a Pure variable as follows:

```
> let dsp = organ::newinit 44100;
```

Note that the constant 44100 denotes the desired sample rate in Hz. This can be an arbitrary integer value, which is available in the Faust program by means of the `SR` variable.

Step 2. The dsp is now fully initialized and we can use it to compute some samples. But before we can do this, we'll generally need to know how many channels of audio data

³ The following Pure code requires Pure 0.47 or later, available on the Pure website at <http://pure-lang.googlecode.com>. If you get a syntax error trying to input the inline code sections, then most likely you have an older Pure version installed.

the dsp consumes and produces, and which control variables it provides. This information can be extracted with the `info` function, and be assigned to some Pure variables as follows:

```
> let k,l,ui = organ::info dsp;
> k,l;
0,1
```

Note that in our example, there's no audio input and just one channel of output samples (i.e., a mono output signal). We'll have a look at the control variables later.

Step 3. Next we'll need to prepare input and output buffers to hold the samples passed to and computed by the Faust module. Pure's Faust interface allows us to pass Pure double matrices as sample buffers, which makes this step quite convenient. In our example, we need a $k \times n$ matrix (which is an empty matrix in this case) for the input and a $l \times n$ matrix for the output. Here, n is the desired *block size* (the number of samples to be processed in one go). That is, there's one row in the matrices for each audio channel, and the size of each row is the block size. Suitable matrices can be created with appropriate calls of the `dmatrix` function defined in Pure's standard library:

```
> let n = 10; // the block size
> let in = dmatrix (k,n);
> let out = dmatrix (l,n);
> in; out;
{}
{0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0}
```

Step 4. We can now apply the dsp by invoking its `compute` routine:

```
> organ::compute dsp n in out, out;
{0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0}
```

So the output is still all zeros, which is hardly surprising since we didn't switch on the `gate` control yet. Information about the available controls can be found in the `ui` variable which we defined in step 2 above:

```
> ui;
vgroup ("organ",
[nentry #<pointer 0xf55ef8>
 ("freq",440.0,20.0,20000.0,1.0),
 nentry #<pointer 0xf55eb8>
 ("gain",0.3,0.0,10.0,0.01),
 button #<pointer 0xf55ec0> "gate"])
```

In general, this data structure takes the form of a tree which corresponds to the hierarchical layout of the control groups and values in the Faust program. In this case, we just have one toplevel group named ‘organ’ which is created by the Faust compiler. This group contains the freq, gain and gate parameters of the Faust dsp, which are represented in Pure as a list containing the relevant information about the type and name of each control, along with a double pointer which can be used to inspect and modify the control value. Where applicable, the data structure also lists the initial value, range and step size of a control, as they were specified in the Faust source.

While it’s possible to access this information in a direct fashion, there’s also a `faustui.pure` module included in the Pure distribution which makes this easier. First we extract the mapping of control variable names to the corresponding double pointers as follows:

```
> using faustui;
> let ui = control_map (controls ui); ui;
{"freq"=>#<pointer 0xf55ef8>,
"gain"=>#<pointer 0xf55eb8>,
"gate"=>#<pointer 0xf55ec0>}
```

The result is a Pure record value indexed by control names. E.g., the pointer which belongs to our gate control can be obtained with `ui!"gate"` (note that ‘!’ is Pure’s indexing operator). There are also convenience functions to inspect a control and change its value. Let’s see what happens if we set the gate control to 1.0 (meaning “on”):

```
> let gate = ui!"gate";
> get_control gate;
0.0
> put_control gate 1.0;
()
> get_control gate;
1.0
> organ::compute dsp n in out, out;
{0.000916099235344598,0.00187158614672218,
0.00283108045199002,0.00376191957462151,
0.00463511838936804,0.00542654633209689,
0.00611762817140048,0.00669550599165544,
0.00715357741464203,0.00749105048578719}
> organ::compute dsp n in out, out;
{0.00771236488124018,0.00782759103035606,
0.00785038625400857,0.00779836818632883,
0.00768982665928043,0.0075451428235108,
0.00738392801978776,0.00722516057726323,
0.0070835316007035,0.00697180757718093}
```

So we finally got some real output now. Note that the compute routine also modifies the internal state of the dsp instance so that a subsequent call will continue with the output stream where the previous call left off. Thus we can now just keep on calling compute to compute as much of the output signal as we need.

6. EXAMPLE: A PD OBJECT

After walking through the computation step by step, it is now an easy matter to turn this into a working program. Figure 2 shows the complete Pure code for an object named `organ~` ready to be loaded in Pd. To make this work, you need the `pd-pure` plugin loader (available as an addon module from the Pure website) which equips Pd with the capa-

bility to run external objects written in Pure [8]. A sample patch showing this object in action can be seen in Figure 3.

Note that the `organ_dsp` function of the program is the main entry point exposed to Pd which does all the necessary interfacing to Pd. Besides the audio processing itself, this also includes setting the control parameters of the Faust dsp in response to incoming “note” messages.

The object is actually implemented by a local function `organ` which carries with it the required information (the dsp instance, number of input and output channels, the control variables and the output buffer) as local variables defined by the `when` clause; this makes it possible to have several different `organ~` objects in the same Pd patch. The `organ_dsp` function returns this local function along with the number of audio inputs and outputs so that Pd can properly set up the object when it is inserted into a patch.

Note that the `organ` function is invoked by Pd with either a matrix or a control message as argument. The former happens when the object is run by Pd’s audio processing loop in order to produce a block of samples; the `organ` function handles this by simply invoking the Faust dsp and returning the output buffer to Pd. The latter case occurs whenever the object receives a control message. In this example, we have added code to handle lists of MIDI note numbers and velocities, which get translated to the appropriate settings of the control variables of the Faust dsp.

Also note that this implementation uses an “actor style” of processing which is close to how Pd works but involves local state. There are other ways to do this in a more functional style by using streams, see [2] for details.

Finally, note that instead of importing the Faust module with a `using` clause, we might just as well have inlined the Faust code as described in Section 4. By using the interactive live editing facilities provided by `pd-pure`, it then becomes possible to change both the Faust module and the control processing in the Pure part of the code on the fly, while the Pd patch keeps running. For details on these “livecoding” facilities we refer the reader to the `pd-pure` documentation [8].

7. CONCLUSION

The facilities described in this paper are fully implemented in the latest versions of the Pure and Faust compilers. They enable programmers to employ Pure as an alternative hosting environment for Faust. Programming the required glue code for interfacing Faust to other environments in Pure offers some substantial advantages over the C++ architecture interface supplied by the Faust compiler. Specifically, Pure provides a convenient interactive environment which facilitates testing and livecoding of Faust components. Pure scripts containing Faust code can also be batch-compiled to native executables in order to implement efficient standalone applications. In either case, you can use Pure’s collection of addon modules for pre- and postprocessing Faust input and output signals in any desired manner. In particular, Pure interfaces nicely with the GNU Scientific Library, Octave and Gnumeric, and it also offers the necessary facilities to deal with MIDI, OSC, audio and graphics in a

```
// organ~.pure

// These are provided by the Pd runtime.
extern float sys_getsr(), int sys_getblksize();

// Get Pd's default sample rate and block size.
const SR = int sys_getsr;
const n = sys_getblksize;

// Load the dsp.
using "dsp:organ";
using faustui;

organ_dsp = k,l,organ with
// The dsp loop.
organ in::matrix = organ::compute dsp n in out $$ out;
// Respond to note messages.
organ [num, vel] = put_control freq (midicps num) $$ // note on
  put_control gain (vel/127) $$ put_control gate 1.0 if vel>0;
  = put_control gate 0.0 otherwise; // note off
// Translate MIDI note numbers to frequencies in Hz.
midicps num = 440*2^((num-69)/12);
end when
// Initialize an instance of the dsp.
dsp = organ::newinit SR;
// Get the number of inputs and outputs and the control variables.
k,l,ui = organ::info dsp;
ui = control_map (controls ui);
{freq,gain,gate} = ui!!["freq","gain","gate"];
// Create a buffer large enough to hold the output from the dsp.
out = dmatrix (l,n);
end;
```

Figure 2. Pure code for organ object to be loaded in Pd.

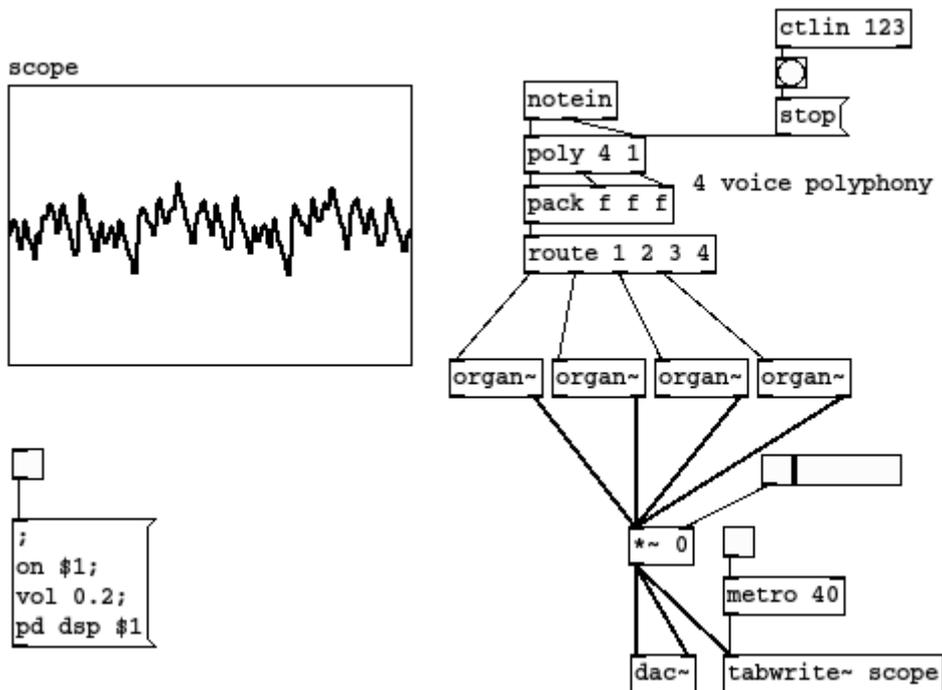


Figure 3. Organ objects in a Pd patch.

direct manner. GUI, database and web programming interfaces are provided, too.

Thus using Pure with Faust provides a great deal of flexibility and interactivity, more than can be achieved by using Faust modules directly in Csound, Pd, SuperCollider and similar environments, which don't provide facilities to change Faust code on the fly (at least not yet), and by design aren't general-purpose programming environments either. On the other hand, these are dedicated audio environments which have been designed specifically for (hard) realtime applications. In contrast, Pure only provides soft (i.e., best effort) realtime capabilities. We have found, however, that since Pure is compiled to native code, it is usually able to keep up with realtime loops quite easily, as long as the block size is sufficiently large and one doesn't try to squeeze too much computation into the realtime loop. E.g., Pd's realtime loop runs at 64 samples by default (offering latencies in the 1 msec ballpark) and Pure code running inside that loop usually works just fine, even if it is considerably more complicated than the simple example shown in this paper.

An interesting avenue for further research is to employ Pure as an interactive frontend to Faust. This is now possible (and in fact quite easy), since Pure allows Faust source to be created under program control and then compiled on the fly using Pure's built-in `eval` function. Taking this idea further, one might leverage Pure's symbolic computation capabilities in order to embed Faust as a domain-specific sublanguage in Pure. This should provide an interesting alternative to other interactive signal processing environments based on Lisp dialects such as Snd-Rt [9].

Acknowledgments

Many thanks go to Stéphane Letz at Grame for his work on the Faust LLVM interface which made this project possible in the first place. Special thanks are also due to Yann Orlarey for inviting me to Grame to work on improving our arsenal of functional signal processing tools.

8. REFERENCES

- [1] Y. Orlarey, D. Fober, and S. Letz, "Syntactical and semantical aspects of Faust," *Soft Computing*, vol. 8, no. 9, pp. 623–632, 2004.
- [2] A. Gräf, "Signal processing in the Pure programming language," in *Proceedings of the 7th International Linux Audio Conference*. Parma: Casa della Musica, 2009.
- [3] C. Lattner et al, "The LLVM compiler infrastructure," <http://llvm.org>, 2011.
- [4] "UnladenSwallow: a faster implementation of Python," <http://unladen-swallow.googlecode.com>, 2011.
- [5] D. A. Terei and M. M. Chakravarty, "An LLVM backend for GHC," in *Proceedings of the third ACM SIGPLAN Haskell Symposium*, ser. Haskell '10. New York, NY, USA: ACM, 2010, pp. 109–120.
- [6] "OpenCL: The open standard for parallel programming of heterogeneous systems," <http://www.khronos.org/llvm>, 2011.
- [7] S. Letz, "LLVM backend for Faust," http://www.grame.fr/~letz/faust_llvm.html, 2011.
- [8] A. Gräf, "pd-pure: Pd loader for Pure scripts," <http://docs.pure-lang.googlecode.com/hg/pd-pure.html>, 2011.
- [9] K. Matheussen, "Realtime music programming using Snd-Rt," in *Proceedings of the International Conference on Computer Music*. Belfast: Queen's University, 2008.

RAPSCOM - A FRAMEWORK FOR RAPID PROTOTYPING OF SEMANTICALLY ENHANCED SCORE MUSIC

Julian Rubisch

Institute for Media Production
University of Applied Sciences
St. Pölten
jrubisch@fhstp.ac.at

Jakob Doppler

Institute for Media Production
University of Applied Sciences
St. Pölten
jdoppler@fhstp.ac.at

Hannes Raffaseder

Institute for Media Production
University of Applied Sciences
St. Pölten
hraffaseder@fhstp.ac.at

ABSTRACT

In film and video production, the selection or production of suitable music often turns out to be an expensive and time-consuming task. Directors or video producers frequently do not possess enough expert musical knowledge to express their musical ideas to a composer, which is why the usage of temp tracks is a widely accepted practice. To improve this situation, we aim at devising a generative music prototyping tool capable of supporting media producers by exposing a set of high-level parameters tailored to the vocabulary of films (such as mood descriptors, semantic parameters, film and music genre etc.). The tool is meant to semi-automate the process of producing and/or selecting temp tracks by using algorithmic composition strategies to either generate new musical material, or process exemplary material, such as audio or MIDI files. Eventually, the tool will be able to provide suitable raw material for composers to start their work. We will also publish parts of the prototype as an open source framework (the RaPScoM framework) to foster further development in this area.

1. INTRODUCTION

1.1 Context

In contemporary film or video productions, score music composition or selection is widely regarded as vital for the movie's reception and the conveying of moods, metaphors and meanings. It is, however, also sometimes treated as an orphan because of its expensiveness and time-consuming qualities. Moreover, movie directors or video producers frequently lack the musical expertise to communicate their wishes and ideas to a film composer. Therefore, in the majority of cases temp tracks are used as a fallback.

Within the community of film composers, however, temp tracks are being disapproved of, as they often confine the composer's imagination. In many cases, directors also cling to their temp tracks' musical features (themes, harmonies, rhythmic features etc.) very tightly, which makes film music production a complex and inefficient process for both sides.

Copyright: ©2011 Julian Rubisch et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1.2 Objectives

The research project GeMMA (Generative Music for Media Applications¹) tries to ameliorate this situation by

- semi-automating the generation of temp tracks,
- enhancing the communication between director and composer (not necessarily by providing the optimal output, but with a focus on optimal description/selection of the desired output),
- providing an intuitive user interface, usable for both musical experts and laypersons and
- processing initial musical structures (audio or MIDI files) so as to facilitate rapid prototyping by example.

It is also a goal to release parts of the employed algorithms as an open source project, the RaPScoM framework.

1.3 Related Work

There have been numerous initiatives to exploit algorithmic composition in an affective or semantic way, of which we will cite a few ones:

[1] describes a state based sequencer for automatic sound track generation that consists of pluggable agents for tempo, key, chord, instrumentation and rhythm. The generation process uses mainly genetic algorithms and markov models for chord progressions but requires a thorough screenplay, character, location and event annotation and does not analyze existing material.

[2] designed an Algorithmic Music Evolution Engine (AMEE) which uses an emotional mapper to link a perceived mood to a set of musical parameters, such as consonance, pitch, mode, articulation, tempo etc. Their model, however, relies on a discrete, taxative set of moods which cannot be altered by the user.

[3] use Russell's circumplex model of affect [4] as a basis to alter a set of musical parameters. Notably, they have divided their approach into a timing (groove), a harmonic and a voicing module to combine several features.

A common weakness heretofore, at least to our knowledge, is the general neglect of the necessity to find novel ways of description regarding audiovisual source material other than plain technical video or audio descriptors. In

¹ <http://gemma.fhstp.ac.at>

the TRECVID evaluation [5], semantic indexing tasks have been rather poorly performing to the present day.

2. METHODOLOGY

The field of algorithmic composition and generative music is structured by the influence of various disciplines, such as computational intelligence, music psychology and philosophy, semiotics and, of course, musicology. Therefore, a tripartite approach towards the problem has to be taken.

2.1 Socio-Cultural Approach

When analyzing music for film or video, it is essential to consider

- the functions it is able to incorporate,
- the levels of impact it triggers as well as
- what semiotic structures it possesses.

[6] [7] [8] and [9] provide an overview of the structured analysis of the field of film music and sound semiotics, while [10] analyzed what classes of functions film music incorporates and utilizes to achieve a certain impact.

These classes include ([10], p.2)

the emotive class: mood induction (emotions experienced by the audience) and communication of emotion (only identified by the audience)

the informative class: communication of meaning, communication of values and establishing recognition

the descriptive class: describing setting or physical activity

the guiding class: indicative (guide the audience's attention) or masking (disguise other noises or narrative elements)

the temporal class: providing continuity (disguising of cuts from scene to scene, usage of leitmotifs etc.), defining structure and form (forming the perception of time and speed)

the rhetorical class: commenting the narrative

The thorough understanding of film music semiotics is crucial here, because music in general uses symbolic gestures to fulfill the mentioned functions. In order to maintain a manageable scope of this vast field of research, we decided to focus on two central aspects:

- representation of affects and moods, and
- analysis of semiosis by abstract musical symbols.

2.1.1 Representation of Affects and Moods

Widely used in music psychology, Russell's Circumplex Model of Affect [4] presents a solid basis for both a computational representation of moods and emotions as well as an intuitive user interface. Briefly, the model assumes that every human affect is a linear combination of two neuro-physiological systems called valence (ranging from unpleasant to pleasant) and activation (ranging from passive to active). While this is indisputably an oversimplification, the model serves well in many music-psychological studies as well as music information retrieval (MIR) tasks [11].

2.1.2 Semantics

By many semioticians, music is seen as a semiotic system without semantic density [12], i.e. musical signs (e.g. melodies, motifs, rhythmic patterns etc.) have syntactic relations, as defined by music theory and harmonics, but no inherent meaning (as compared to linguistics, where words are assumed to carry a certain meaning). On the other hand, certain musical segments do carry clear denotative (e.g. hunting horn signals) or connotative (e.g. pastoral, sacral, etc. music) significances, and film music makes use of these connotative meanings quite excessively.

In fact, it seems advisable in the special case of film music to not only regard syntactic and semantic features, but to see the sounding material in the context of the plot - i.e. to consider the pragmatic aspects of a movie. After all, photography, editing, acting, sound design, music, lighting, and many more aspects of a movie are welded together to form a certain narrative. It is thus possible to charge a simple musical gesture (e.g. a simple chord or tune) with a clearly defined meaning by setting it in an appropriate, coded context. The relation of the music to the (mostly visually defined) context can be either

paraphrasing: duplicating what is seen on the screen (e.g. a romantic tune to a love scene),

polarizing: charging a neutral context with meaning, or

contrapuntal: contradicting the visual narrative, thus introducing another semantic layer (cf. [7] [8] [13]).

To establish a language system, it is necessary to introduce conventional codes, i.e. stereotypes, which render film music a communicative art and form styles and traditions. However, it is necessary to differentiate stereotypes from cliches, which are stereotypes reduced to a certain, isolated meaning. The use of cliches is affirmative, it merely uses fixed assignments of signifieds without questioning underlying socio-cultural developments, i.e. the progress of tradition (cf. [13], p. 83f).

To analyze the usage of musical signs in various contexts (i.e. their use as stereotypical gestures), we divided our analysis in

- events (dominant, temporally confined narrative elements of a scene) and
- symbols (higher-level dramaturgical motifs of a scene) (cf. [6] [8])

2.2 Aesthetic Approach

It is of course crucial to consider whether and how aesthetically interesting output can be produced by an algorithmic engine. While it is also clear that this field entails many related questions (ethical, philosophical, cognitive ones), it is impossible to approach it in a quantitative way. The aesthetic content of a piece of music to a great degree relies on the listeners' anticipations, associations from their personal history as well as cultural backgrounds.

To obtain musically interesting results, artificial intelligence (AI) and/or life (AL) methods are experimented on, including pattern recognition and supervised learning methods, as well as artificial neural networks (e.g. echo state networks) and genetic algorithms.

In order to monitor the music's aesthetic impact on listeners, we are performing qualitative user and listener reviews as an accompanying measure (including interviews with composers and directors concerning style and impact of the generated music).

2.3 Technical Approach

The technical realization of the project yields yet another number of problems:

2.3.1 High-Level Architecture

As mentioned in the introduction, the tool is meant to be usable for both experts and laypersons. Therefore, it has to be clarified which set of parameters should be exposed to the users as well as how audio or MIDI input can be analyzed, including the evaluation of salient harmonic, melodic, and rhythmic features. Furthermore, models and algorithms for the generation of musical content have to be reviewed and tested.

2.3.2 User Interface

A central question that has to be addressed is in what way (non-)expert users should be enabled to interact with an intelligent music-generating engine.

2.3.3 Low-Level Building Blocks

Finally, the pivotal issue in this project is the question how the findings from the above mentioned approaches can be broken down into independent components which serve as building blocks for a generative music-making automaton.

3. RESULTS

3.1 Requirements and Constraints for Temp Tracks

Common score music production workflow is a loose triangular communication between editor, director and composer. First, in the spotting sessions editor and director use temp tracks from similar productions to produce a rough cut and expose ideas on the music theme and semantics of a scene. The composer then is required to transform these ideas into unique sounding score music. In an iterative process involving all persons the score music is merged with the simultaneous evolving rough cut to form the final product [13] [14]. The RaPScoM framework aims at improving

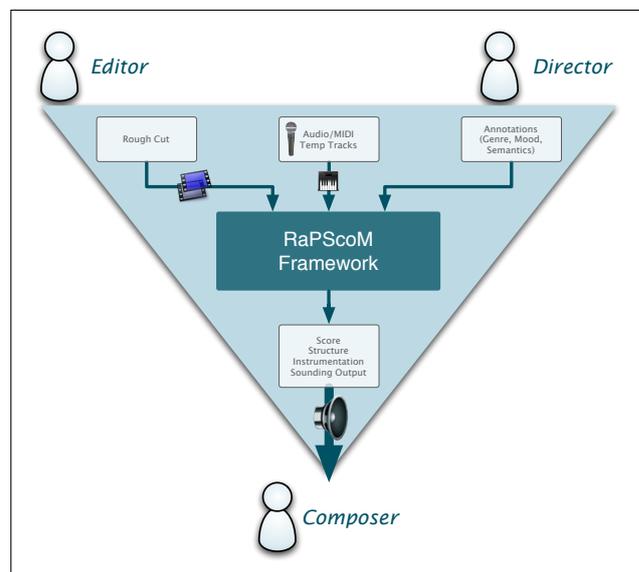


Figure 1. Triangular communication between editor, director and composer, enhanced by the RaPScoM framework.

the film scoring workflow by generating musically meaningful raw material such as musical structures and instrumentation for temp tracks that can directly be processed by the composer (see figure 1).

3.2 Semantic Analysis

As indicated above, a thorough understanding of film (music) semiotics is pivotal to the project of providing accurate music according to the director's raw ideas.

To gain insight concerning affective and semantic de- and connotations used in the language of film music, we conducted the following investigations:

3.2.1 Representation of Affects and Moods

A listener test was conducted, using the above mentioned Circumplex Model of Affect, to discover possible correlations between library score music and musical/timbral parameters by playing a set of music clips to a group of listeners (8 male, 6 female, between 21 and 50 years old) and asking them to

- place the respective clip in the Circumplex, and
- fill out a semantic differential of these musical/timbral parameters (*brightness, harmonicity, loudness, consonance, tempo, rhythm, pulse, dynamics, quietude and hardness*).

The subsequent statistical analysis indicated some strong correlations between loudness, hardness, quietude, tempo and activation, and somewhat weaker correlations between harmonicity, rhythm, pulse and activation, as well as brightness, consonance, tempo and valence. To a great extent, these findings correspond to phenomena described in relevant literature (e.g. [15]).

3.2.2 Symbolic Content of Movie Scenes

In order to find possible similarities regarding musical symbols and their significances, we analyzed a corpus of approx. 400 short movie clips (randomly selected) according to symbol, event and the employed musical instruments. To facilitate the analysis, symbols and events were aggregated into clusters by agglomerative hierarchical clustering [16].

In order to obtain a meaningful distance measure for these nominally scaled sets of data, we decided to construct a binary vector to represent the occurrences of symbols/events per clip (where 1 means *clip is tagged with a symbol* while 0 means the opposite). Thus, the record sets could be compared to each other by use of a hamming distance and clustered accordingly [17].

It turned out that agglomerative clustering with single linkage tends to quickly merge small clusters into larger ones, leaving single clips unclustered, which is why finally complete linkage clustering was employed. Agglomerative hierarchical clustering features the advantage of being able to determine the amount of clusters after the clustering process has completed. Therefore, the following 8 *symbol clusters* and 8 *event clusters* were selected and labelled by hand:

| Symbol | Event |
|-----------------|-------------|
| Action/Violence | Movement |
| Fear/Tension | Drama |
| Freedom | Accident |
| Joy/Comedy | Shock |
| War | Violence |
| Tragedy | Surprise |
| Romance | Death |
| Desolation | Celebration |

Table 1. Symbol and Event super categories retrieved by machine clustering

Further listening tests showed that 59 % of test persons (N=87) identified a strong relationship between symbol and instrumentation of a scored movie scene. Only within the symbol groups of *freedom* and *war*, melody is awarded a higher degree of correspondence with the intended symbolic meaning. Rhythmic features are almost never associated with semantic content of a movie scene.

Currently, a correspondence analysis of symbols and used instrumentation/solo instruments and melodic parameters (e.g. melodic contour, mode, ambitus) is conducted. Plausibly, as can be argued from music history, instrumental (or in a smaller degree melodic) stereotypes are used to convey a certain scene setting, e.g. horns for a war or hunting scene, flutes for a pastoral scene etc. Ideas for this approach were taken from the german Handbook of Film Music [13] and van Leeuwens Speech, music, sound [18]. The goal here is to provide a probability matrix for the instrumentation and melodic composition of scenes according to their semantic features.

3.3 Implementation Prototypes

In order to test the validity of the above mentioned affective and semantic models, we decided to implement different generator algorithms in a bottom-up approach first, before designing the entire framework. These include:

3.3.1 MotifFactory

This building block is planned to operate on a low level of the framework, and comprises methods to model a melody (and variations thereof) as well as a tune's consonance and rhythm (and variations). It is able to analyze and process initial MIDI or audio input and provide appropriate variations according to a predefined set of parameters. To accomplish this, the melody is broken down into a first- or second-order Markov chain and reassembled randomly.

Moreover, it is possible to pick a motif according to its melodic envelope (e.g. a falling slope, or first rise and then fall, etc.). On a higher level, the most appropriate variation will be selected and formed into a complete musical segment by an intelligent algorithm (e.g. an artificial neural network or an agent-based artificial life algorithm).

The major aim of this experimental implementation was to gain experience about how the variation of very short musical segments can already influence the perceived mood or affect of a tune. First results of this evaluation, which was conducted on short known melodies (e.g. Beethoven's *Für Elise*) sound very promising in terms of musical originality, while maintaining a clear similarity to the original and providing affectively biased variations.

3.3.2 SemanticChordProgressionGenerator

The purpose of this demo implementation was to investigate how larger-scale musical segments, spanning over a wider range of e.g. 8 bars, can be used to create, sustain and release musical tension. For the realization of this task, statistical chord progression data from [15], as well as algorithms from [19] and [20] were used. Currently, we are reviewing and implementing composition rule frameworks (e.g. prohibit the use of parallel fifths, encourage the use of a certain register, use close or open harmony, divide in antecedent/consequent etc.) to be included in this model. We are strongly convinced that in the use of small-scale variation of certain parameters (such as arpeggio style, dynamics, direction, rhythmic complexity and others) in a larger-scale context of harmony lies one of the pivotal foundations of semantically enhanced music generation.

In a first attempt to include symbolic information (e.g. *war* or *romance*), we decided on including an orchestral sample library here, and have the algorithm lock a certain instrumental arrangement before generating the chord progression. Another task we are focusing on is the investigation and evaluation of harmonies frequently used in film music.

4. THE RAPSCOM FRAMEWORK

4.1 Requirements

Our approach to rapid prototyping for score music is based on the semantic annotation of the rough cut. A movie de-

scriptor contains a set of global properties (musical film style/genre) and timeline parameters (movie semantics, emotions) which are tailored to the vocabulary of films. In the above mentioned study we found that music inherent movie semantics are best described as a set of 6 scene symbols (e.g. action, fear, romance) which are sparsely intermitted by 8 events (e.g. violence, surprise, shock). For the representation of emotions the mentioned Circumplex Model of Affect is used. Various input modalities such as tagging tools and 2D panels for example on touch tablets are currently under review for generating and editing annotations.

4.2 Environment

The framework's generation algorithms are geared to producing MIDI raw material; it is thus necessary for the user to install a MIDI-based host-application (such as any contemporary digital audio workstation, or sampler). To ensure a certain channel-instrument mapping (e.g. violins on MIDI channel #1, violas on channel #2 etc.), templates for a certain set of audio workstations and samplers will be provided. In order to be compatible with the General MIDI standard, the appropriate program change messages will be sent on each channel, too. It is however also possible to alter the default channel mappings in an external configuration file.

4.3 Structure

4.3.1 Models

Following [19], we decided to devise a hierarchical framework of musical structures to manage the analysis and generation of music on several symbolic levels. Thus, the backbone of the framework consists of models of *Note*, *Chord*, *Motif*, *Theme*, and *Piece*, where the latter can always contain multiple instances of the former (a *Piece* can contain many *Themes* etc., see also figure 2).

Every class in the hierarchy (except for *Note*) extends a common base class holding general properties of *NoteContainers*, e.g. a key, time signature, and handles for traversing the hierarchy (*getParent*, *getChildren*). They furthermore implement two interfaces, *IAnalyzeable* (making it compatible to several analyzer methods, see below), and *IVariable* (making it possible to create intelligent variations of a certain element).

For rhythmic purposes, every note has a *NotePosition* in upper (beat-), mid (tactus-) and lower (sub-tactus-)level format, as Temperley proposed in [20]. This note position can be moved by note values, such as (trioletic, dotted) quarter notes etc.

4.3.2 Processors

To accomplish the task of generating, analyzing and varying musical content, it is advisable to conceive a set of processor classes, divided in analyzers, such as

- KeyAnalyzer
- RhythmAnalyzer

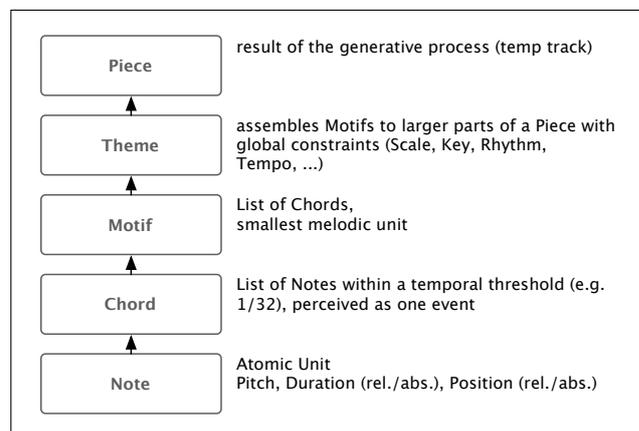


Figure 2. Model hierarchy

- ConsonanceAnalyzer

and generators, such as

- ChordProgressionGenerator
- ArpeggioGenerator
- MonophonicMotifFactory

These modules rely on *KnowledgeBases* to provide essential rulesets for the tasks of music analysis and generation (see also figure 3).

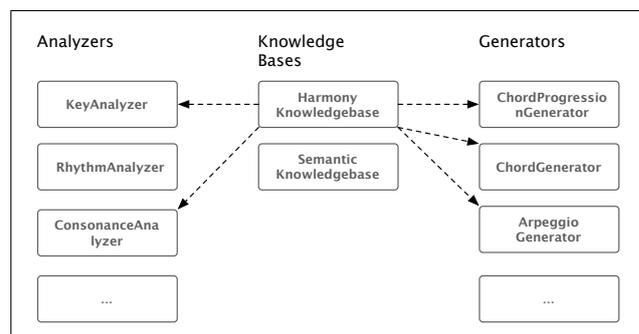


Figure 3. Processor modules

4.3.3 IO Modules

Every instance of the above mentioned hierarchy can be sent to a *Renderer* (e.g. a MIDI file renderer) to produce the respective output. A central part of the IO framework is the *ChannelsDescriptor*, where relevant information on the instrument-channel-mappings (such as name of the mapped instrument, pitch range, etc.) are stored.

4.4 Session

A RaPScoM project with inputs, outputs, requirements and defined workflow is called a *session*. A session is taken to mean a complete reference implementation of the framework's independent modules, allowing to semi-automatically generate musical output, as outlined in the introduction. It comprises (also see figure 4)

Semantic Movie Descriptor: film material annotated with valence/arousal, symbol, event, style and genre

Instrumentation: we pursue an *instrumentation first* policy here - before generating the actual musical elements, the orchestral arrangement is locked, based on the symbolic annotation of the scene. However, we also implement a feedback loop to be able to try out different instrumentations of the same score after it has been generated.

Sequence: the music sequence (a hierarchy starting with a *Piece*, down to the single *Notes*, is generated in accordance to instrument mappings, affective, stylistic and semantic annotations, as well as composition rules and harmonic guidelines. Every decision is logged, in order to trace back every step and from there start another generation of variations.

Channel Descriptor: as indicated above, information on the instrument-channel-mapping, voicing etc. is stored here

Host Application: determines the way the MIDI data is transformed to sounding material on the user's computer. It will be a future task to develop a description format (e.g. XML-based) of sound generators, in order to ensure that the generated audio material fits the listener's expectations

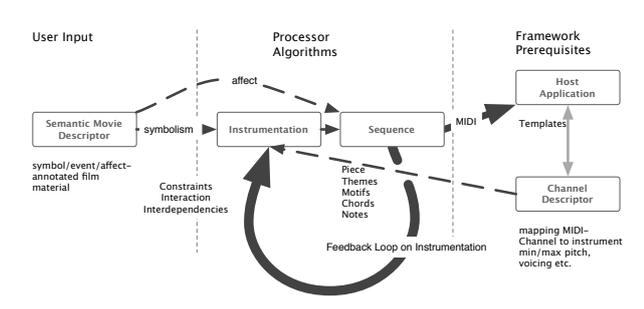


Figure 4. Flowchart of a RaPScoM session.

5. CONCLUSION

In this paper, we have introduced first steps toward a software framework to achieve semi-automated production of film music temp tracks in order to enhance the video and film production process. The main difficulty in such a project derives from the various different methodologies that have to be included, as well as the requirements put to the prototype by professional users, such as directors or video producers.

We proposed different ways of approaching these independent problems and identified salient factors for describing and evaluating score music according to semiotics and aesthetics. We are currently in the process of reviewing and experimenting on generative music-making algorithms in order to produce meaningful content which is capable of replacing or augmenting the wide practice of using temp tracks in film and video production.

We have further outlined the structure of the RaPScoM (Rapid Prototyping of Semantically Enhanced Score Music) framework, its requirements and constraints as well as in what type of environment it is meant to be used. We hope that we will be able to complete the core of the framework along with a prototypical reference implementation by mid 2012, which will fuel further discussions about music production in the creative industry.

Acknowledgments

GeMMA is funded by the Austrian Research Promotion Agency (FFG²) under the COIN (Cooperation & Innovation) programme on behalf of the Austrian Federal Ministry for Transport, Innovation and Technology³, as well as the Austrian Federal Ministry of Economy, Family and Youth⁴.

6. REFERENCES

- [1] M. O. Jewell, "Motivated music: Automatic soundtrack generation for film," Thesis, 2007. [Online]. Available: <http://eprints.ecs.soton.ac.uk/13924/>
- [2] M. Hoeberechts and J. Shantz, "Real-Time emotional adaptation in automated composition," in *Proceedings of Audio Mostly 2009 - a conference on interaction with sound*, Glasgow, UK, 2009.
- [3] I. Wallis, T. Ingalls, and E. Campana, "Computer-Generating emotional music: The design of an affective music algorithm," in *Proceedings of the 11th Int. Conference on Digital Audio Effects (DAFx-08)*, Espoo, Finland, 2008.
- [4] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, pp. 1161-1178, 1980.
- [5] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*. New York, NY, USA: ACM Press, 2006, pp. 321-330.
- [6] B. Flückiger, *Sound Design : die virtuelle Klangwelt des Films*. Marburg: Schüren, 2001.
- [7] M. Chion, *Audio-vision : sound on screen*. New York: Columbia University Press, 1994.
- [8] H. Raffaseder, *Audiodesign*, 2nd ed. München [u.a.]: Fachbuchverl. Leipzig im Carl-Hanser-Verl., 2010.
- [9] T. V. Leeuwen, *Speech, music, sound*. Houndmills Basingstoke Hampshire ;New York: Macmillan Press St. Martin's Press, 1999.
- [10] J. Wingstedt, "Narrative functions of film music in a relational perspective," in *Proceedings of ISME - Sound Worlds to Discover*, Santa Cruz, Tenerife, Spain.

² <http://www.ffg.at>

³ <http://www.bmwit.gv.at>

⁴ <http://www.bmwfj.gv.at>

- [11] L. Lu, D. Liu, and H. Zhang, "Automatic mood detection and tracking of music audio signals," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 5–18, 2006.
- [12] U. Eco, *Einführung in die Semiotik*, 9th ed. UTB, Stuttgart, Feb. 2002.
- [13] E. Schneider, *Handbuch Filmmusik*, [Verschiedene aufl., nachdruck] ed. Konstanz: UVK Verlagsgesellschaft, 2006.
- [14] R. Davis, *Complete guide to film scoring : the art and business of writing music for movies and TV*. Boston MA Milwaukee Wis.: Berklee Press Distributed by Hal Leonard, 1999.
- [15] D. Huron, *Sweet anticipation : music and the psychology of expectation*. Cambridge Mass. London: MIT, 2008.
- [16] R. Duda, P. Hart, and D. Stork, *Pattern classification*, 2nd ed. New York: Wiley, 2001.
- [17] X. Hu, M. Bay, and J. Stephen, "Creating a simplified music mood classification Ground-Truth set," in *Proceedings of the 8th International Conference on Music Information Retrieval*, Wien, Sep. 2007, pp. 309–310.
- [18] T. V. Leeuwen, *Speech, music, sound*. Houndmills Basingstoke Hampshire ;New York: Macmillan Press St. Martin's Press, 1999.
- [19] R. Rowe, *Machine musicianship*. Cambridge Mass. London: MIT, 2004.
- [20] D. Temperley, *Music and probability*. Cambridge Mass. London: MIT Press, 2010.

FOLEY SOUNDS VS. REAL SOUNDS

Stefano Trento

Conservatorio C. Pollini, Padova
trento.stefano@gmail.com

Amalia de Götzen

Sound and Music Processing Lab - SaMPL
Conservatorio C. Pollini, Padova
coordinatore@sampl-lab.org

ABSTRACT

This paper is an initial attempt to study the world of sound effects for motion pictures, also known as *Foley sounds*. Throughout several audio and audio-video tests we have compared both Foley and real sounds originated by an identical action. The main purpose was to evaluate if sound effects are always better than real sounds [1]. We found a similarity in subjects preferences between real sounds and Foley sounds, with a limited discrimination ability between them.

1. INTRODUCTION TO THE FOLEY ART

The majority of movies that are made today demonstrate such an effective and intensive use of Foley effects that their importance in animation and movie production has been widely recognized. The movie-goer will be affected by the sound so that her sonic experience will undoubtedly enhance the narrative stream of the movie. Therefore, it is appropriate to think about Foley sounds as a support for visual composition and characterization. The art of Foley grew up thanks to Jack Foley's inventiveness and open mind. He was the charismatic sound editor who invented this craft out of necessity, during the production of *Showboat*, a musical made at Universal Studios in 1929. From that moment on, with the passing of years and with the contribution of many Foley artists such as John H. Post, Ken Dufva, David Lee Fein and Robert Rutledge, this craft has become very popular in movie production [2]. The idea that Foley had was to provide a scene with sound effects by performing and adding them while the scene itself was being staged. The art of performing and creating these sounds effects consists in handling various kind of props and doing some strange movements in a special recording stage. The person who does this is called a *Foley artist*. She performs footsteps, clothes movements, props and everyday sounds both for movie, radio programs and TV shows. Essentially, she has to pay attention to what the actor is doing on screen and on account of that, she must choose the correct props to reproduce the better sound for that scene. For instance, she must observe if the actor is walking on a wooden floor or if he is beating somebody up with either his hands or any kind of prop. Her role is



Figure 1. Typical Foley stage, with various pits, props and monitor for sync.

important because through Foley effects she can emphasize, enhance, support, replace and even create the reality of an action. The stage where Foley artists perform is called a *Foley stage* – an example of a Foley stage is shown in fig. 1. We have to think about the Foley stage as a place to design custom sound effects. Normally, it consists of two separated rooms. One is dedicated to the performance of Foley artists, while the second is used to record them. The sound technician, who records hand props, footsteps, clothes and custom effects that are needed to be performed on the Foley stage, is called a *Foley editor*.

Basically, a Foley stage needs an extensive technical equipment. The set up of this room is composed by: a monitor or projector, one or more pits, microphones and props. The monitor is used to help the synchronization between the action performed by the Foley artist and the action played on the screen. The pit - whose dimensions are at least 3 x 4 feet with a depth of about 6-8 inches - is used to perform the footsteps on different surfaces. For example, a pit could be filled up with coffee and another one with stones. In this way, the Foley artist can perform footsteps from different floors at the same time, without blocking the action. This is of key importance in order to be fluent and to enhance the narrative stream of the movie. Moreover, some Sound editors - known as Mixers - prefer a Foley stage with very little reflection so as to obtain an acoustically dead room. After all, a Foley effect recorded flat – that is without equalizations and perspective – is easier and quicker to process.

2. OUR FOLEY EFFECTS

2.1 Choosing Foley Effects

After a preliminary study of the Foley world, the next step was to find out which sounds – or, to be more precise, which actions – were needed for our objectives. We chose the eight different actions that follow: slapping, uncorking a bottle of wine, breaking bones, bird wings flapping, kissing, walking up the stairs, walking on summertime grass, closing a sliding door. What follows is a list which defines the Foley methods to produce each action. For the equivalent real sounds such a list is obviously not needed as it is quite simple to imagine.

- **Slapping:** this sound is created by holding a piece of raw steak with one hand and hit it with the open palm of the other in its center. To simulate a person being slapped it is common practice to use the same method with slices of steak of different thickness depending on the part of the body being hit.
- **Uncorking a bottle of wine:** the simulation of this action is obtained by removing the piston of a big syringe previously filled with air.
- **Breaking bones:** usually, this is recreated by breaking into two halves a stick of celery in front of a microphone.
- **Bird wings flapping:** achieved by quickly wiggling a pair of leather gloves in front of a microphone.
- **Kissing:** this is done by wetting one's lips and then kissing the most hair-less part of one's forearm making sloppy kissing sounds.
- **Walking up the stairs:** there are lots of tricks¹ to perform this Foley effect. We chose the simplest and easiest method. Sitting on a chair wearing noisy sneakers a ceramics tiles surface must be hit in various ways and with different intensities.
- **Walking on summertime grass:** walking on or hitting with one's hands a 14 audiotape balled up. This Foley effect is shown in fig. 2.
- **Closing a sliding door:** this is achieved by making a roller skate slide on a piece of wood whose height is about 4 feet.

2.2 Recording Foley and real sounds

We selected our equipment according to the typical recording studio and technical equipment used by the Foley artist. We needed a low reverberant room as a Foley stage, and for this reason we ended up recording the majority of Foley and real sounds in *SaMPL*'s silent cabin - which has only 0.12 seconds of reverb time at a frequency of 1600 Hz. Some real sounds (such as: walking on summertime grass,

¹ Foley is an art, not Science. Therefore, for each Foley effect we might have different techniques to produce it. In this research we chose the simplest and the most traditional method to create Foley effects.



Figure 2. Microphone Sennheiser MKH 8020 and the 14" audiotape balled up.

walking up the stairs and closing a sliding door) needed a field recording session. The microphone used for the silent cabin recording session was a Sennheiser MKH 8020 - an omnidirectional microphone with a very linear frequency response curve and high sensitivity whereas for the field recording session we used a Sennheiser MKH 8040 – a cardioid microphone. Moreover, in order to achieve the best quality of audio files, we used an Orpheus FireWire audio interface. All sounds were recorded at a 48 kHz sample rate with a 24-bit resolution in order to avoid a downsampling process during the editing of the video. All the audio files that we recorded are available for the listening or the downloading on the web site <http://freesound.org> - by searching the username "stereostereo".

3. TESTS

3.1 Preamble

The type of test needed by the analysis process has been chosen on account of the goal that we wanted to achieve. Therefore, we restricted the main objective to the direct comparison between Foley and real sounds. There are several tests that we could use to this end, but none of them is a specific standard. For that reason, we adopted an international standard [3], which is the *MUSHRA*², albeit with some variations. Through this method and we organized two different tests. The first was an audio-only test whereas the second was an audio-video test. In this way we could be able to compare the data stored from each test involving thus two different sensorial modalities. Moreover we programmed two graphical interfaces that allowed us to store data automatically and to control the audio and video playback (using the Max/MSP/Jitter application). All subjects were supposed to use the same type of transducer. Therefore, the equipment used for all the tests was composed by a laptop, an USB audio Interface - the Edirol ua-101, and a pair of professional Sony Mdr-7506 headphones. Furthermore, we conditioned the audio

² MUSHRA is the acronym of "Multi Stimulus test with Hidden Reference and Anchors".

for each test somewhat, sometimes compressing or editing it and sometimes adding in and out fades. After that, we chose all the video fragments for the audio- video test and mixed their soundtracks with our recorded audio files. The chosen video clips were:

- Notting Hill 1999: ©Universal Pictures
- The Protector 2005: ©Eagle Pictures
- Edward Scissorhands 1990: ©Twentieth Century-Fox Film Corporation
- A Walk in the Clouds 1995: ©Twentieth Century-Fox Film Corporation

3.2 The MUSHRA method

This method was designed by the EBU project group to give a reliable and repeatable measure of the audio quality of intermediate-quality signals. It is a “double-blind multi-stimulus” test with both hidden reference and anchors³. In a MUSHRA test [3] the subject judges his “preference” for one type of artifact versus many others. Basically, she has to assess the impairments on “B” compared to a known reference “A” and then to evaluate “C” (“D”, “E” etc.) compared to “A”, where B, C, D, E are randomly assigned to a hidden reference, a hidden anchor and to the tested objects. The assessment is given according to the five-interval Continuous Quality Scale (CQS). It is a graphic scale that has a range from 0 to 100 and which is divided into five equal intervals that are: Bad, Poor, Fair, Good and Excellent.

3.3 Participants

It is very important that each participant has some experience in listening critically to the sound sequences, in order to reach results that are more reliable than those obtained with a non-experienced listener. We recruited forty experienced volunteers, twenty for the audio test and twenty for the audio-video test. All of them undertook a test which lasted less than fifteen minutes in order to avoid stress and fatigue.

3.4 Audio-Video tests

In the audio-video test ten different types of movie action were selected, defined as follows: Walking on grass, Kiss, Broken Hand, Sliding Door, Slap, Up the Stairs, Bird Flight, Bottle Cap, Double Kiss, Head and Arm Broken. For each of these actions there were three movies with the same picture but with different sounds. As a matter of fact, in one there was a Foley sound, in another one there was a real sound and in the last one there was an anchor sound⁴,

³ Generally, the anchor signal has a bandwidth limitation of 3.5kHz and is processed with a low-pass filter. Other anchor processings may include: reducing the stereo image or adding noise.

⁴ On our test, the anchor sound is quite different both the real and Foley sounds. If the listener evaluated the anchor sound as a very realistic sound, his results were discarded. The anchor helps to discriminate with sufficient accuracy the correct results.

for a total of thirty movies⁵ and an estimated total duration⁶ of ten minutes and thirty seconds. The subject was asked to evaluate how realistic was each video using the CQS scale. For all the videos that needed the anchor sound we used some audio samples downloaded from <http://freesound.org>. These sounds distributed by <http://freesound.org> are licensed through a Creative Commons license, which allows changing the sounds as we want provided we give credit to the author. A list of all the anchors used along with the action they represented follows:

- 85604_horsthorstensen_walk_mud01 Walking on the summertime grass
- 66073_joerhino_DVD_BREAKING_3 Breaking Bones
- 77534_Superex1110_Glass_Crush_7 Breaking Bones
- 26341_nannygrimshaw_London_Underground - Closing a sliding door
- 37162_volivieri_soccer_stomp_02 Walking up the stairs
- 40161_Nonoo_flobert1_20070728 Slapping
- 64401_acclivity_SwansFlyBy Sparrow
- 8000_cfork_cf_FX_bloibb Uncorking a bottle of wine

For the Kissing action we used the Foley sounds filtered with a high-pass filter in order to obtain a very bright and unreal sound.

3.5 Audio tests

The Audio Test included ten different types of action which were the same as those included in the Audio-Video test. Each of them contained two hidden audio files. One was the Foley sound while the other was the real sound for a total of twenty audio files and an estimated duration of four minutes and four seconds⁷. As in the video test, the listener was asked to evaluate from 0 to 100 how realistic was each audio clip compared to the action it represented. In this test there was no anchor sound as it was not essential to our main objective.

4. ANALYSIS OF DATA

4.1 Preliminary observations

Throughout the study of the data previously collected from the tests we wanted to be able to discern whether sound

⁵ Each video had a PAL 4:3 resolution whose dimension were 720 x 576 and an linear PCM audio codec - which had a sample rate of 48Khz and a depth of 24bit.

⁶ The estimate was made according to two hypotheses: the listener played each video at least twice and she usedat leasttwo secondsforeachassessment.

⁷ The estimate for the audio test was made according to two hypotheses: the listener played each sound at least twice and he usedat leasttwo secondsforeachassessment.

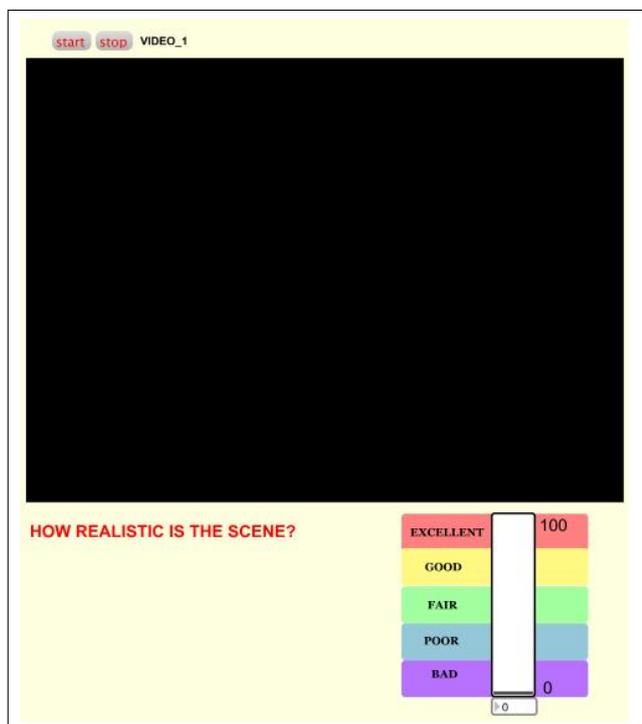


Figure 3. Interface details of the audio-video test. At the right bottom the slider of the CQS scale.

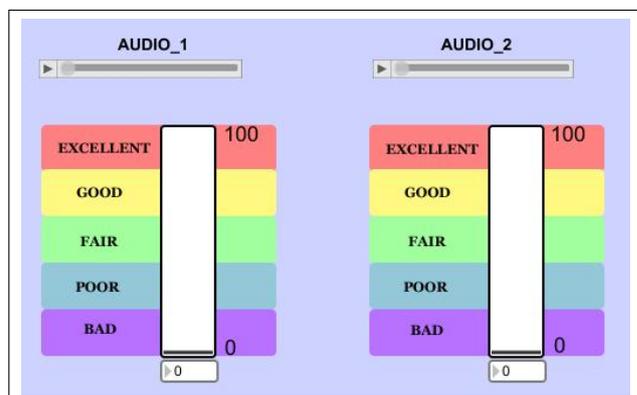


Figure 4. Interface details of the audio test. At the top there are two playbars for the audio playback.

| Screen action | Audio Test | Audio-Video Test |
|---------------------|-------------|------------------|
| Walking on grass | Real 76.15 | Foley 59.33 |
| Kiss | Foley 54.50 | Foley 71.76 |
| Broken hand | Foley 48.85 | Foley 54.25 |
| Sliding door | Real 70.15 | Foley 56.93 |
| Slap | Foley 56.85 | Real 58.42 |
| Up the Stairs | Real 70.30 | Real 52.25 |
| Bird Flight | Real 62.45 | Foley 50.57 |
| Bottle Cap | Foley 81.65 | Real 56.58 |
| Double Kiss | Foley 67.95 | Foley 76.33 |
| Head and Arm Broken | Real 63.90 | Foley 61.85 |

Table 1. Summary table of the highest means of each action for the audio and audio-video test.

effects are always better than real sounds. The main objective was the direct comparison between the average result of the different evaluations given to Foley and real sounds. Therefore, we have analyzed all the data with the ANOVA statistic method. A table with the direct comparisons between the highest means of the two tests follows.

First of all, we will analyze the summary table 1. We can observe that for the audio test the participants judged as realistic sounds five Foley sounds - the 50% of the total. On the other hand, the results for the Audio- Video test are quite different. In fact, the subjects preferred the 70% of actions with Foley sounds. This analysis was done without the ANOVA and for this reason on the next chapters we will do an accurate analysis with this method for each test.

In each test 50% of the subjects were musicians. Consequently, it is interesting to understand if the evaluations differ between musicians and non-musicians. To this end

we performed a separate ANOVA analysis for each of these two categories. But neither the audio tests nor the audio-video ones showed significantly different behaviours between musicians and non-musicians. This is directly attributable to many reasons. As a matter of fact, the tasks were not musical tasks but they involved everyday sounds, and everyone today has some experience with them even if non-musician. Furthermore, we did not ask to judge some musical features, but we only asked to evaluate generically their correspondence with a real sound. We will do a final consideration. Some of the Foley effects are very difficult to perform, because they need a deep experience in order to recreate them. Even if we trained in performing Foley sounds for two weeks, some Foley effects were not reproduced perfectly because we are no Foley artists. Definitely, this might have influenced the subjects in judging the life-likeness of each task.

4.2 Analysis of the audio data test

Each subject compiled a form questionnaire with different filling in the following fields: first name, gender, age, education and other questions such as “Do you usually listen to music?” and “Do you play any instrument? If so, how long have you been playing it?”. With these questions we were able to profile each participant as well as checking that they had a certain experience in listening critically to sound sequences. Mainly, we are going to analyze the principal ANOVA values, which are: the means, the F factor and the p-level. The F factor is simply the ratio of the two variance estimates. In the data that we will show next we had to omit all the results where the p-level was higher than the value 0.05 as they implied that the assumption that Foley sounds are different from real sounds was not true. In the audio test the action that had a p-level lower than 0.05 were:

- Walking on the summertime grass, which had a p-level of 0.001 an F factor of 12.71 and an average of 76.15 for the real sound while for the Foley sound is 53.05.
- Kiss, which had a p-level of 0.039 an F factor of 4.55 and an average of 54.50 for the Foley sound

| Screen action | Preference |
|----------------------------|-------------|
| Walking on grass | Real 76.15 |
| Kiss | Foley 54.50 |
| Uncorking a bottle of wine | Foley 81.65 |
| Passionate kisses | Foley 67.95 |

Table 2. Summary of the preferences for the actions with a lower p-level for the audio test.

and 37.30 for the real one.

- Uncorking a bottle of wine, which had a p-level of 0.018 an F factor of 6,08 and an average of 81.65 for the Foley sound while for the real sound is 65.95.
- Passionate kisses, which had a p-level of 0.020 an F factor of 5.85 and an average of 67.95 for the Foley sound and 47.25 for the real one.

The list above shows that only 40% of actions were completely distinguished. This is quite different from the results obtained with an average analysis (50%). Now we will draw up a list of the preferred sounds for each action with a low p-level for the audio test - referring to Table 1.

In our hypothesis the lack of important differences between Foley sounds and real sounds might be related to the fact that each sound expresses the action that it represents even if it has a different source. Therefore, we can assess if Foley sounds are equivalent to real sounds only with a numerical analysis of the signal, such as MFCC analysis or Onset detection.

4.3 Analysis of the Audio-Video data test

As the audio test, also the Audio-Video test lasted a week and employed twenty participants. Each of them compiled a questionnaire with the following fields: first name, gender, age, education and other questions such as “How often do you usually go to the movies?”, “How many movies do you watch in a year?”, “Do you play an instrument?”. The purpose of the questionnaire was the same as the one of the audio test. Before analyzing the scores of the test we have to do a preliminary observation. First of all we discarded all the results from the listeners that evaluated the anchor sound as a very realistic sound. As a matter of fact, the anchor helped to discriminate with sufficient accuracy the participants that were not able to distinguish between different sound artifacts. Then we calculated the One-Way ANOVA for each action expected for the “Hand and Arm broken” because in this action twelve listeners out of twenty evaluated positively the anchor video. That might be due to the striking resemblance between the anchor sound, the Foley and real sound. Finally we kept the most significant data, which had a p-level lower than 0.05:

- Walking up the stairs, which had a p-level of 0.0196 an F factor of 13.29 and an average of 52.25 for the real sound while for the Foley sound is 28.50.

| Screen action | Preference |
|-----------------------|-------------|
| Walking up the stairs | Real 52.25 |
| Kiss | Foley 71.76 |
| Passionate kisses | Foley 76.13 |

Table 3. Summary of the preferences for the actions with a lower p-level for the audio-video test.

- Kiss, which had a p-level of 0.0002 an F factor of 13.12 and an average of 71.76 for the Foley sound and 36.29 for the real one.
- Passionate kisses, which had a p-level of 0.00001 an F factor of 12.97 and an average of 76.33 for the Foley sound and 45.11 for the real one.

According to us, the lower score for the Foley sounds on the “Walking up the stairs” is due to the fact that the real sound has more features than the Foley sound, such as shoes noises or deeper reverberations, which allow to recognize it better. In this test only the 30% of actions were discriminated:

We can thus assert that there are no significant differences between a movie with Foley sounds and a movie with real sounds. As Michel Chion proposed in his book [4] the audio on a movie is only an added value to the pictures of the screen. We can demonstrate this hypothesis only through other tests that employ a higher number of subjects. However, even if our results highlight the fact that there are important differences between the audio and the Audio-Video test, it is not the main purpose of this paper to understand the psychological relationship between audio and video [5].

5. CONCLUSIONS

The main purpose of this paper was the direct comparison between Foley effects and real sounds, in order to understand if Foley sounds are always better than the real ones. What appears quite clearly observing and analyzing the findings is a similarity in judging preferences between real sounds and Foley sounds. As a concluding remark, we highlight the fact that the results of the tests demonstrate the participants partial discrimination ability between Foley effects and real sounds even though the sounds are remarkably different from each other. The final outcome of these experiments indicate a path of wider investigation on the world of Foley and everyday sounds. Therefore, future work will involve:

1. Further recording sessions of Foley sounds, so as to subdivide them in categories such as: impulsive sounds, continuous sounds, rhythmic sounds and so on.
2. Move from the realistic investigations of sounds to the evaluation of their expressivity.
3. A deep numerical analysis – with MFCC, centroid and many other methods – of the real and Foley sounds

in order to find out similarities or differences between them. In this way we aim at discovering which are the features that allow to recognize and better emphasize a sound.

4. Repeat both tests with more participants for each one and also with an equal number of female and male evaluators.
5. Understand how Foley effects exaggerate important acoustic features. These are the basis for being able to create a database of expressive sounds, such as audio caricatures, that will be used in different applications of sound design such as advertisement or soundtracks for movies.

Acknowledgments

We would like to thank Nicola Bernardini for his expert and enthusiastic support, Sergio Canazza for his useful advice for the practical tests and the *SamPL* laboratory for having provided us with all the technical equipment needed for the recordings.

6. REFERENCES

- [1] M. L. Heller and L. Wolf, "When sound effects are better than the real thing," *Journal of the Acoustical Society of America*, no. 111, p. 2339, 2002.
- [2] A. V. Theme, *The Foley Grail: The Art of Performing Sound for Film, Games, and Animation*. Elsevier, 2009.
- [3] G. Stoll and F. Kozamernik, "Ebu listening tests on internet audiocodecs," *Ebu Technical Review*, june 2000.
- [4] M. Chion, *L'audio-vision. Son et image au cinema*. Paris: ditions Nathan, 1990.
- [5] A. Kohlrausch and F. V. de Par, "Audio-visual interaction: From fundamental research in cognitive psychology to (possible) applications," *Human Vision and Electronic Imaging*, pp. 33-44, 1993.

ROBOTIC PIANO PLAYER MAKING PIANOS TALK

Winfried Ritsch

Institute for Electronic Music
and Acoustics Graz
ritsch@iem.at

ABSTRACT

The overall vision of a piano which can talk, a piano that produces understandable speech playing notes with a robotic piano player has been developed as artwork over the last decade. After successfully transcribing recorded ambient sound for piano and ensembles, the outcome of this mapping was applied by the composer Peter Ablinger in his artwork, which explores the auditory perception in the tradition of artistic phenomenologists¹. For this vision a robotic piano player has been developed to play the result from the mapping of voice recordings, by reconstructing the key features of the analyzed spectrum stream, so that a voice can be imagined and roughly recognized. This paper is a report on the artistic research, mentioning different solutions. The output as artworks will be referenced.

1. INTRODUCTION

The basic idea from Peter Ablinger was to create a kind of phonorealism [1], which can be compared to photo-realist painting in visual arts. There are also other aspects than the technical aspect of the "Quadraturen" for his series of works in this area to be considered, which has been covered by musicologists more precisely [2] and not described here. For his ideas and first experiments with half-tone filters, the aesthetic principles lead him to use the analysis data of recorded sounds as a base material for his compositions, leading to the one of the most challenging disciplines, using voices as reconstruction of voices in instrumental domains, like the piano domain.

On the other side, a piano is a machine producing velocity depended sound on key press, which is rich on overtones and also includes the noise of the attack. So it seemed at a first glance, that it is impossible to reconstruct a voice with a piano.

Understanding human voice is an essential capability of man and trained from childhood. It is what we are able to recognize easily and fast, even out of noisy audio material and surroundings. So discrimination of unwanted sound

¹ phenomenologist in the sense of the work done by Alvin Lucier, who himself described him as phenomenologist, which is not a scientist, but uses phenomenas as a material for his art.

on reception of voices is very well implemented in the psychoacoustic processing in the brain. On the other side we are most sensible in the recognition of small differences in speech. So speech reconstruction needs to focus on the key features of speech very precisely, but is robust to introduce additional spectral noise, such as mechanics-noise and attack spectra and we are able to mask them during speech recognition.

Looking deeper in this aspect, we recognized, there is only a need to trigger the key features for the voice reception, which are the impression of vowels and some unvoiced pitches. Like on opera singing, the unvoiced parts can be easily overheard without losing the context, they are masked by volume, superseded by music and can be imagined and reconstructed to understandable words and text during reception. The reconstructed voice is recognized more and more clearly with repetition showing, that the human brain can learn to understand a piano talking.

It is also the ambiguity between the recognition of the sound as piano and as understandable speech, which is intended and makes this work unique. To get over this border to both sides during reception as voice or piano is an important aspect for these compositions.

To render the mapping on an real piano implies the use of a computer controllable player-piano. Since none of the available player-pianos could play this, the Autopianoplayer was developed, to drive the pianos to their limits, inspiring new art work.

2. QUADRATUREN - MAPPING

The mapping of recorded sound in other domains has been developed for the idea of phonorealism in the series of Quadraturen ("Squarings") at the Institute of Electronic music during first studies in 1996 from Peter Ablinger and implemented by Thomas Musil. Squaring refers to raster recorded sound in time and frequency. This led to his cycle of works for symphony orchestras, sound installation to compositions for computer-controlled player piano.

Actually however, my main concern is not the literal reproduction itself, but precisely this borderline between abstract musical structure and the sudden shift into recognition - the relationship between musical qualities and "phonorealism": the observation of "reality" via "music". (Peter Ablinger [3])

He came up with the idea of the two dimensional raster of the recording in time and frequencies, which he recognized

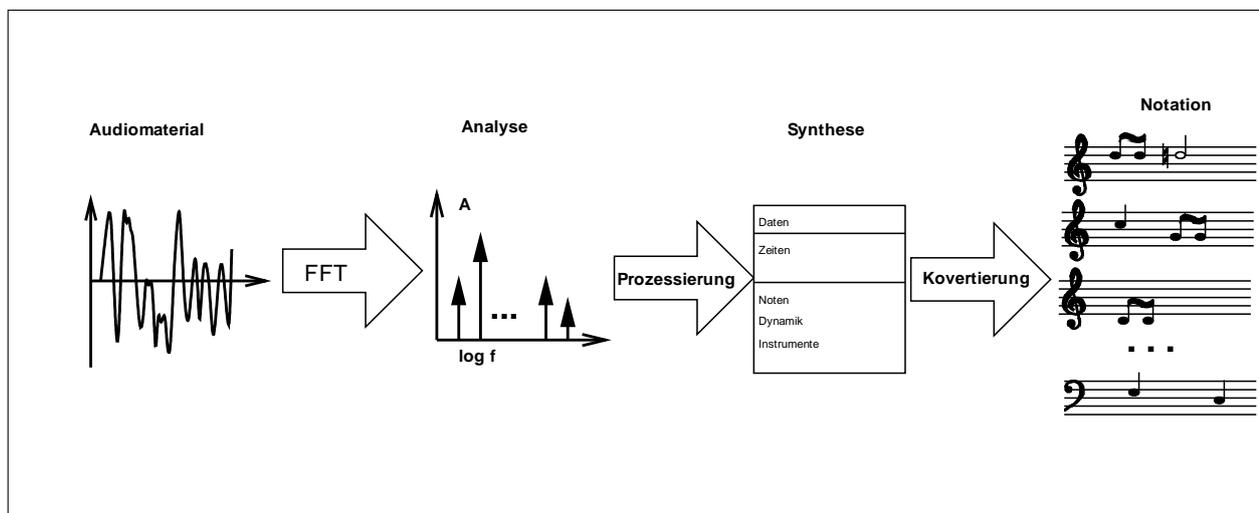


Figure 2. Analysis path.

as squares. Technical speaking it is the quantization in time grains holding the spectral power within each frequency ranges, for instrumental reproduction mostly half-tones.

2.1 Analysis

To get sufficient results, several experiments with different analysis methods has been done over the years of development. Starting with FFT, where the power of the bins are mapped to semitones like half tone filters, trying constant-Q transform[4] and wavelet analysis[5]. They have shown to work all properly with different qualities:

The plain FFT tends to wrong results in distribution in lower octaves, especially when tested with a sinus sweep as input. So for lower frequencies additional larger FFTs has been implemented, realizing a banded analysis but rising problems with the resolution of time. So a kind of constant-Q was used instead, leading to problems in mapping to equal time slices.

Wavelet analysis did not result in better transformations compared to FFT and constant-Q and was not used in later versions of the mapping software, since it was harder to handle of the data and lost the plausibility.

But overall, the algorithm of transformation, had not much influence in the result of perception of a voice via piano, especially using a real piano, instead of piano-simulators or synthesizer in comparison to influence of the extraction of the key features. As an effect, mostly the simpler to use FFT algorithm has been used, applied on different frequency bands. Also the preprocessing of the audio data before applying the frequency transformation had an great impact, and also was easy to accomplish by the composer listening to the spectral distribution. Using constant-Q was chosen on later pieces mostly for lower octaves in combination with the FFT algorithm. This was chosen by composer for different purposes.

The process for art pieces like “Deus cantando“ or “A letter from Schnberg” the composer himself selected the recordings and did a preprocessing with common sound editors, to get a sound composition which is usable for the mapping process. Segmenting this audio data in pieces to

work with different parameters for different parts and frequency bands like octaves. Therefore it is hard to judge the analysis step of the process for the artistic purpose.

2.2 Extraction of Key Features

The more sophisticated part was the extraction of notes from the analysis, to be played by the piano for the reconstruction of the targeted voice in respect of the best recognition. Also the more significant notes has to be selected in favor for a playable piece. All this can be controlled by parameters for composition. One of this is the number of parallel played notes in different bands. This was not only very useful for precise reconstruction, but also for the aesthetic and style of the artistic expression, which goes beyond the technical aspects.

Like in audio compression algorithms, several rules has to be implemented to accomplish the suppression of not relevant notes. The most important rule is the masking of successive notes for the same key, selecting the most significant one over time. Also the note neighborhood has to be taken into account, for the decision of the selection of the most significant note. This was needed not only for the piano to be played not beyond its limits, but also reducing unnecessary noises of not relevant notes. Unnecessary key presses can be in rising the noise floor a deal breaker. The exclusion of this notes is also controllable by parameters to vary the composition.

As a next rule, the processing and choosing of the note length in combination with concatenation of successive notes became skillful. It could be accomplished that he spectral elements of a vowel, which are present over longer time in the analysis data, can be used for reconstruction reducing onset noises. Here also predecessor notes with lower velocity are taken into account. Since it seems, that during the reconstruction of the voice in the brain, also delayed spectra elements are effective. This was percept by hearing to it, but was not proven and could be a subject for further research.

The mapping of the power of the semitones bins to velocity of the piano has shown to be an important aspect for

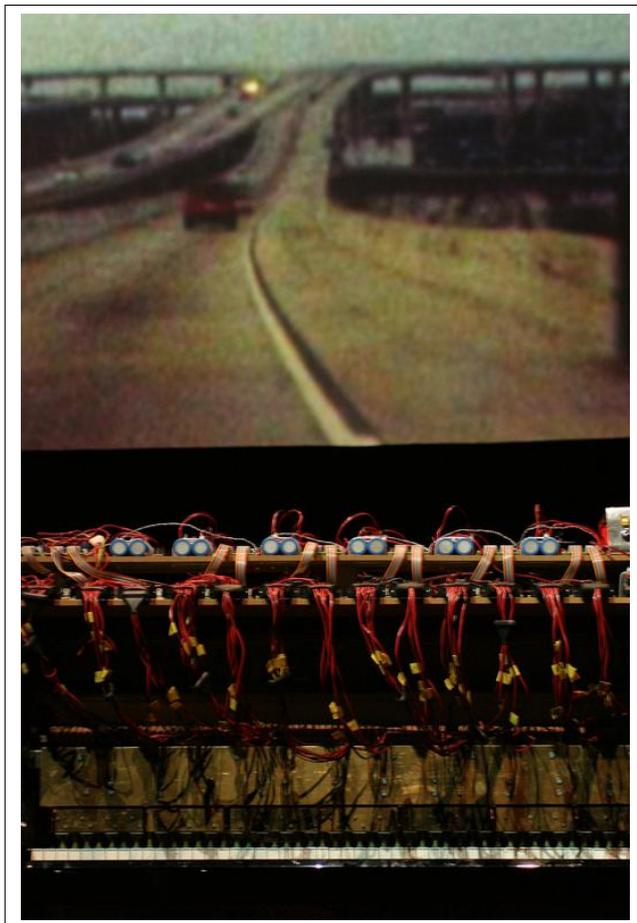


Figure 1. Playing the noise of a road with automatic piano player for the opera “Stadtofer“ of Peter Ablinger performed 2004 in Graz.

the reception. This is mapping is also accomplished in an second step within the interpretation of the notes by the robotic piano player. Since interpretations also depends on different situations of playback, this has to be modified on stage.

Other rules like the avoidance of neighborhoods in notes, or the amount of parallel notes over the frequency range, the portamento within a analysis segment, the isolation of notes representing fundamental pitch was added for compositional purposes and cannot be evaluated only from a technical standpoint to enhance the perception of voices. It is also hard to describe them here precisely, since they are part of the artwork applied by the composer after the software development.

So it came out that the interlock between software and composition was very high and a general mapping software could not be finished nor released and is changed very often for new pieces, since parts of the composition is coded in this mapping.

2.3 Rendering

For the reproduction targets files are produced or a special version for real-time rendering is used. Files can be MIDI

standard files², for more precise data message-files interpreted by Pure Data [6] and also music-xml files for scores are foreseen.

Real-time output for the piano player, since some of the art works needs the real-time operation of the mapping described above, can be achieved by a special version of the software. This version uses a limited set of steps and therefore parameters, excluding those which introduces to much delay. Especially some steps has to be skipped in note the processing and a lower frequency resolution are needed for a faster analysis.

These leads to software tools, which must be handled by the composer, for doing new pieces and studies without technician support.

The rendering during the composition process is mostly done by software simulators, which does not really reflect the limits of a real piano. They differ in repetition rates at different velocities and mostly produce less noise in their attacks. In a second step the Autopianoplayer is targeted for corrections, because of the previously listed limits. On some pieces, also the different pianos with different behavior has to be considered for a correct performance.

3. AUTOKLAVIERSPIELER

A massive frame with 88 electromechanical finger, which are moved by solenoids, is mounted on a keyboard. controlled by micro-controllers, which are driven over a dedicated computer, the Autoklavierspieler can be controlled over Network, MIDI files and real time generated music and has been constructed at the Atelier Algorhythmics³. As a reference some player pianos has been analyzed and the idea of a robot piano player sitting in front of a piano, has been taken from the idea of Trimpins player piano [7]. This fits also the performance purposes, since pianos are widely spread and hard to transport.

When the first piece of this serie “Zeit im Bild 2” has to be performed in 2003 on a festival, we tried to play it on severnal different player pianos, but failed. The Yamaha Disklavier could only play 16 keys in parallel and only with 4 different parallel velocities. Marantz player could not play different velocities and the “Bsendorfer Computerklavier“, precessor of the Bsendorfer Ceus system invented 2006, with his Zilog controllers always crashed after few seconds if played as fast as we needed and more than 32 keys in parallel.

A major problem to all of them, seemed to be, that they do not have enough power, especially power supply, to play that many parallel keys at high velocities. Since quite all of the western music literature for pianos can be played on these, nobody thought on the need of this for our application. Also the repetition rate was mostly to slow and repetitions at various velocities was not clean enough.

Since Trimpins automata was not available any more for being adapted, we have been forced to develop our own robot piano player, specialized for this purpose. It had also the advantage taking the needed aesthetics of performance

² The Music Instrument Device Interface -file standard, was the facto standard exchange for notation software imports

³ Atelier Algorhythmics Graz: <http://algo.mur.at/>



Figure 3. Millitron Autoklavierspieler 2010.

into account, but the disadvantage to do a lot of work. Also it should be affordable and usable over a longer period playing unattended.

The first version of the Autoklavierspieler, a robotic piano player named **Kantor**, powered by 1,5 kW and a weight of 120kg, was constructed 2003. As also an artistic research on robotic electromechanical instruments for extreme performances, the main target was the realization of algorithmic compositions for 88 finger, focused, but not only, on the work of Peter Ablinger. The initial project was interpreting audio recordings on the piano for the series "Quadraturen III".

With the need of dialogs in the compositions and more performances in Europe, an additional new Autoklavierspieler was needed, especially for opera production "Stadtopfer" from Peter Ablinger. **Millitron**, as this one was named, could be optimized from previous experiences. It has half the weight, a dedicated micro-controller board, the *algopic* and *algofet*⁴, it played more precise, was better usable, especially for quiet pieces and therefore had a better "piano-forte" dynamics. Exchanging the use of a hold circuit, not using PWM Modulation during the hold phase of a key, eliminated the high frequency noise from the solenoids.

Rhea was developed with the focus on even faster repetition, better dynamics for pianissimo, easy transportation and fast setup. It also was a first prototype series for a performance for 12 robotic piano player⁵. The new electronics developed for this, should enable much finer calibration and better adoption on old imprecise pianos and an easier control over Ethernet. Furthermore Rhea should be a first series for reproduction as open hardware, enabling others to build and handle the robot pianoplayer.

⁴ Based on PIC16F877 and two stage FET solenoid driver this circuits wa also used in a lot of other artwork, see [8]

⁵ See Maschinenhalle performance with Bernhard Lang and Christine Gaigg <http://algo.mur.at/>

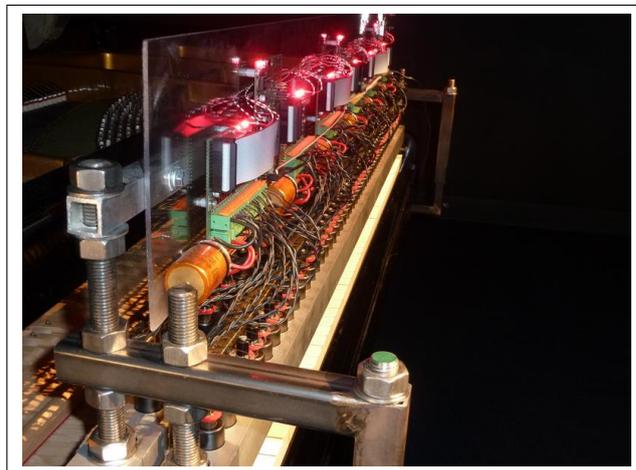


Figure 4. Rhea Autoklavierspieler 2010.

3.1 Finger Strike

The idea was, that every key is playable individually, driven by a force envelope which can be adjusted to any piano, also elderly imprecise ones. It has shown that a punch of up to 3 kg per finger is needed for full velocity. The repetition rates are restricted by the mechanics of the piano and can go as low as 50 ms on pianos with good repetition mechanics. Grand pianos seems to be general faster and have better dynamics.

In the first phase of the strike the maximum force accelerates the finger, where the acceleration is more important than the actual force at the end of the strike, since velocity correspond to sound intensity. If pushed soft, like pianissimo notes, also the attack shape has to be adapted since a bouncing effects has been detected. With the new electronics it is also possible to press the key without a pluck sound of the hammer. After the attack phase, the hold phase is initiated. At this phase the key should not be pressed to the limit to enable a faster repetition. The release phase is enhanced with the reset spring of the solenoid. All of this leads to a complex system, where parameter depends on each other and these parameters has to be calibrated for each key by software.

3.2 Electronics

The electronics for Rhea uses the micro controllers boards Escher, with Ethernet and serial interfaces driving FET amplifiers for the custom made solenoids. Escher has been developed with the Autopianoplayer in mind.

3.2.1 Escher Board

The open hardware Escher is based on the motor control DSP-microcontroller dsPIC33F708MC from Microchip, a 16 bit controller with 160MIPS and dsp-unit and also hosts an Ethernet controller. Escher boards provides more than 48 I/O pins including 12 channel hardware PWM driven by 20kHz with 10-12 bit.

Each Rhea needs 3 Escher, where one master receives OSC⁶ commands over Ethernet, driving the piano-player

⁶ Open Sound Control over Ethernet UDP



Figure 5. Escher Controller.

and forwards OSC messages to the 2 slave boards via fast serial connection. So each player piano has one IP address and status can be accessed during playing.

3.2.2 EscherFET

To drive the solenoids each Escher drives two EscherFET boards with each 16 channel of FET amplifiers.

This enables the individual control of the solenoids up to 100 MHz switching frequency, and each of 4 A constant and up to 20 A peak load, if the power supply is strong enough. After using high frequency PWM the speed of the piano could be improved from 80 ms up to 50 ms repetition rate on fast pianos. It was crucial to get a fine control over amplitude of each key for a better reconstruction of the formant spectra.

3.2.3 Solenoids

The custom designed solenoids for Rhea have been constructed for optimum power on the needed 10 mm stroke length at maximum 30 V and has been manufactured for this purpose. They originate also a simpler mechanical construction and therefore reduce the overall weight. Driving them by 16 times of the power, of the allowed 100 % duty cycle, up to 20 ms increases the acceleration enough and they can be operated up to 70 degree Celsius temperature.

Another critical task was to silence them, especially on the stroke to the key. This was done via Kashmir felt proven to work for pianos for centuries. It showed that

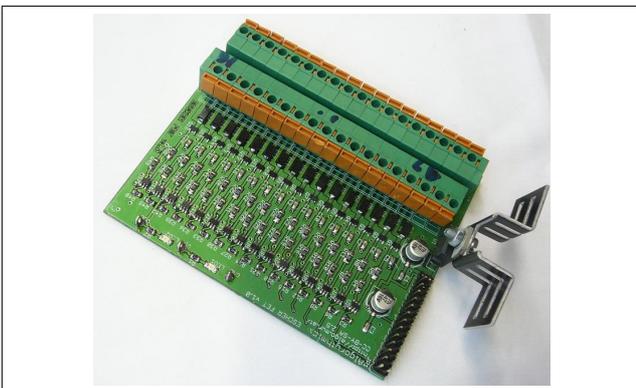


Figure 6. Escher Fet Amplifier Board.

the felt reduces a lot more playing noise than industrial offered damper, so the mechanical finger also was made of non magnetic material brass with felt as finger tips.

3.3 Software

The overall system was designed for realizing performances playing within ensembles, synchronized playing to video projections, installations with automatic operations and simple interfaces and standalone solo performances. Also a fast set-up and individual calibration to different pianos and grand pianos has been implemented.

Making the piano speak needs three stages: A stage for the composition phase, which is done mostly off line, the performing software implementing the art works, integration in other environments and the micro-controllers firmware as standalone software, representing the robot player. In figure7 the structure and parts of the software system is shown.

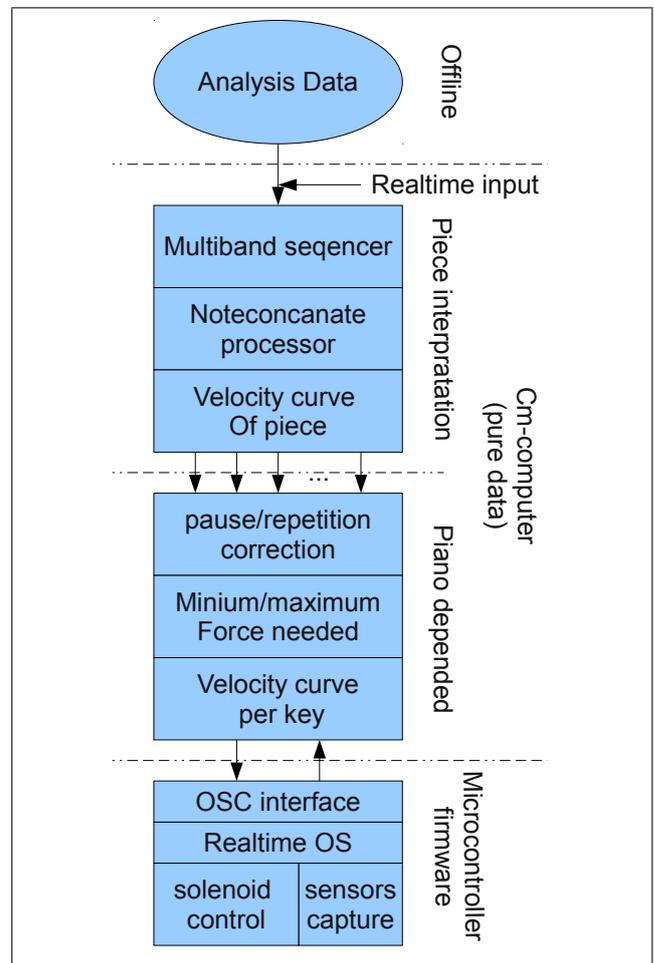


Figure 7. Software Structure.

3.3.1 Composition Tools

For the composition a set of programs, which are operated on command line interpreter was developed at the IEM for the art work of Peter Ablinger, mostly by Thomas Musil.

3.3.2 The Player

For playing with the robot piano player the programming language Pure Data was used to enhance fast prototyping and adapt and integrate easily to other projects and guarantee a fast extension to new needs. This software can also be used as a framework for usage in other projects and runs under Linux. Following task has to be matched:

3.4 Escher Realtime OS

The firmware of the player Kantor and Millitron was written in assembler language, to get proper performance on the 16F877 micro-controller. With Escher, also C with inclusion of has been used in combination with assembler code for time critical tasks.

The firmware was realized as a small real-time OS, which has to glue the OSC commands to parallel running and independent hardware control threads.

4. OUTCOME

Like mentioned at the beginning, the motivation for the robot piano player, was transcription of recordings in the domain of pianos, following a special aesthetic principle, so the outcome can be evaluated on the pieces and their performances at various festivals and in installations. Here the two major works as examples for the speaking piano from Peter Ablinger are described.

4.1 Audioanalyse / Die Auflsung / Freud in England / Le Grain de la Voix

For computer-controlled piano and video text, done in 2006. Maybe the only recording of Freud from 1938, where he already suffered on tongue cancer and explained why he immigrated to England. Therefore the beginning of the piece was underlaid with with the noise where over time the the voice appears and afterwards is more and more thinned out so that the voice ends monophonic. Here clearly the border recognizing voices can be heard depending on the trained ear for this material. With the projection of the text behind the piano the recognition of the heard speech is enhanced. The piece was commissioned commissioned by the MAK Center Wien.

4.2 Deus Cantando

This piece was commissioned for the opening of "World Venice Forum 2009" for the demand of an international environmental courtyard, where the declaration written by the Dalai Lama was spoken by a young boy from Berlin. Here the focus was the text, produced with a multi-band analysis. This piece is presented since March 2011 in the standard exhibition of the ars electronica center in Linz.

5. CONCLUSIONS AND FURTHER OUTLOOK

After successful proven with mentioned artworks, that voices can be understood played on pianos, the aesthetics of music done with this technique was the important step in this field. What first was foretold never will work, that a piano



Figure 8. Installation at Ars Electronica Center Museum 2011.

can talk and was first only a vision, became a successful artistic concept and can be surely enhanced furthermore.

Acknowledgments

I want to thank all who helped developing this pieces of hardware especially Thomas Musil for his analysis program, Peter Ablinger for artwork and the many helpers implementing the robot piano player and hardware beside their work.

6. REFERENCES

- [1] P. Ablinger, *Phonorealism*. <http://ablinger.mur.at/phonorealism.html>, accessed 30.3.2011, online article.
- [2] C. Mello, *Mimesis und musikalische Konstruktion*. Shaker Verlag, Aachen 2010, 2011, ch. Zwischen Abbild und Selbstreferenzialitaet: Mimesis und Rauschen bei Peter Ablinger, dissertation.
- [3] A. Peter, *Peter Ablinger - Quadraturen*. <http://ablinger.mur.at/docu11.html>, accessed 30.3.2011, online article.
- [4] J. C. Brown, "Calculation of a constant q spectral transform," in *J. Acoust. Soc. Am.*, 89(1), 1991, pp. 425–434.
- [5] P. G. R. Kronland-Martinet and S. Ystad, "Modelling of natural sounds by time–frequency and wavelet representations," in *Org. Sound*, 2(3), 1997, pp. 179–191.
- [6] M. Puckette, "Pure data," in *Proceedings, International Computer Music Conference.*, San Francisco, 1996, pp. 224–227.
- [7] Wikipedia, "Trimpin — wikipedia, the free encyclopedia," 2011, [Online; accessed 27-April-2011]. [Online]. Available: <http://en.wikipedia.org/w/index.php?title=Trimpin&oldid=421091526>
- [8] W. Ritsch, "Algopic - algofet," accessed 30.3.2011, online article. [Online]. Available: <http://algo.mur.at/projects/microcontroller/algopic/algopic>

SOUND SPHERES: A DESIGN STUDY OF THE ARTICULACY OF A NON-CONTACT FINGER TRACKING VIRTUAL MUSICAL INSTRUMENT

Craig Hughes

Michel Wermelinger

Simon Holland

Computing Department and Centre for Research in Computing
The Open University
Walton Hall, Milton Keynes MK7 6AA, UK
ch3375@student.open.ac.uk, {m.a.wermelinger, s.holland}@open.ac.uk

ABSTRACT

A key challenge in the design of Virtual Musical instruments (VMIs) is finding expressive, playable, learnable mappings from gesture to sound that progressively reward practice by performers. Designing such mappings can be particularly demanding in the case of non-contact musical instruments, where physical cues can be scarce. Unaided intuition works well for many instrument designers, but others may find design and evaluation heuristics useful when creating new VMIs. In this paper we gather existing criteria from the literature to assemble a simple set of design and evaluation heuristics that we dub *articulacy*. This paper presents a design case study in which an expressive non-contact finger-tracking VMI, Sound Spheres, is designed and evaluated with the support of the articulacy heuristics. The case study explores the extent to which articulacy usefully informs the design of a non-contact VMI, and we reflect on the usefulness or otherwise of heuristic approaches in this context.

1. INTRODUCTION

With traditional acoustic musical instruments, there is a strong coupling between the playing gestures and the mechanisms that produce the sound: these two areas of concern exert powerful constraints on each other. By contrast, in the case of Virtual Musical Instruments (VMIs) [4,6] interaction gestures and sound design are, in principle, orthogonal. Consequently, the design of VMIs generally requires careful explicit attention to the mapping from gesture to sound manipulation.

Despite the freedom thus afforded to VMI design, a review of sources such as Mulder [4] and the Taxonomy for real-time Interfaces for Electronic Music performance (TIEM) [6] suggests that the majority of VMI controllers nevertheless rely on physical interaction between player and instrument. That is to say, many if not most VMI designs involve exerting a tangible force on a musical instrument in order for it to produce a sound. This is unsurprising. Research in areas such as Physicality in Hu-

man Computer Interaction [12] and embodiment in Music Interaction Design [11] suggest various routes by which physical contact offers rich affordances for designers and performers.

However, some VMIs are controlled without physical contact interaction [4,6] and instead rely on the proximity, or movement (gestures), of parts of the body. Non-contact VMIs raise interesting challenges for designers and performers alike in creating satisfying interaction designs for music making. The present case study explores some of these challenges.

Interaction designs for VMIs are often arrived at intuitively, and in the hands of many digital luthiers this is an optimal approach. By contrast, some instrument designers may find design and evaluation heuristics [13] useful when designing and evaluating new VMIs, particularly in focusing the process of iterative design. This paper reports on a design case study in which an expressive non-contact finger tracking VMI is designed and evaluated using a candidate set of design heuristics and evaluation heuristics for VMIs. These heuristics, which we have labeled *articulacy* (defined in section 3.1 below) are derived from design considerations from the literature [1,2,5,7]. The present case study affords a first look at how design and evaluation heuristics such as articulacy can inform a non-contact VMI design, and a preliminary reflection on the usefulness of such heuristics for this purpose.

2. BACKGROUND

Until recently, hardware to support finger tracking has been expensive and confined to specialist use. However, Lee [3] showed an accessible and affordable finger tracking technique utilizing the Nintendo Wii Remote controller (Wiimote) for the Nintendo Wii game console. He cleverly exploited the Wiimote's built in infrared camera and simple Bluetooth connectivity, demonstrating how to implement a finger tracking application. More recently, Microsoft's Kinect introduced another low cost opportunity for developing body-tracking applications. More generally, Vlaming [8] identifies a wide range of motion capture techniques and systems. The present case study focuses on the design and evaluation of a new non-contact virtual musical instrument, Sound Spheres, which is aimed both at musicians and novices, and which uses Lee's finger tracking motion capture technique for its gestural interface.

3. DESIGN CRITERIA

As already noted, because interaction gesture and sound design may vary independently in Virtual Musical Instruments, the designer must generally pay explicit attention to the mapping from gesture to sound manipulation. VMIs that successfully appeal to performers involve rich and subtle constraints on the connections between gesture and sound. However it is hard to characterize explicitly the nature of these constraints. Such characterization is particularly challenging in the case of non-contact VMIs where interaction with physical objects is absent. The HCI literature suggests many candidate design considerations, some relatively simple, such as clarity of feedback [13], and others more complex, such as appropriate exploitation of physicality [12] and systematic consideration of issues of embodiment [11]. For the present purposes, simple considerations are needed, suitable for guiding the design and evaluation of non-contact VMIs.

The Thummer Mapping Project [5] identified four common physical instrument variables (pressure, speed, angle and position) that control instrument dynamics, pitch, vibrato and articulation. In a later study Paine [7] re-iterated these control parameters as important factors for the design of new musical interfaces. Jordà [1,2] described other factors considered important to the consideration of a good musical instrument, suggesting playability, progression (learning curve), control and predictability. He also suggested that the balance between challenge, frustration and boredom must be met. Ferguson and Wanderley [9] highlighted reproducibility as one more important factor for digital musical instruments, suggesting that musical instruments that allow a performer to be expressive must also permit a performer to imagine a musical idea and be able to reproduce it.

In order to provide a simple set of heuristics for the design and formative evaluation of a non-contact VMI, we have borrowed and adapted these various considerations. Note that the simplicity of the approach reflects our preference in the present case for a light-weight methodology. For heavier-duty methodologies, see section 10.

3.1 Articulatory heuristics

We will consider the articulatory of a non-contact VMI to refer to (a) the degree to which pressure, speed, angle and position can be used to control the instrument and (b) the degree to which the design achieves playability, progression, control, predictability, reproducibility, and balance between challenge, frustration and boredom.

This set of considerations can be applied straightforwardly to VMI design simply by using them as a checklist of desirable properties. Similarly, they can be applied to formative VMI evaluation by considering, or measuring (see section 6), the extent to which they are achieved in a given design. Despite the extreme simplicity of this method, closely related approaches have been found useful in HCI design elsewhere. Indeed, our approach broadly echoes such approaches as Molich and Nielson's [14], which has been widely applied to user interaction design in general.

The purpose of this paper is to present a design case study, which includes a simple formative evaluation using eight test subjects, to explore the extent to which the articulatory approach, or similar approaches, might usefully inform the design and evaluation of non-contact VMIs.

4. OVERVIEW OF SOUND SPHERES

The Sound Spheres VMI is controlled solely by the movement of the musician's fingers in the air. Unlike some finger tracking applications, complex finger gestures are avoided and only the finger tips are used. Highly reflective tape placed on the fingertips reflects infrared light to the Wiimote's infrared camera (figure 10). The Wiimote then passes data concerning the positioning of the fingertips to the Sound Spheres VMI software.

The position of the finger tips is represented on the user interface (figure 1) as small spheres (*tracking spheres*). Only four fingertips can be simultaneously tracked with the Wiimote's infrared camera and hence this poses a limitation of up to a maximum of four tracking spheres. The movement of the tracking spheres is used to trigger sounds through collision with a set of fixed larger spheres (the *sound spheres*), which are organized in two rows, each comprising the 12 notes of an octave (figure 1). The two rows correspond to two different octaves, one octave apart. To differentiate the natural notes from sharp notes, sound spheres of different sizes are used. This type of visual differentiation is used in many traditional musical instruments, loosely echoing for example, the layout of piano keys or glockenspiel bars.

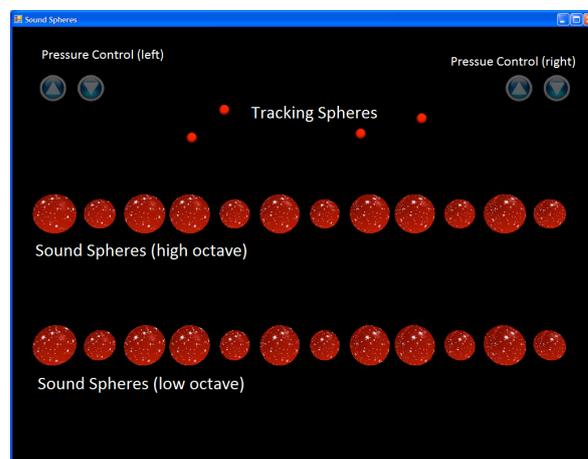


Figure 1. Sound Spheres User Interface



Figure 2. Playing the Sound Spheres

5. DESIGN OF SOUND SPHERES

To support the design of the Sound Spheres VMI we used the articulatory design heuristics outlined above to guide a rapid prototyping approach. Some limited pilot testing was carried out with users during parts of this process (see section 5.4). However, the design heuristics were used to guide design decisions when user testing was impractical, in ways discussed below.

Given the starting point – fingers in free air directing the collision of spheres to produce sounds – there are, broadly speaking, three principal categories of design decision to be made, which are summarized in table 1. The first is the design of specific gestures, or aspects of gesture, for each of the four *instrument control parameters* identified by articulatory, i.e. position, angle speed and pressure. In practical terms, this decision particularly concerns how the values of the various control parameters are to be derived from the finger tracking data. The second category of design decision is to map each control parameter to an appropriate sound shaping operation. The third is to design visual feedback as needed.

Generally, design decisions for the first two control parameters, position (fig. 3) and speed (fig. 4) were relatively non-problematic, whereas decisions for the angle and pressure control parameters were more challenging, especially pressure, in the absence of tactile feedback.

In the remainder of this section, we outline the principal design decisions associated with each of the four control parameters in turn (sections 5.1 - 5.4) and then consider visual feedback for the VMI as a whole (section 5.5).

| <i>Instrument control parameters</i> | <i>Effect on sound</i> | <i>Visual feedback</i> |
|--|--|---|
| <i>Position</i> Position of a tracking sphere at point of collision (figure 3). | <i>Stereo Panning</i> The sound is increasingly panned to the left or right speaker dependent on the position of collision. | <i>Flying Sparks</i> The direction of sparks is dependent of the position of tracking sphere collision (figure 6). |
| <i>Speed</i> Speed of a tracking sphere's movement at point of collision (figure 4). | <i>Volume</i> A greater speed results in a higher volume. | <i>Spin</i> The greater the speed of the tracking sphere the faster the sound spheres spin on collision. |
| <i>Pressure</i> Based on momentum of tracking sphere at point of collision. Tracking sphere size is changed to increase or decrease momentum. | <i>Parametric EQ</i> A greater pressure results in a tone where the higher frequencies are boosted. | <i>Size</i> The greater the pressure the larger the tracking sphere. |

| <i>Angle</i> | <i>Chorus</i> | <i>None</i> |
|---|--|-------------|
| Angle generated by a tracking sphere's start position and collision point (figure 5). | An acute angle results in a chorus effect with a greater degree of modulation than a less acute angle. | |

Table 1. Outline of principal design decisions

5.1 Key design decisions for Position

When a tracking sphere collides with a sound sphere the value of the *position* control parameter is taken to be the horizontal distance from the point of collision and the central line of the sound sphere. This position is used to modify the sound generated at the point of collision by stereo panning to the left or right according to the distance from the sound sphere's central line (see figure 3 and table 1).

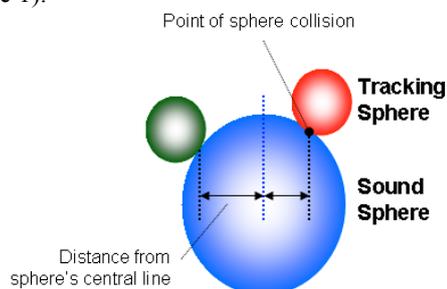


Figure 3. Position articulation

5.2 Key design decisions for Speed

When a tracking sphere collides with a sound sphere the average *speed* of the tracking sphere is taken to be the distance between the start and collision positions divided by the time difference between the start and collision positions, as illustrated in figure 4. The speed is used to adjust the sound generated at the point of collision simply by adjusting the volume, with a greater speed resulting in a higher volume.

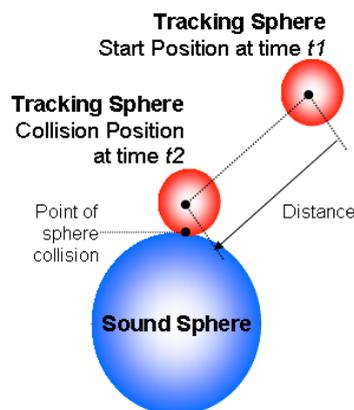


Figure 4. Speed articulation

Prompting design decisions associated with instrument control parameters is helpful, but the articulatory heuristics

also prompt a consideration of the degree to which any design decisions impact on playability, progression, control, predictability, reproducibility, and balance between challenge, frustration and boredom.

In the case of the above-mentioned design decisions for speed, articulation's playability design heuristic prompts the question of how it might be possible for a player to execute low speed gestures when there is a need to strike sound spheres rapidly in succession. However, reflection reminds us that an analogous problem exists in many traditional instruments without playability being impaired. For example, the volume of a xylophone is dependent on the speed on which the player strikes the bars, despite the fact that the mallets may have to be moved quickly to keep time. Playability is not thereby destroyed. Of course, playability depends on skill, but the present design appears to offer a broadly welcome design trade-off between playability, progression and challenge.

Continuing the prompted reflection on playability and challenge, an analogy with piano fingering suggests that the Sound Spheres player has a choice of playing a forthcoming note with any of the four *tracking spheres* and hence finger distance could be minimized with practice. Finally, a small movement of the fingers can affect a big movement in the tracking spheres (sensitivity) allowing individual adjustment of the "action" of sound spheres to assist playability.

5.3 Key design decisions for Angle

Compared with the design decisions associated with position and speed, the design decisions for *angle* are necessarily a little more oblique. The key facilitating step turned out to be to consider the *starting* position for a finger trajectory, as well as the collision point.

Thus, when a tracking sphere collides with a sound sphere the *angle* is taken to be the acute angle between three points, as illustrated in figure 5: point 1 is the center of the tracking sphere at the start of its movement towards the sound sphere, point 2 is the center of the tracking sphere at the point of collision with the sound sphere, and point 3 is any point horizontally displaced from the point of collision. The sound generated at the point of collision is adjusted dependent on the acute angle between these points. The echo, distortion and chorus effects provided by Microsoft's DirectSound were tried and the latter was judged the most suitable for the collision sound. In particular, a more acute angle results in a chorus effect with a greater degree of modulation than a less acute angle.

Thus, the collision of a tracking sphere with a sound sphere at an identical position can sound different depending on the starting position of the tracking sphere. This enables musical expression by swiping fingers in different ways. The articulation heuristics again direct us to consider the degree to which this design decision may impact on such factors as playability, progression, control, predictability, reproducibility, and the balance between challenge, frustration and boredom.

This leads to a reflection on analogous situations, such as when a drummer strikes a cymbal. A change in the angle at which a drummer strikes a cymbal will produce a

different sound. Sometimes a player will use a shallow angle and appear to brush the drumstick over the surface of the cymbal, and sometimes a more direct hit is executed, with widely different sounds being generated. Returning to the design decision in Sound Spheres, little can be concluded about playability, but these considerations do suggest challenge and possible progression.

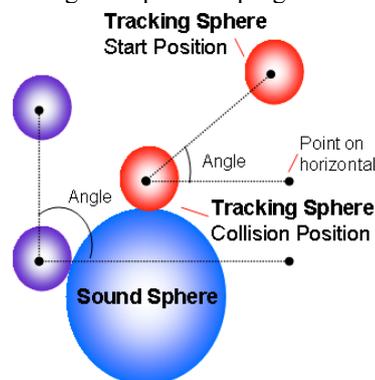


Figure 5. Articulation of Angle: two different collisions are illustrated, each with their own angle.

The method outlined above for determining the angle of a collision assumes that the starting position of each finger-driven tracking-sphere trajectory is well defined. In practice, the transition of a tracking sphere from playing one note to the next will frequently involve continuous motion, and hence the point at which a movement corresponds to the start of playing a new note can be difficult to ascertain. The engineering decision as to how the starting point is to be identified will have implications for articulation factors such as playability, so reflection on playability is prompted. When playing a traditional percussive instrument such as a xylophone or steel drums, or even a stringed instrument like the piano, the movement of the striking object (be it a mallet, stick or fingers) from one note to the next is rarely linear. A player generally lifts the object from one striking position before they start the movement to make another strike. With this in mind, the decision was taken for the tracking sphere's starting position to be determined by the point at which the movement changes from a positive direction in the y-plane to a negative one, i.e. the point at which a downward movement begins, after an upward movement. Sound is also generated if a tracking sphere hits a sound sphere from below (i.e. with an upward movement not followed by a downward movement), but the angle and speed controls are not applied in that case, in order to nudge players towards the xylophone-like playing of Sound Spheres to reinforce the articulation of playability and predictability, while not restricting the free movement afforded by a non-contact VMI.

5.4 Key design decisions for Pressure

In a non-contact environment, finding an appropriate gesture, or aspect of gesture, to map onto a *pressure* control parameter presents a design challenge.

To help guide design, pressure was deemed to be closely related conceptually to momentum. Momentum is defined as the product of an object's mass and its velocity.

Consider two objects with different masses travelling at the same velocity, and consequently different momenta. If they were both to collide against the same surface then the one with the larger mass would exert more pressure. If we assume the virtual mass of tracking spheres to be proportional to their size, we can conclude that a larger tracking sphere would exert a greater pressure on a sound sphere than a smaller tracking sphere travelling at the same velocity. In other words, by varying the size of a tracking sphere we can vary the pressure being applied to a sound sphere during collision.

To implement the ability to dynamically and rapidly change the size of the tracking spheres, the user interface displays a visual component called a *pressure control* (Figure 1). A pressure control has been placed on either side of the user interface so that it can be quickly accessed by tracking spheres controlled by either the player's right or left hand. The pressure control has two circular surfaces, one containing an upwards facing arrow representing increasing pressure and one a downward facing arrow representing decreasing pressure. This control will gradually increase or decrease the size (and hence the implied pressure) of *all* tracking spheres when the center point of *one* of the tracking spheres is positioned over one of the pressure control's surfaces.

The design of the pressure control was motivated by the need to provide an interface that is intuitive to non-musicians while providing the degree of control expected in music technology. As such, while the upward and downward arrows are familiar from home electronics (e.g. to modify sound volume in discrete steps), they provide the same continuous control as e.g. modulation wheels. Without any additional movement, just by hovering a tracking sphere over an arrow, the size of all tracking spheres is changed in a continuous way.

Reflecting once more on playability, progression and challenge, it is clear that by using one hand to vary pressure while the other hand triggers sounds, it should be possible to change pressure relatively rapidly.

Pressure is used to modify the sound generated at the point of collision in the following way: a greater pressure results in a tone where the higher frequencies are boosted using parametric EQ.

5.5 Visual feedback

The articulatory heuristics encourage the use of visual feedback to assist with the communication of position angle and speed, moderated by considerations such as playability, progression, control, predictability, reproducibility, challenge, frustration and boredom. The Sound Spheres VMI provides visual feedback to the player when the tracking spheres collide with sound spheres in several ways, as follows.

Firstly, graphics are displayed at the point of each collision (figure 6). A graphics particle engine was implemented to display a set of flying sparks at the point of collision. The direction and dispersal of the sparks is dependent on the *position* of the collision in the sense defined in section 5.1. This is illustrated in Figure 6.

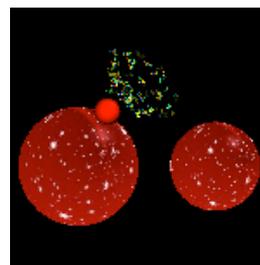


Figure 6. Sphere Collision Sparks

Secondly, when a tracking sphere collides with a sound sphere, the sound sphere vibrates as if it were on a spring. The vibration diminishes over time and then stops. The direction of the vibration is always up and down. Consideration was given as to whether the direction of vibration should also be dependent on *angle*, however reflection on articulatory issues prompted this idea to be dropped. As the sound spheres are placed close together, any sideways vibration could result in their collision, with likely negative consequences for playability, control and frustration. Hence, the vibration of the tracking spheres is not related to any specific control parameter and indicates sphere collision only.

Thirdly, when a tracking sphere collides with a sound sphere, the sound sphere spins around its horizontal axis. The initial speed of spin is dependent on the *speed* of the colliding tracking sphere, and the speed of rotation diminishes over time until the spinning stops. In order to ensure that the speed of spin is readily apparent to the user, we use spheres instead of circles, in an otherwise 2D layout (Figure 1), and then map graphical textures onto the sound spheres.

Visual feedback for *pressure* has already been described in the previous section.

To sum up, there are three elements of visual feedback for the collision of spheres (sparks, spin, vibration) in order to provide a better sense of collision and better compensate for the lack of tactile feedback.

6. EVALUATION

Heuristic evaluation is often used in HCI when user testing is impractical. However, it can also be used to help structure tests with users. The latter approach was used in the evaluation of the Sound Spheres VMI. In the formative user testing, eight participants took part in individual sessions to play the Sound Spheres. Five of the participants were musicians. Participants without prior music knowledge or instrument playing experience were included to check whether they were disadvantaged in using finger tracking for playing music. Three participants (including one musician and two non-musicians) had previously participated in design prototyping. The sessions were split into a number of stages that required the participants to try out different elements of the instrument (Figure 7).

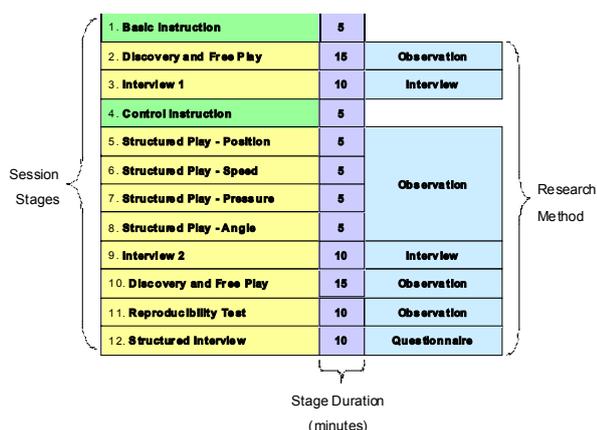


Figure 7. User study stages

Parts of the evaluation were suggested by the articulatory heuristics, while others were intended to explore wider issues. Both qualitative and quantitative data were collected. Observation notes were taken during the user study sessions to provide data for a comparative study to determine patterns of use and behaviour (feelings), body movement and posture, ease of use of the interface, ability to understand and use the control parameters, progression of learning, likes and dislikes, etc. Video recordings were also taken to support, validate and clarify observation notes. Interviews were conducted with each participant after each stage, and the responses were also used for a comparative study.

At the end of each user study session the participant was asked to complete a questionnaire with 49 questions. The initial 5 questions served to identify the participant and their ability to play and read music. Two questions asked the participant to rank the control parameters in terms of ease of use and importance to musical outcomes. The last 3 questions asked for general comments about what participants liked most and least about Sound Spheres. The remaining 39 questions covered the various design factors (playability, progression, control, predictability, reproducibility, and balance between challenge, frustration and boredom), asking participants to respond using a 5-point Likert rating scale (strongly disagree, disagree, neither agree or disagree, agree, and strongly agree) thus providing quantitative data to which statistical analysis could be applied.

Spearman's rank correlation method was used to determine the relationship between 57 pairs of questionnaire responses, e.g. if the ease of use of the speed control parameter correlated with the preference for its applied visual feedback. Furthermore, due to the small sample size, the non-parametric Mann-Whitney U Test was systematically applied to each of the 41 questions to test the hypotheses that questions may be answered differently between musicians and non-musicians, and between those who did and did not participate in the prototype reviews.

7. RESULTS

Statistical analysis of the questionnaire responses showed strongly positive feedback to many factors relating to the Sound Spheres VMI. For example, 87.5% of participants thought that the Sound Spheres VMI facilitated the crea-

tion of music well and that their playing improved over time. 75% of participants thought that it was easy to move the tracking spheres using the finger tracking method. Responses to questions about factors such as general playability, the progression of the musician's ability, control, and balance between challenge, frustration and boredom suggested that the Sound Spheres VMI was generally judged positively in these respects. Responses to questions on the factors of predictability and reproducibility generally showed negative judgments in these areas. In fact, as observed during the reproducibility test, all participants were able to repeatedly play a simple tune but only two of them performed it with good timing. However, observation and the results of the Mann-Whitney U Tests suggested less negative judgments where more playing time (i.e. practice) was given to the participants, which indicates that Sound Spheres allows progression towards more accurate reproduction.

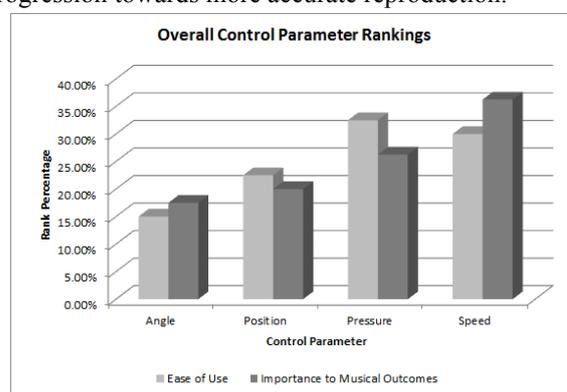


Figure 8. Control parameter rankings

The control parameters of *pressure*, *speed*, *angle* and *position* were ranked from 1 to 4 based upon their ease of control (1 being the easiest and 4 being the hardest) and also for their importance to musical outcomes, i.e. which control could be used best for affecting the musical outcome (1 being the most important and 4 being the least). A scoring system was applied to the rankings received by each of the participants (4 points were given to a rank of 1, 3 to a rank of 2, etc.) and a ranked scoring was calculated for each control parameter. The percentage of the sum of all the control parameters scorings was calculated for each. These percentages are shown in Figure 8. In general the control of *pressure*, *speed*, and *position* was considered easy, and the sounds generated for each of these controls were considered apparent, consistent and appropriate. *Angle* was the control parameter that received the most negative feedback in terms of its ease of control and associated audio result.

There appear to be several reasons for this, which we will briefly review. Firstly, the positioning of the sharp note sound spheres (which were placed lower than the natural notes) made them difficult to hit at an angle. Secondly, participants found that they often played more than one intended note when using the *angle* control due to the close proximity of sound spheres. Thirdly, visual feedback was not implemented for the *angle* control parameter. This suggests the combination of both audio and visual feedback (synchresis) may play an important role in non-contact VMIs.

Only 8 of the 57 Spearman's rank correlation results showed statistical significance and through further analysis 5 of these results were considered unreliable. For example, one negative correlation coefficient value suggests that the Sound Spheres VMI facilitates the creation of music better as the control of the tracking spheres gets harder. This is the reverse of what would be expected, especially considering that 87.5% of participants thought that the Sound Spheres VMI facilitated the creation of music well and 75% thought that the movement of the tracking spheres was easy. However a strong correlation exists between the improvement of ability to play the Sound Spheres VMI over time and the ability to distinguish the application of more than one control parameter at a time. This suggests that progression of ability or skill in playing the Sound Spheres VMI can be achieved. Correlation also suggests that accuracy in positioning the tracking spheres increases as the consistency in control of tracking sphere movement increases.

The Mann-Whitney U Test results indicated that there was no significant difference between musicians and non-musicians in the way questions were answered. However, there were five questions that identified significant (i.e. $p < 0.05$) differences between the responses of those who participated in the prototype review sessions and first time users of the Sound Spheres VMI. These results indicate that participants of the prototype review sessions were more able to consistently control the movement and position of the tracking spheres. They also used the control parameters to add expression during play more than first time participants. Participants of the prototype review sessions more strongly agreed with the change in sound being *apparent* and *consistent* when using the pressure control.

8. IMPLEMENTATION

The Sound Spheres software was developed using Microsoft's Visual Basic programming language and DirectX graphics libraries. The .NET managed library WiimoteLib [10] is used for handling and interpreting Wiimote data. The VMI's components are:

- The Sound Spheres software.
- Laptop computer and 24-bit sound card, external speakers and wide-screen monitor.
- Bluetooth adapter and supporting driver.
- Wiimote controller.
- Infrared LED array with cover.
- Four reflective markers.

The components are setup on a two-tiered desk with the top tier used as a surface on which to stand the speakers and computer monitor and the lower tier used as a surface for placement of the Wiimote and LED arrays. Separate tiers enable the Wiimote and LED arrays to be positioned horizontally central to the monitor and speakers without obstructing the player's view of the monitor. The Wiimote and LED array can be adjusted up or down to suit desired playing positions. An adjustable chair also allows players to raise or lower their playing position. The reference speakers are positioned either side of the monitor so that stereo effects are maximized. The system's setup can be seen in Figures 9 and 10.



Figure 9. Sound Spheres Implementation



Figure 10. LED Array, Wiimote and Cover

9. LIMITATIONS

There are many other criteria which could be used to support VMI design. Articulatory simply offers one example of design and evaluation heuristics.

The evaluation used subjective measures for the various articulatory characteristics. However, this reflects accepted approaches to heuristic evaluation.

We have reported on a single case study. Further work would be required to more deeply understand the possible value of articulatory, or related heuristics.

Articulatory might be argued to have a potentially 'reactionary' influence on VMIs. It may perhaps focus attention on relatively well-explored conceptual metaphors [11] such as position and speed. However, where other metaphors might be more appropriate, the present study offers a good starting point for constructing alternative sets of heuristics with contrasting characteristics.

10. RELATED WORK

The HCI literature on design and evaluation in Music Interaction is relatively sparse, though now growing. Wanderley and Orio [15] carried out a systematic review of existing mainstream HCI techniques for evaluating input devices and considered how these might be applied to music interaction. They also explored the notion of benchmarks (i.e. common musical tasks) that might potentially form part of a task-centric evaluation methodology. Kiefer et al. [16] reviewed some newer developments, and reported on a case study which stressed the value of interview data for identifying unexpected usability issues. Seago [17] critiqued existing user interfaces for timbre design from an HCI point of view. Wilkie et al. [11] introduced a novel approach to evaluating user

interfaces for music, based on embodied cognition, image schemas and conceptual metaphor. Holland et al. [18] outlined how this approach might be applied to whole body and other non-contact interfaces for music. Modh-rain [19] offers a valuable reflection on the role of evaluation in Digital instrument design. Interestingly, phenomenological approaches appear little used in this area; the Second Person technique [20] seems particularly suited to exploring the experience of musicians playing such instruments, to assist designers and evaluators.

11. CONCLUSIONS

Design for non-contact VMIs is challenging. By borrowing and adapting VMI design criteria from the literature we have assembled a simple set of design and evaluation heuristics dubbed *articulacy* for supporting VMI design. We have presented a case study demonstrating how articulacy has been used to design and formatively evaluate a novel non-contact VMI called Sound Spheres. Articulacy has been shown to help structure or guide varied design decisions, including aspects of: the design and refinement of various finger tracking measures for controlling instrument control parameters; the mapping of control parameters to sound shaping operations; and the design of visual feedback (which appears to be particularly important in non-contact VMIs). In some cases the heuristics directed the search for non obvious features of a design, e.g. prompting a controlling role for pressure in the absence of contact, and motivating various kinds of visual feedback. In other cases the heuristics motivated reflection on possible design decisions using various design criteria, e.g. considering how a mapping for speed might affect playability and progression.

After the design phase, articulacy has also been demonstrated to be useful in helping to structure the formative *evaluation* of a non-contact VMI. For example, the participants' answers to interviews and the questionnaire were designed to cast light on how well the factors of playability, progression, control, and balance between challenge, frustration and boredom were achieved in the context of various control parameters. Not all designers find design and evaluation heuristics useful, but some do. In the present case study we have demonstrated some of the ways in which a heuristic design approach might support some VMI designers in gaining experience as a step to acquiring more intuitive mastery.

12. REFERENCES

- [1] Jordà, S. "Digital Instruments and Players: Part 1 – Efficiency and Apprenticeship". *Proc. Int'l Conf. on New Interfaces for Musical Expression*, 2004.
- [2] Jordà, S. "Digital Instruments and Players: Part II – Diversity, Freedom and Control". *Proc. Int'l Computer Music Conf.*, 2004.
- [3] Lee, J. "Hacking the Nintendo Wii Remote", *IEEE Pervasive Computing*, 7(3):39–45, 2010.
- [4] Mulder, A. "Virtual Musical Instruments: Accessing the sound synthesis universe as a performer", *Proc. 1st Brazilian Symposium on Computer Music*, pp 243-250, 1994.
- [5] Paine, G., Stevenson, I., Pearce, A. "The Thummer Mapping Project (ThuMP)". *Proc. Int'l Conf. on New Interfaces for Musical Expression*, 2007.
- [6] Paine, G., Drummond, J. "TIEM - Taxonomy for real-time Interfaces for Electronic Music performance". MARCS Auditory Laboratories at the University of Western Sydney, 2008. <http://vipre.uws.edu.au/tiem>
- [7] Paine, G. "Towards unified design guidelines for new interfaces for musical expression". *Organised Sound*, 14(2):143-156, 2009.
- [8] Vlaming, L. "Human Interfaces – Finger Tracking Applications", Department of Computer Science, University of Groningen, 2008.
- [9] Wanderley, M. "Gestural Control of Music". IRCAM, Paris, France. 2000.
- [10] Peek, B. "Managed Library for Nintendo's Wiimote". <http://wiimotelib.codeplex.com/>
- [11] Wilkie, K., Holland, S. and Mulholland, P. "What Can the Language of Musicians Tell Us about Music Interaction Design?" *Computer Music Journal*, 34(4), 2010.
- [12] Hornecker, E. "The role of physicality in tangible and embodied interactions", *interactions* 18(2):19-23, March 2011.
- [13] Preece, J., Rogers, Y., Benyon, D., Holland, S., Carey, T. *Human-Computer Interaction*. Addison Wesley, 1994.
- [14] Molich, R., and Nielsen, J. "Improving a human-computer dialogue", *Communications of the ACM* 33(3): 338-348, March 1990.
- [15] Wanderley, M. and Orio, N. "Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI". *Computer Music Journal* 26(3):62–76, Fall 2002.
- [16] Kiefer, C., Collins, N. and Fitzpatrick, G. "HCI Methodology For Evaluating Musical Controllers: A Case Study". *Proc. Int'l Conf. on New Interfaces for Musical Expression*, 2008.
- [17] Seago, A. *A new user interface for musical timbre design*. PhD Thesis, The Open University, 2009.
- [18] Holland, S., Wilkie, K., Bouwer, A., Dalglish, M. and Mulholland, P. "Whole Body Interaction in Abstract Domains", in England, D. (ed.), *Whole Body Interaction*, Springer, 2011.
- [19] O'Modh-rain, S. "A Framework for the Evaluation of Digital Musical Instruments". *Computer Music Journal* 35(1):28–42, Spring 2011.
- [20] Doan, T.B. "Using second person interview techniques". E-Sense Project, The Open University, 2009.

PRIORITIZED CONTIG COMBINING TO SEGREGATE VOICES IN POLYPHONIC MUSIC

Asako Ishigaki, Masaki Matsubara, Hiroaki Saito

Graduate School of Science and Technology

Keio University

asako@nak.ics.keio.ac.jp

ABSTRACT

Polyphonic music is comprised of independent voices sounding synchronously. The task of voice segregation is to assign notes from symbolic representation of a musical score to monophonic voices. Human auditory sense can distinguish these voices. Hence, many previous works utilize perceptual principles. Voice segregation can be applied to music information retrieval and automatic music transcription of polyphonic music. In this paper, we propose to modify the voice segregation algorithm of contig mapping approach by Chew and Wu. This approach consists of 3 steps; segmentation, separation, and combining. We present a modification of “combining” step on the assumption that the accuracy of voice segregation depends on whether the segregation manages to correctly identify which voice is resting. Our algorithm prioritizes voice combining at segmentation boundaries with increasing voice counts. We tested our voice segregation algorithm on 78 pieces of polyphonic music by J.S.Bach. The results show that our algorithm attained 92.21% of average voice consistency.

1. INTRODUCTION

Human auditory sense can distinguish plural melodic lines out of music. In musicology, these melodic lines are called voices. The task of voice segregation is to assign notes from symbolic representation of a musical score, such as MIDI, to voices. Unlike audio source separation, voice segregation cannot seek such clues as sound source or timbre, but relies solely upon pitch height, onset time, and duration and employs perceptual principles such as pitch proximity and stream crossing, indicated by Huron [1].

Voice segregation can be applied to music information retrieval and automatic music transcription. In queries by humming and theme finding, for instance, preliminary voice segregation of polyphonic music and homophonic music should be able to facilitate pattern recognition and extraction of monophonic queries and improve hit rates. Furthermore, in automatic music transcription, since the result of multiple pitch estimation by acoustic signal process-

ing is segregated, voice segregation generates easy-to-read scores.

In this paper, we propose a voice segregation algorithm in polyphonic music based on modification of contig mapping approach by Chew and Wu [2]. Research by Chew and Wu of contig mapping approach provided high accuracy of voice segregation in polyphonic music. This approach consists of 3 steps; first, the piece of music is split between rests into units called contig, then voices are separated within each contig, and finally, the contigs are combined. We recognized the possibility of improving the contig combining step of this approach. Pitch proximity determines the voices to be connected across adjacent contigs in the contig combining step. We noticed that accuracy of connection may vary, depending on whether combining is done prior to a rest or after a rest. In this study, therefore, we propose an algorithm to determine the combining priority.

This paper is organized as follows. Section 2 presents a number of recent voice segregation algorithms. Section 3 describes the contig mapping approach and our modification. Section 4 presents our evaluation methods and experimental results. In Section 5, we discuss the results of experiment. Finally, Section 6 concludes this paper and presents future work.

2. RELATED WORK

In recent years, a number of approaches have been proposed for voice segregation [2, 3, 4, 5, 6, 7, 8].

Some of them employ machine learning. “VoiSe” system by Kirilin and Utgoff [3], for instance, learns about features of pairs of notes in the music and checks whether they belong to the same voice, and the learned decision tree is utilized for voice segregation. Some other algorithms which do not rely upon machine learning utilize perceptual principles by Huron [1] or the preference rules by Temperley [9]. Those approaches that group together a number of notes tend to have higher accuracy. Madsen and Widmer [4] group together notes with the same onset time and duration, establish the cost based upon Temperley’s well-formedness rules, and then arrive at the shortest path by branch and bound search.

In contig mapping approach advocated by Chew and Wu [2], music piece is split into units known as contigs before and after rests. When a certain voice part is resting, fewer voices are sounding during that period of time. The number of voices vary before and after a rest, but since the

music piece is split into contigs at that timing, the number of voices remains constant within each contig.

Then voices are separated within each contig. According to stream crossing principle by Huron [1], "humans have great difficulty of tracking auditory streams that cross with respect to pitch." Therefore Chew and Wu assumed that voices would not cross. Since there are constant voice counts within a contig, voices can be separated simply by numbering in the order of pitch.

Finally, contigs are combined. Then it would be necessary to determine which voices are to be interconnected across adjacent contigs. Chew and Wu rely upon pitch proximity to determine which voices are to be connected. When all voices sound synchronously within a contig, such a contig is called maximal voice contig. Those contigs that are adjacent to a maximal voice contig are combined to that maximal voice contig. This combining process shall be performed to all of the maximal voice contigs of the entire music piece. Voice segregation of the music is complete when all the contigs are connected to the voices within the maximal voice contigs. Chew and Wu proposed metrics to measure the correctness of voice separation algorithm and achieved high hit ratio of polyphonic music data.

Voice segregation is applied to polyphonic as well as homophonic music, and the term 'voice' has different meanings between them, as described by Cambouropoulos [6]. Polyphonic music is comprised of independent voices sounding synchronously, and each voice is monophonic. In polyphonic music, voice structure noted by the composer on the score is defined as the ground truth. The proportion of the notes assigned to the correct voice facilitates quantitative evaluation of the voice segregation. In this paper, we aim to segregate polyphonic music into monophonic voices, matching up precisely with the voice structure noted in the score.

In music piece with many voice parts, voice segregation of polyphonic music is prone to errors because music pieces with many voices tend to have also many rests. For example, when 3 voices out of 5 are resting, it would be difficult to infer which 2 voices out of 5 are continuing. This affects the combining step of the contig mapping approach. We propose to enhance the hit rate by improving this step.

3. THE ALGORITHM

3.1 Contig Mapping Approach

We assume that the accuracy of voice segregation depends on whether the segregation manages to correctly identify



Figure 1. Example of a 3-voice polyphonic score. Measure 8 of Bach's Sinfonia No.9 (BWV795)

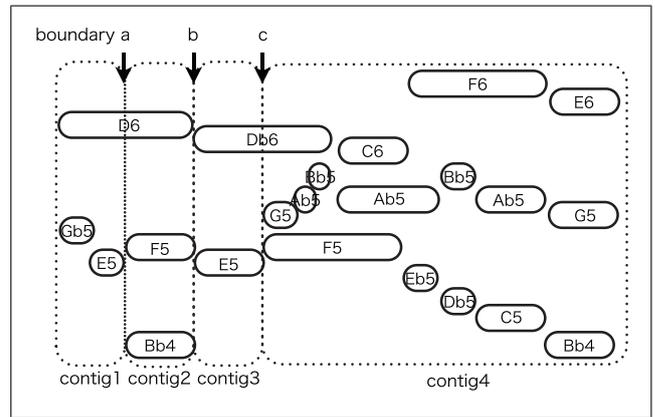


Figure 2. Example of segmentation into contigs

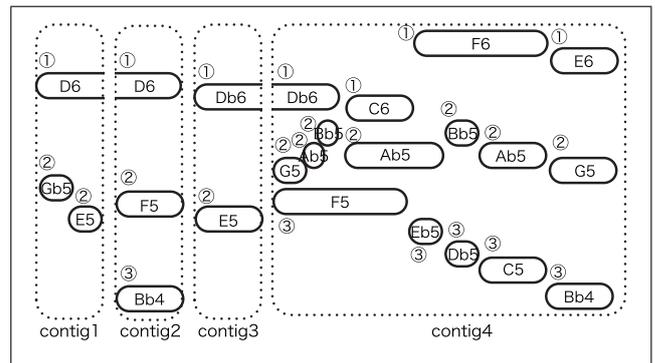


Figure 3. Example of voice segregation within each contig

which voice is resting.

Figure 1 illustrates an example of a 3-voice music. The upward stem notes of the upper staff correspond to the first voice, downward stem notes of the upper staff correspond to the second voice, and the downward stem notes of the lower staff correspond to the third voice. There are 2 eighth rests, one at the first beat, the other at the second. It would be necessary to correlate those rests with the corresponding voices. In contig mapping approach by Chew and Wu, the segregation process is divided into 3 steps to address this issue. The followings describe each step with its underlying concept.

3.1.1 Segmentation

First, music piece is split into units called contigs before and after rests. Figure 1 describes an example of a 3-voice score. This music is split into contigs before and after the eighth rests at the first and the second beat. Figure 2 illustrates how the segmentation is done. Contig 1 has one voice resting and 2 voices sounding, and contig 3 does the same. The number of voices changes at the boundary between contigs, but remains unchanged within a contig. If a note extends across a contig boundary, such as the quarter note on the second beat, a clone for the note is placed in the subsequent contig.

3.1.2 separation

According to the perceptual principles review by Huron, “humans have great difficulty of tracking auditory streams that cross with respect to pitch”, which means that streams would hardly cross in polyphonic music, as it is intended by one musical instrument.

On the assumption that voices would not cross within a contig with constant voice counts, notes would be easily assignable to voices following the pitch order. Figure 3 illustrates the numbering of notes within each contig for the example given in Figure 2. Experiments by Chew and Wu demonstrated 99.8 percent accuracy of voice separation within a contig, but this applies only to allocation of notes to streams within a contig. For example, if a contig in a 3-voice music has 2 voices, this phase of separation does not infer as to which 2 out of the 3 voices they corresponded to. Such an inference is performed in the following combining step.

3.1.3 Combining

The final step combines the whole music piece which has been segmented into contigs. In this step, connection of voices across two adjacent contigs is inferred. In Figure 3, for instance, 2 voices in contig 3 have provisionally assigned numbers ① and ② respectively, but in reality, the first and the third voices are sounding. In the combining step, therefore, these two voices should be inferred to be connected to voices ① and ③ out of the 3 voices of adjacent contigs.

Huron reports, “the coherence of an auditory stream is maintained by close pitch proximity in successive tones within the stream.” It is inferred that across adjacent contigs, voices with closer pitch proximity belong to the same original voice.

The following is the method to connect voices with closer pitch proximity by the definition of the cost of voice connection based on pitch proximity.

First, the number of voices in each of the 2 contigs to be combined are counted. Then list all possible combinations of the voice connection. In the case of a 2-voice contig(P_1, P_2) to be combined to a 3-voice contig ($N_1 \dots N_3$), 3 patterns of connection combinations would be possible as indicated in Table 1, assuming that voices would not cross.

| 3-voice contig | 2-voice contig | | |
|----------------|----------------|-------------|-------------|
| | pattern 1 | pattern 2 | pattern 3 |
| N_1 | \emptyset | P_1 | P_1 |
| N_2 | P_1 | \emptyset | P_2 |
| N_3 | P_2 | P_2 | \emptyset |

Table 1. Combinations of combining a 2-voice contig and a 3-voice contig

In pattern 2, P_1 is connected to N_1 and P_2 to N_3 , and N_2 is on rest. It should be possible to enumerate all the connection combinations each time combining process takes place, as the number of voices is limited. Then arrive at

the sum of connection costs for all of the patterns listed above. The cost shall be defined as the absolute difference between the pitches of the two notes; the final note of the preceding voice and the first note of the subsequent voice. If, however, the first note of the subsequent voice is the clone of the last note of the preceding note, the cost should be substituted by a negative value(-1), to ensure connection between these voices. “0” cost shall apply when either of the voices is on rest. Between contig 1 and contig 2 on Figure 3, since the first note of N_1 is the clone of the last note of P_1 , the cost shall be -1, the cost for combining of N_2 and a rest shall be 0, and the cost between the first note of N_3 and the last note of P_2 shall be 6 because they are 6 semitones apart. That brings the total sum of cost for pattern 2 will be 5, arrived as $-1 + 0 + 6 = 5$. Following the same procedure, the cost for pattern1 shall be 15 and cost for pattern 3 shall be 0.

Then finally, choose the pattern with the minimum cost. In the above-mentioned example, the cost for pattern 3 is 0 and therefore the smallest. Pattern 3 combination is inferred to be optimum to connect contig 1 and contig 2 between closer voices. Thus voices ① and ② are identified as first and second voices, respectively.

However, an issue arises when contig 3 is combined. The above-mentioned optimum pattern to combine contig 2 and contig 3 indicates that voices ① and ② of contig 3 are first and second voices. On the other hand, optimum pattern to combine contig 3 and 4 infers contig 3 voices ① and ② to be first and third voices. In order to overcome this incoherence, we define the priority of combining.

3.2 Proposed Modification

3.2.1 Type of Boundaries between Contigs

At the boundary between contigs, the number of voices changes. There are two kinds of boundaries; one kind is boundary with increasing voice counts, such as boundary a, c on Figure 2 where voices increase from 2 to 3, and the other kind is boundary with decreasing voice counts, such as boundary b where voices decrease from 3 to 2.

Polyphonic music would develop by presenting the theme with each voice. The beginning of a stream is most often accompanied by the theme, so when a new stream begins at a boundary, which means that the voice counts increase with this boundary, the new stream should stand out. Based on the pitch proximity principle, therefore, the very first note of the new stream should have a pitch substantially apart from the notes of the remaining part of the voices that sounded immediately before. It is desirable to focus on boundaries where voices are easily distinguishable, as notes with closer pitch is the most relied-upon clue for voice connection.

We presumed that voice connection would be more accurate when contigs are combined at boundary with increasing voices than at boundary with decreasing voices. We verified our presumption with 78 pieces of music of J.S.Bach, such as invention, sinfonia, and fugue [10, 11, 12]. We made distinction between inter-contig boundaries with increasing voices and decreasing voices for all of the

78 pieces of music. Then we counted the accurate connection of voices across boundaries, following the contig combining procedure as described in 3.1.3. The result is shown in Table 2.

Out of the 2271 boundaries with increasing voice counts, 2004 boundaries achieved correct connection of adjacent contigs; the success ratio is 88.2%. Whereas boundaries with decreasing voice counts recorded 79.1% of connection accuracy. The result indicates voices are combined more accurately across contigs at boundaries where voice number is increasing. We propose an algorithm to prioritize voice combining at boundaries with increasing voice counts to improve the accuracy of voice segregation. The prioritization algorithm is described in the following section.

3.2.2 Connection Algorithm

We propose an algorithm that combines contigs only at those boundaries where the voice counts are increasing and then all the contigs are to be connected to one. The prioritization algorithm is described below.

- (1) Specify contigs $C_0 \dots C_M$, and $|C_M|$ represents the number of voices of C_M
- (2) Combine C_0 and C_1 if $M = 1$, then go to Step (9)
- (3) Go to Step (9) if $M = 0$
- (4) Set $N \leftarrow 0$
- (5) Go to Step (8) if $N = M$
- (6) Combine C_N and C_{N+1} if $|C_N| \leq |C_{N+1}|$
- (7) Set $N \leftarrow N + 1$ then go to Step (5)
- (8) Specify the combined contigs $C_0 \dots C_I$, set $M \leftarrow I - 1$, then go to Step (1)
- (9) All contigs are combined into a contig

The following describes this algorithm taking an example illustrated sequentially in Figure 4, Figure 5, and Figure 6. In Figure 4, dotted lines show 8 boxes corresponding to contigs 1 through 8. It indicates that contigs 3 and 5 have 2 voices each, contigs 1, 4, 6 and 8 have 3 voices each, and contigs 2 and 7 have 4 voices each. The number of voices increase at 4 boundaries out of 7, i.e., boundaries *a*, *c*, *e*, and *f*. In Step (4) through (8), contigs before and after these 4 boundaries are combined following the procedure described in 3.1.3. Then, contigs are combined into 4 broken-lined boxes, contigs $\{1, 2\}$, $\{3, 4\}$, $\{5, 6, 7\}$ and $\{8\}$, which are now called contig 1', 3', 5', and 8'. Contig

| boundary type | increasing | decreasing |
|-----------------------|------------|------------|
| no. of boundaries | 2271 | 1995 |
| successfully combined | 2004 | 1579 |
| success ratio | 88.2% | 79.1% |

Table 2. Success ratio in two boundary types

3' and 8' contain 3 voices each and contig 1' and 5' contain 4 voices each. It is still unknown which 3 voices out of the 4 are contained in contig 3' and 8'. As we repeat combining of these 4 contigs only at the boundaries where the voice counts increase, contig 3' and 5' are combined and we have 3 contigs; contig 1'', 3'' and 8''(indicated by the solid lines in Figure 5).

Now we still have a 3-voice contig, that is contig 8''. Final combining is between contig 1'' and 3'', and then Step (2) is applied to combine contig 3'' and 8'' at a declining boundary as an exceptional measure. As shown in Figure 6, out of the iterations described above, all of the 8 contigs in Figure 4 have all their voices connected as indicated by gray lines, and the combining step is complete.

4. EXPERIMENTS AND RESULTS

4.1 Evaluation Methods

In polyphonic music, voices are recorded onto the score as is intended by the composer. The correspondence between notes and voices are established as the ground truth.

We tested the voice segregation algorithm on polyphonic music dataset. We chose the average voice consistency (AVC) out of the three metrics advocated by Chew and Wu [2] to quantify the algorithm performance. The voice segregation algorithm assigns all the notes in the music to voices. The assigned result is compared to the ground truth, and AVC is obtained as the percentage of the number of notes assigned to the correct voice.

4.2 Experiments

We obtained AVC from MIDI data of 78 music pieces by J.S. Bach [10, 11, 12]; 15 Inventions (2-voice), 15 Sinfonias (3-voice), 48 Fugues (2 to 5 voices). Since J.S. Bach is the representative composer of polyphonic music, a large number of related researches utilize this dataset for their experiments. Each track of the MIDI data contains a voice segregated in accordance with the ground truth. In this study, we iterated combining of contigs as described in 3.1.3. After iteration of combining up to the phase described in Figure 6 is referred to as Full Experiment. The part of experiment up to the phase described in Figure 4, i.e. combining just once before iteration is referred to as Subset. We tested another algorithm which combines contigs only at the boundaries with decreasing voices and iterates the combining, referred to as Reverse, to contrast with

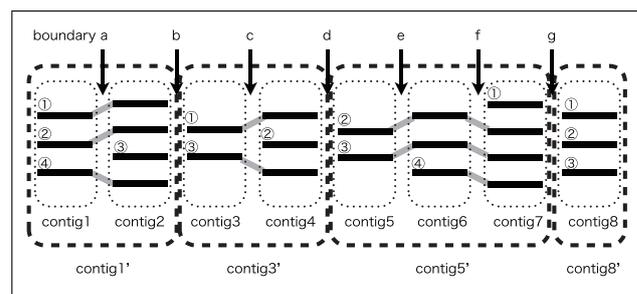


Figure 4. First phase of combining contigs

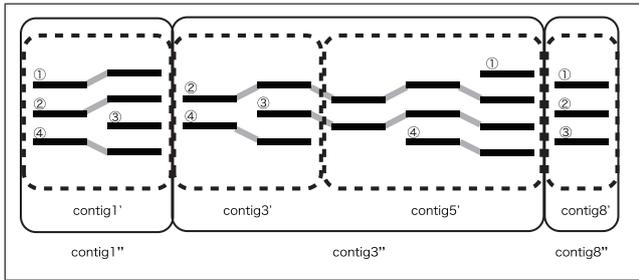


Figure 5. Second phase of combining contigs

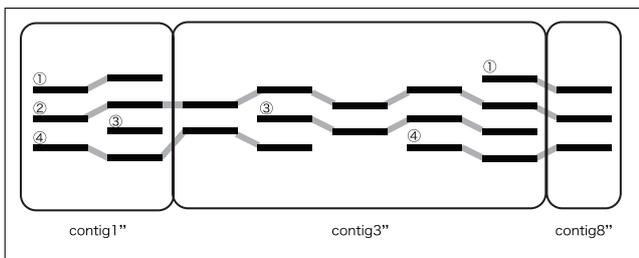


Figure 6. Last phase of combining contigs

Full Experiment.

4.3 Results

Table 3 shows the comparison of experimental results(AVC) of Full Experiment, Subset and Reverse as well as preceding studies based on the dataset of 78 music pieces, which is the same dataset this study is based upon.

| | Invention | Sinfonia | Fugue | Avg. |
|-----------|---------------|---------------|---------------|---------------|
| Full Exp. | 98.73% | 95.27% | 89.21% | 92.21% |
| Subset | 99.02% | 95.30% | 86.43% | 90.56% |
| Reverse | 98.87% | 92.33% | 80.75% | 86.46% |
| Chew | 99.29% | 93.35% | 84.39% | 88.98% |
| Madsen | - | - | - | 70.11% |

Table 3. Experimental results of voice segregation

Fugues vary in voice counts; from 2- up to 5-voice. Table 4 derives from the same dataset as Table 3, but specifically classified by voice counts, which is unique to this particular study.

In the researches to-date, the algorithm by Chew and Wu recorded by far the highest hit rate. The result of our Subset alone surpasses the percentage achieved by Chew and Wu, as well as the result of Reverse fall below them. Furthermore, with the iteration in Full Experiment, the voice segregation accuracy improved even more than Subset. AVC higher than 85% was obtained for relatively more complex and difficult 5-voice music in the dataset.

| | 2-voice | 3-voice | 4-voice | 5-voice |
|-----------|---------------|---------------|---------------|---------------|
| Full Exp. | 98.81% | 93.73% | 84.05% | 85.50% |
| Subset | 99.11% | 93.65% | 79.30% | 67.68% |
| Reverse | 98.86% | 88.51% | 73.37% | 69.50% |

Table 4. Accuracy in 2-5 voices

5. DISCUSSION OF RESULTS

5.1 The result of prioritization of combine process

Table 3 shows that the Subset gave better results than the experiment by Chew and Wu and our Reverse, which indicates that the prioritization of boundaries with increasing voice counts was effective in the “combining” step of contigs.

For example, the result on Figure 7 is the correct voice segregation of the score of Figure 1. On this colored score, the first, the second, and the third voices are represented by pink, sky blue, and orange, respectively. If contigs are combined at the boundaries with decreasing voice counts, it would result in an error as indicated on Figure 8, where the note E on the second beat is represented by sky blue, although it really should be orange. Whereas the algorithm we propose combines contigs at the boundaries with increasing voice counts and generates correct results as indicated on Figure 7 for this example.

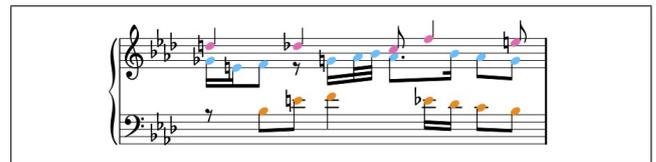


Figure 7. Example of the correctly combined contigs. Measure 8 of Bach's Sinfonia No.9 (BWV795)

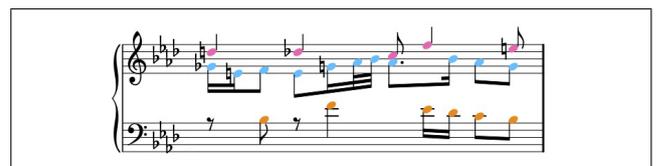


Figure 8. Example of the incorrectly combined contigs. Measure 8 of Bach's Sinfonia No.9 (BWV795)

5.2 The Iteration Effects

Four-voice or 5-voice music tend to have relatively longer duration of rests for 1 or more voices, and it is difficult to presume which of the voices are resting. Furthermore, if there is an error in presuming the voices on rest, that affects many notes and assign them to wrong voices. The success of voice segregation for 4-voice or 5-voice music depends upon the ability to correctly presume which voices out of the 4 or 5 corresponds to the 2 or 3 voices that are sounding.

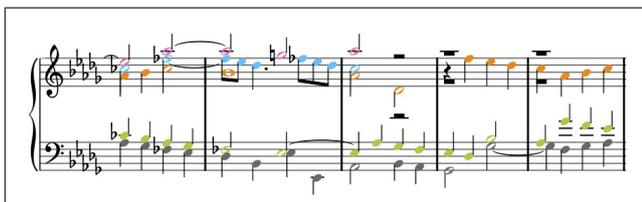


Figure 9. Example of correctly connected voices. Measures 35-39 of Bach's Fugue No.22 (BWV867)

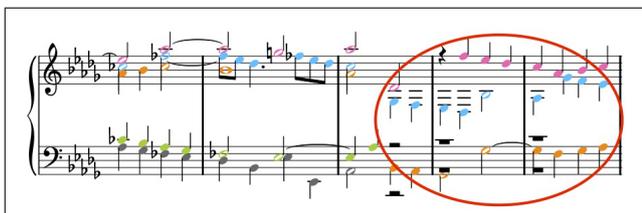


Figure 10. Example of disregarded potentiality of 4th or 5th voices. Measures 35-39 of Bach's Fugue No.22 (BWV867)

Table 3 shows further improvement of success ratio of Full Experiment over Subset. In Full Experiment, combining of contigs was iterated at boundaries with increasing voice counts. Furthermore, the improvements are specifically conspicuous for 4-voice and 5-voice music pieces, as shown in Table 4. It is therefore evident that the iteration by the proposed algorithm assigns sounding notes to the appropriate voices in those contigs with a number of synchronous rests.

Figure 9 illustrates the result of a partially correct voice segregation of a 5-voice music. Color-coding for the first, second and the third voices is the same as in Figure 7, and green and gray represent the fourth and the fifth voices, respectively. As is illustrated in Figure 4, to the extent of Subset, in 2-voice or 3-voice contigs that correspond to the red circle over the score on Figure 10, the potentiality of fourth or fifth voices sounding are disregarded; those 3 voices in reality must be represented by orange, green, and gray colors end up pink, sky blue, and orange. As the experiment advances to Full Experiment, however, the 3 voices are connected to the correct voices out of the five voices, and the result conforms to Figure 9, which is right.

Although this algorithm still does not guarantee perfect connections for all of the cases, our study demonstrated that the ratio of success can be enhanced by prioritizing which contigs are to be combined first.

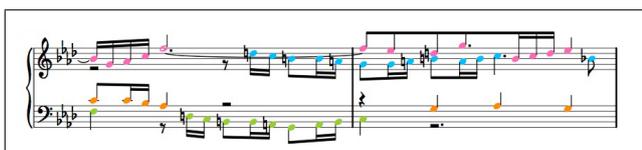


Figure 11. Example of voice permutation. Measures 18 and 19 of Bach's Fugue No.12 (BWV857)



Figure 12. Example of error caused by voice permutation. Measures 18 and 19 of Bach's Fugue No.12 (BWV857)

5.3 Error Analysis

Exceptionally, some music pieces do not end with monophonic voice. Chew and Wu excluded such exceptional contigs from their experiments beforehand; our study did not apply any such exceptions and had some errors. On Table 3, the success ratio of the proposed method for 2-voice was slightly below the experimental result by Chew and Wu. But we consider that this is not a major issue.

There are, however, certain errors that cannot be overcome by contig mapping approach based on pitch proximity principles and voice number variations. Those errors are caused by permutation and crossing of voices.

Figure 11 illustrates the music piece with voice permutation. On the second beat of the second bar, the fourth voice in green has a rest, and at the same time, a new stream begins for the third voice in orange. Contig mapping approach segments contigs by checking the voice number variations. So, if a certain stream ends at the same time with the beginning of another stream, segmentation does not take place. Therefore, the C sound in green and G sound in orange are regarded as one continuing voice. In the red-circled area of Figure 12, the green voice of the first bar is indicated as orange, and the orange voice is indicated as sky blue.

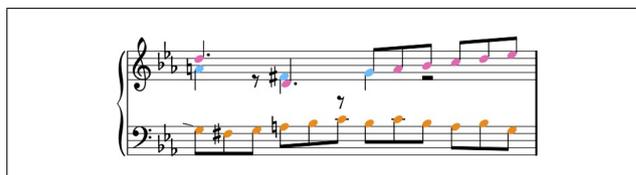


Figure 13. Example of voice crossing. Measure 12 of Bach's Sinfonia No.2 (BWV788)

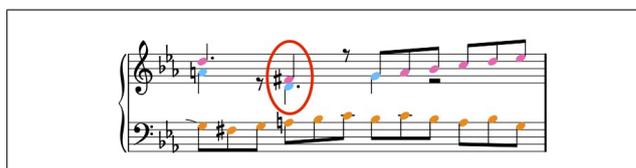


Figure 14. Example of error caused by voice crossing. Measure 12 of Bach's Sinfonia No.2 (BWV788)

Figure 13 illustrates the example of music piece with crossing voices. On the second beat, the D sound in pink has lower pitch than F# sound in sky-blue, and the first and the second voices cross. Many methods proposed to-date including our algorithm rely upon pitch proximity as the clue for voice segregation. The red-circled area on

Figure 14 shows the assignment of F# sound to first voice. This is a difficult problem to overcome in the current circumstances.

As voice crossing and permutations rarely occur, 92% of success ratio on the average was obtained despite of these errors.

6. CONCLUSION AND FUTURE WORK

This study proposes a voice segregation algorithm that has modified the contig mapping approach by Chew and Wu. The “combining” step of contigs has been improved by prioritization of boundaries with increasing voice counts, as higher success ratio was observed than at the boundaries with decreasing voice counts. The proposed algorithm was tested on music data of 78 pieces by J.S. Bach. It was evident from the test results that the success ratio improved and that the prioritization of combining of contigs is effective.

Future work of this study would be to test this algorithm with polyphonic music by other composers as well as other genre of music to broaden its application. We presume that theme finding would be clue for improving voice segregation algorithm because notes within a theme phrase should belong to a same voice. Assembling theme finding and voice segregation may help to overcome voice-crossing and permutation problems.

7. REFERENCES

- [1] D. Huron, “Tone and voice: A derivation of the rules of voice-leading from perceptual principles,” *Music Perception*, vol. 19, no. 1, pp. 1–64, 2001.
- [2] E. Chew and X. Wu, “Separating voices in polyphonic music: A contig mapping approach,” *Computer Music Modeling and Retrieval(CMMR)*, pp. 1–20, 2005.
- [3] P. Kirilin and P. Utgoff, “Voise: Learning to segregate voices in explicit and implicit polyphony,” *International Conference on Music Information Retrieval(ISMIR)*, 2005.
- [4] S. Madsen and G. Widmer, “Separating voices in midi,” *International Conference on Music Information Retrieval(ISMIR)*, 2006.
- [5] A. Jordanous, “Voice separation in polyphonic music: A data-driven approach,” *Proceedings of the International Computer Music Conference(ICMC)*, 2008.
- [6] E. Cambouropoulos, “Voice and stream: Perceptual and computational modeling of voice separation,” *Music Perception*, vol. 26, no. 1, pp. 75–94, 2008.
- [7] I. Karydis, A. Nanopoulos, A. Papadopoulos, and E. Cambouropoulos, “Visa: The voice integration/segregation algorithm,” *Proceedings of the International Conference on Music Information Retrieval(ISMIR)*, 2007.
- [8] J.Kilian and H. Hoos, “Voice separation - a local optimisation approach,” *Proceedings of the Fourth Annual International Symposium on Music Information Retrieval(ISMIR)*, 2003.
- [9] D. Temperley, “The cognition of basic musical structures,” *The MIT Press*, pp. 85–114, 2001.
- [10] J.S.Bach, *Inventions and Sinfonias BWV772-801*. Henle, 1979.
- [11] —, *Well-Tempered Clavier - Part I BWV846-869*. Henle, 2007.
- [12] —, *Well-Tempered Clavier - Part II BWV870-893*. Henle, 2007.

RENCON WORKSHOP 2011 (SMC-RENCON): PERFORMANCE RENDERING CONTEST FOR COMPUTER SYSTEMS

Mitsuyo Hashida
School of Music, Soai University
hashida@soai.ac.jp

Keiji Hirata
Future University Hakodate
hirata@fun.ac.jp

Haruhiro Katayose
Kwansei Gakuin University
katayose@kwansei.ac.jp

ABSTRACT

The Performance Rendering Contest (Rencon) is an annual international competition in which entrants present computer systems they have developed for generating expressive musical performances, which audience members and organizers judge. Recent advances in performance-rendering technology have brought with them the need for a means for researchers in this area to obtain feedback about the abilities of their systems in comparison to those of other researchers.

The Rencon contest at SMC2011 (SMC-Rencon) is going to have two different stages of evaluation. In the first stage, the musicality of generated performances and technical quality of systems will be evaluated by expert reviewers using a blind procedure for evaluation. In the second stage, performances generated on site will be openly evaluated by the SMC audience and Internet viewers. The SMC-Rencon Awards will be bestowed on the systems exhibiting excellent performances at both stages.

1. INTRODUCTION

Performance expression is as important as composition or arrangement. Performance rendering has been one of the main topics since the dawn of music information science[1]. It is an ideal target to test the potential of artificial intelligence. Research that ushered in performance-rendering systems dates back to the 1980s. Since then, a great deal of commercial software for desktop music, digital audio workstations, and voice-singing synthesizers has been published. Performance rendering has also attracted attention due to its importance as an objective in the design of musical content creation.

Generative-music information processing, including performance rendering, is needed to subjectively evaluate generated performances, and this not only involves investigations into the ratio of recognition and reproduction but also sensuousness and emotionality, which are both important in musical performances. Competition is an effective way of obtaining such evaluations and should promote further advances. The rendering contest (Rencon), which was started in 2002, is an annual international competition

in which entrants present computer systems they have developed for generating expressive musical performances, which audience members and organizers judge[2].

Rencon had focused on making an objective guideline of ability and possibility of automatic rendering systems by ranking performances generated by those systems, referring to human performance competitions. Since 2008, the Rencon have held an interactive section, which competes performance rendering by human operators with systems that supports their expression design as a tool.

The competition at SMC2011 will feature a new approach, which involves two different evaluations (<http://www.renconmusic.org/smc2011/>). In the first stage, the musicality of generated performances and technical quality of systems will be evaluated by expert reviewers using a blind procedure of evaluation. In the second stage, performances generated on site will be openly evaluated by the SMC audience and Internet viewers.

The rest of this paper is organized as follows. Section 2 describes the current state of performance rendering and the necessity for two kinds of evaluations. Then, we present an overview of the competition in Section 3. Details on the two different kinds of evaluations are described in Sections 4 and 5. We end with some concluding remarks in Section 6.

2. PERFORMANCE RENDERING SYSTEMS AND EVALUATION

2.1 Performance Rendering by Automated Systems

Research that ushered in performance-rendering systems dates back to the 1980s [3, 4]. Approaches involving music-recognition theories such as the generative theory of tonal music [5], the implication-realization model [6], learning systems [7, 8], and example-based reasoning [9, 10] have been proposed since the 1990s. In addition, a competition for system-rendered performances has been held since 2002 [2]. Moreover, a great deal of commercial software for desktop music and digital audio workstations has been published.

Figure 1 is a diagram of the flow in a typical performance-rendering system. Automated performance-rendering systems generally take a score as the input, generate a performance of this using an original rendering process, and output the rendered performance in MIDI file format. Performance-rendering systems are often categorized into rule-based and case-based schemes. In the rule-based approach, which is used by many commercial music-software sys-

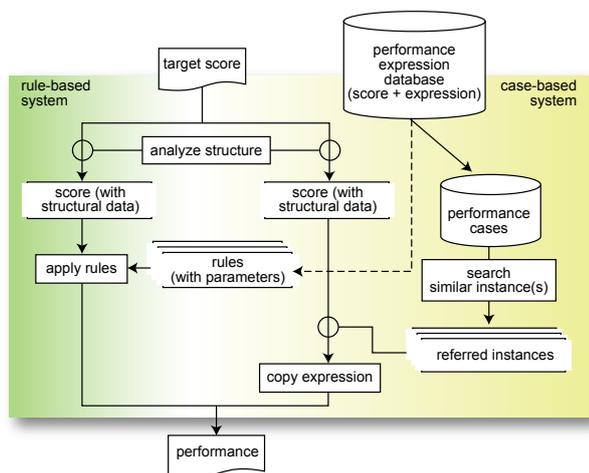


Figure 1. Illustration of typical rendering system

tems, the performance of a score is generated using musical knowledge (rules). In the case-based approach, the system finds a melody (or other sequence) similar to the target melody (or sequence) and directly transfers its expression. This approach enables the user to produce musical expressions even if he/she does not know the rules of expression for the target melody (or sequence).

Some rule-based systems have been applied to extracting rules with parameters from human performances in the same way as case-based systems do[9]. The use of these rules as examples for reference is one trend in performance-rendering studies. Structural information contained in the score has recently been used in both rule-based and case-based systems to emulate the way musicians render musical performances.

2.2 Issues in Evaluation of Musical Performances

It is crucial to introduce subjective evaluation to assess music-related artifacts, including musical performances. This section discusses the designing of a standard to evaluate listening, especially focusing on performance rendering with computers.

2.2.1 Aspects in Evaluation

Of the numerous issues to consider, we address the most crucial aspects in assessment, which are: *musicality*, *adaptability/flexibility*, and *autonomy*.

Musicality: Performances generated with computers should be evaluated in terms of “expressiveness,” as is done in human music contests. From the viewpoint of computer science, it is preferable to give more objective standards to subjective evaluations done by human evaluators. We have two options regarding who evaluates performances, i.e., musical experts or the public who vote on performances.

Adaptability/Flexibility: Judging adaptability in performance rendering with computers is more crucial than that with human performers. For instance, simple memory-based performance-rendering systems can

reproduce fully human-like performances, when they have examples of performances in their database. However, they cannot add any expressions to a score played for the first time. We need to measure adaptability in how well systems can generate expressions of plural pieces and plural genres.

Autonomy: One of the largest concerns from the viewpoint of artificial intelligence is to what degree humans should participate in rendering performance. We should consider views to evaluate autonomy, or contest frameworks, in which the more autonomous a system is, the greater the advantage given to the system.

In addition, *usability* substituting for autonomy should be evaluated when we evaluate performance rendering with interactive sections.

2.2.2 Contest Framework Management

We should implement the aspects described in Section 2.2.1 into the actual contest framework.

We stated, in our introduction to musicality, that there were two major methods of evaluation, i.e., evaluation by musical experts and public voting. In the former, musical experts should execute judgments carefully using sufficiently long time slots. In the latter, the public should vote within a short time and make their evaluations entertaining. The “blind evaluation” that the scientific review process adopts is likely to bore public audiences. The voting procedure by the public is motivated by them watching the process with which performances are generated and the expressiveness of the entrants.

The basic way the adaptability of performance rendering is measured is to let systems generate their own performances of a newly composed piece of music.

Limiting the time for rendering performances and preventing operators from listening to performances that the system generates is an effective framework to measure autonomy. Another effective way of comparing automatic systems fairly and rationally is that the committee generates performances using each of the systems collected from participants. It is a near future work when each system of the entrants works as a fully automatic one.

A different point of view that cannot be ignored is evaluating the extent to which systems can exhibit musicality, even though humans has to tune these up.

3. SMC-RENCON

3.1 Overview

SMC-Rencon is going to have two different evaluation stages (Stage I, II) and two system sections.

3.1.1 Evaluation Stages

In the first evaluation stage (Stage I), the musicality of generated performances and the technical quality of systems are evaluated by expert reviewers using a blind-evaluation procedure. All participants are required to generate expressive performances of a set piece and encouraged to tune

their systems or elaborate on their performances in two days, after the set piece becomes available at the Rencon webpage.

In the second evaluation stage (Stage II), performances generated on site will be openly evaluated by the SMC audience and Internet viewers. The participants are required to generate expressive performances of the set pieces chosen by lot at the venue within a limited time. Adaptability as well as musicality will be evaluated in this section.

3.1.2 System Sections

There will be two system sections in the competition: an **autonomous section** and an **interactive section**. The autonomous section is for autonomous computer systems, such as those using rule-based or case-based approaches to render performances. Entrants will not be allowed to manually edit the performances during the rendering process. The aim of this section is to evaluate performances rendered by autonomous computer systems using rule-based or case-based approaches, for example. The interactive section is for entrants using commercial music software or original applications to render performances. The aim of this section is to build common ground for evaluating human performances accomplished with computer systems as well as to make Rencon more widely appealing.

3.2 Set Pieces and Data Files

All systems are required to render set pieces of music in each stage.

- Stage I: newly composed piano piece (about 1 min)
- Stage II: existing piano piece (about 30 min)

All of the set pieces will be prepared by using Finale 2010. The data files for input to contestants' systems will be provided from software in two formats of MusicXML and a Standard MIDI file. A printed score will also be provided as a reference for human operators. The data in all three formats will be provided to each entrant at the beginning of the competition.

The document type definition (DTD) for MusicXML was developed by Recordare LLC¹. Two files described in versions 1.0 and 2.0 will be provided. The files will be generated by the pre-installed plugin of Finale 2010 (Dolet 5 for Finale). Partwise.dtd is adopted as the top-level format. Note that data and expression marks (e.g., *f*, *p*, *crescendo*, *andante*, *slur* and *staccato*) will be included. Neither phrase structure nor chords are specified. If the piece includes some specific notation (e.g., trills/tremolo and grace notes), the participants should transform these notations to actual notes by themselves.

All note events will be assigned to Channel 1 for the standard MIDI file format (Format 1). The data will be generated by the pre-installed plugin of Finale 2010 (Dolet 5 for Finale). Any information on notation and expression marks will not be included. All velocities will be set to 64. Tempo (bpm) will be set to the value that the set piece indicates. Any control message including the damper pedal

¹<http://www.recordare.com/>

| System name (section) / Author(s) - Institution(s) |
|---|
| usapi (autonomous) [11] Keiko Teramura - Kyoto Univ. |
| Shunji System (autonomous) Shunji Tanaka - Kwansai Gakuin Univ. |
| YQX v0.2 featuring The BasisMixer (autonomous) [12] Sebastian Flossmann, Maarten Grachten, Gerhard Widmer - Johannes Kepler Univ. |
| Kagurame Phase-II (autonomous) Taizan Suzuki, Tatsuya Hino, Shibasaki Masahiro, Yukio Tokunaga - Picolab Co., LTD / Shibaura Inst. of Technology |
| Kagurame Phase-III (autonomous) Taizan Suzuki, Tatsuya Hino, Shibasaki Masahiro, Yukio Tokunaga - Picolab Co., LTD / Shibaura Inst. of Technology |
| DIRECTOR MUSICES (ACCENT-BASED FORMULATION) (interactive) Erica Bisesi, Anders Friberg, Richard Parncutt - Univ. of Graz / KTH |
| VirtualPhilharmony (interactive) [13] Takashi Baba - Kwansai Gakuin Univ. |

Table 1. Candidate Entrants of SMC-Rencon Awards

(Sustain 64) will not be included. The data will not describe the complete music structure, including the phrase structure, harmony, or chord progress.

3.3 Evaluation Process

In Stage I, the musicality of generated performances and the technical quality of the systems will be evaluated by expert reviewers. The final place in Stage I will be calculated from the total of the places in performance and technical quality. In Stage II, performances generated on site will be voted on by the SMC audience and Internet viewers.

3.4 Entrants

Table 1 shows the candidate entrants of SMC-Rencon Awards. Additionally, the following system is to take part in Stage II.

- **CaRo 2.0** (interactive):
Sergio Canazza, Antonio Rodà, Massimiliano Barichello and Davide Ganeo, University of Padova

Submission to Stage II would be open until the deadline of SMC2011 Registration. Each system is briefly introduced at the Rencon webpage.

3.5 Rencon Award

The SMC-Rencon Award will be bestowed on the system with the highest number of musicality rank combining the results of Stages I and Stage II. The Rencon technical award will be bestowed on the participant whose technical point evaluated at the stage I is the highest.

4. STAGE I

4.1 Overview

Stage I was held from March 27–28, 2011 through the Internet. The set piece was a newly composed piano piece “A Little Consolation” by Tadahiro Muraio that lasts about one minute and twenty seconds shown in Figure 2. This stage focuses on the musicality and technical quality of the systems and their performances. All the submission data

A Little Consolation

Tadahiro Murao

Set piece of Rencon Workshop 2011 (SMC-Rencon)
Murao & Rencon (c) 2011

- 2 -

Figure 2. The set piece of Stage I

were sent by email, and the generated performances were attached in SMF format. Participants were given two days to download the set piece to submit their performance.

All of the submitted MIDI data were recorded into mp3 data through playing with a MIDI synthesizer, Acoustic Grand Piano of Yamaha's MOTIF-RACK XS.

4.2 Evaluation Process

In Stage I, the musicality of generated performances and the technical quality of the systems were evaluated by expert reviewers.

Five evaluators were asked to score the musicality of each performance on a scale of 1 to 10, (10: equal to human pianists, 5: mechanical without expression or mediocre, and 1: very poor.) using a method of single-blind evaluation. The places were calculated similarly to the judging system for figure skating (6.0 System[14]). The evaluators were also asked to write comments of 150 words, which were sent back to the contestants. The technical evaluation were executed in the same manner by reviewing the participants' extended abstracts.

The final place in Stage I was calculated from the total of the places in performance and technical quality.

4.3 Results

Table 2 shows the evaluation results of the Stage I.

For the musical evaluation, reviewers graded each of the performances on a scale of 10. And we calculated the place with a modification of prior figure skating scoring system (6.0 system) based on the your evaluation. The result was the same as the result based on the total score.

And also the reviewers stated the five viewpoints with more than 1 sentence to each of them: Level of technical Quality Human(like), Expressiveness Rhythmic accuracy and Musicality. Here we introduce some comments of reviewers:

"This sounds like a well-done computer-generated performance. The rit. in 29-31 is well taken care of; the fermata nicely long. The dynamics in bar 25 also well played." — To No. 7, by Reviewer 4:

"The theme of this melody sounds classical hymn in which tempo should not be fluctuated too much. In this performance, however, the tempo is fluctuated as a romantic piece. ..." — To No. 1, by Reviewer 1:

For the technical quality, each of two reviewers (R3, R5) avoided to review No. 7 and No. 1 because they were co-author or closely related to the system development. To keep five reviewers for each system, six reviewers in all evaluate the systems. All the extended abstracts were hidden the authors' information to the reviewers. Six reviewers of the seven put 1.0 for No. 7. No. 4 and 5 placement were both 5th.

For the final placement, both of the rank of musicality and technical quality were summed up as (a) + (b) in the Table 2. As the result, the first place were No. 7 (YQX[12]), the second was No. 1 (Director Musices), and the third was No. 4 (VirtualPhilharmony[13]) and 6 (Shunji System).

5. STAGE II

5.1 Overview

Stage II will be held at the SMC2011 venue. Participant systems will generate musical performances on site of limited durations. Table 3 lists the time schedule for Stage

| Stage I | place number of musicality | | | | | | place number of technical quality | | | | | | (a)+(b) | final rank | |
|---------------------------|----------------------------|-----|-----|-----|-----|----------|-----------------------------------|-----|-----|-----|-----|-----|---------|------------|----------|
| | R1 | R2 | R3 | R4 | R5 | rank (a) | R1 | R2 | R3 | R4 | R5 | R6 | | | rank (b) |
| No. 1: Director Musices | 2.0 | 4.0 | 4.0 | 3.5 | 2.0 | 3 | 4.0 | 2.0 | 1.0 | 7.0 | - | 3.0 | 2 | 5 | 2 |
| No. 2: usapi | 6.0 | 4.0 | 5.0 | 6.5 | 1.0 | 5 | 7.0 | 3.0 | 6.0 | 2.0 | 5.0 | 6.0 | 7 | 12 | 6 |
| No. 3: Kagurame Phase-II | 5.0 | 6.5 | 6.0 | 5.0 | 7.0 | 6 | 1.0 | 7.0 | 2.0 | 4.0 | 4.0 | 7.0 | 4 | 10 | 5 |
| No. 4: VirtualPhilharmony | 4.0 | 1.0 | 3.0 | 1.0 | 4.0 | 2 | 6.0 | 4.0 | 4.0 | 5.0 | 2.0 | 5.0 | 5 | 7 | 3 |
| No. 5: Kagurame Phase-III | 7.0 | 6.5 | 7.0 | 6.5 | 6.0 | 7 | 3.0 | 6.0 | 5.0 | 3.0 | 6.0 | 4.0 | 5 | 12 | 6 |
| No. 6: Shunji System | 3.0 | 4.0 | 2.0 | 3.5 | 5.0 | 4 | 5.0 | 5.0 | 3.0 | 6.0 | 3.0 | 2.0 | 3 | 7 | 3 |
| No. 7: YQX | 1.0 | 2.0 | 1.0 | 2.0 | 3.0 | 1 | 2.0 | 1.0 | - | 1.0 | 1.0 | 1.0 | 1 | 2 | 1 |

Table 2. Table 2. Evaluation results of SMC-Rencon Stage I (R1-R6 means Reviewer 1, Reviewer 2... 6.). Each number of the table is the place number calculated from the obtained point by each reviewer. The ranks were calculated similarly to the judging system for figure skating (6.0 System[14]).

II.

All of the entered systems will first render expressive performances in the autonomous section, and then the rendered performances will be played by an automated grand piano and evaluated by the audience. The entrants will render performances in the interactive section using commercial or original music applications.

5.2 Autonomous Section

The autonomous section is for performances rendered by autonomous computer systems using, e.g., a rule-based or case-based approach. The systems in this section should be able to

- Read score data in MusicXML or standard MIDI format,
- Render an expressive performance (using, e.g., a rule- or case-based approach), and
- Output the generated data in standard MIDI format.

Figure 3 illustrates what the entrants are allowed and not allowed to do. For example, they are not allowed to manually edit the rendered performances.

5.3 Interactive Section

The interactive section aims to build common ground for evaluating human performances by using computer systems.

| |
|---|
| Competition Session (2 hours) |
| Autonomous sections & Interactive section |
| 1. Score input & pre-processing |
| 2. Performance rendering |
| 3. System introduction by entrant |
| 4. Performance playing |
| 5. Audience evaluation |
| 6. Results |

Table 3. Time table for Stage II

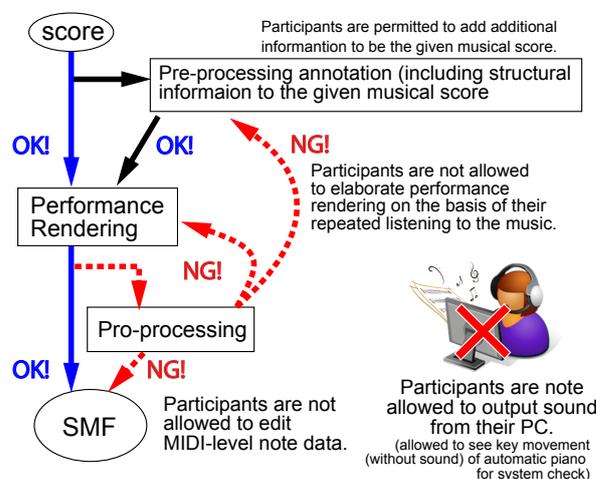


Figure 3. Performance rendering in autonomous section

Entrants will perform a musical piece using commercial music software or an original application.

As seen in Figure 4, the entrants are allowed to elaborate on the performance expressions at any step while listening to playback. They are also allowed to generate expressions by using mice, keyboards, or abstracted body movements like hand conducting. They are not allowed to directly play their musical instruments.

5.4 Set Piece and Rendering Procedure

The set piece is specified on the day of competition from a list of the following candidates. The candidate pieces will be shown at the Rencon webpage. The participants will be required to render two different styles of performance expressions.

5.5 Entered Systems

As of the beginning of May, five systems would have been entered in the autonomous section and three in the interactive section for Stage II. The latest information will be an-

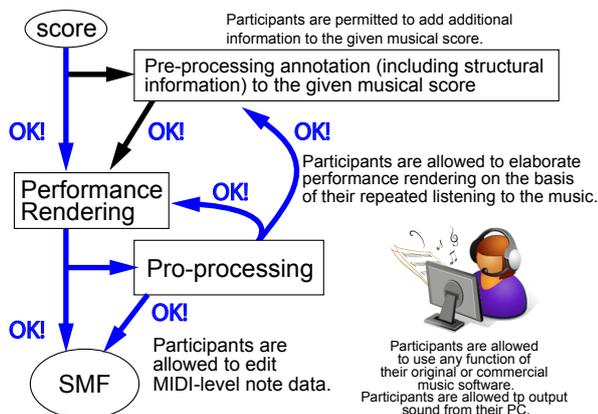


Figure 4. Performance rendering in interactive section

nounced at the Rencon webpage (<http://www.renconmusic.org/smc2011/>).

5.6 Evaluation

The rendered performances will be played by an automated grand piano in turn after the 60-min rendering period.

The performances for both the section will be evaluated by the audience, taking into account the degree of goodness. We are planning to broadcast the performances in Stage II on the Internet. Those who are watching it in real time will be able to join in with the online voting.

6. CONCLUSIONS

This paper introduced the current state of performance rendering and summarized the performance-rendering competition that is to be held at SMC2011. We expect this competition will trigger a discussion of interactive design for performance interfaces, music interpretation models, and their applications to music education. We aim to contribute to the development of modeling techniques for human mental activities, the formulation of musical performance expressions, its application to education, and the creation of novel music to enjoy. We intend to make many people aware of performance-generation systems.

Acknowledgments

The Rencon Committee sincerely thanks Professor Federico Avanzini and all the SMC2011 organizers for helping us arrange this event. We are grateful to Professor Tadahiro Muraio for composing the set piece for Stage I and providing us with useful advice. We also thank all the participants and attendees who plan to be present at SMC2011.

7. REFERENCES

[1] A. Kirke and E. R. Miranda, "A survey of computer systems for expressive music performance," *ACM Comput. Surv.*, vol. 42, pp. 3:1–3:41, December 2009.

[2] R. Hiraga, M. Hashida, K. Hirata, H. Katayose, and K. Noike, "Rencon: toward a new evaluation method

for performance," in *Proc. of International Computer Music Conference (ICMC)*, 2002, pp. 357–360.

- [3] L. Frydén and J. Sundberg, "Performance rules for melodies. origin, functions, purposes," in *International Computer Music Conference (ICMC) Proc.* ICMA, 1984, pp. 221–225.
- [4] M. Clynes, "A composing program incorporating microstructure," in *Proc. of International Computer Music Conference (ICMC)*, 1984, pp. 225–232.
- [5] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. MIT Press, 1983.
- [6] E. Narmour, *Beyond Schenkerism: The Need for Alternatives in Music Analysis*. Univ of Chicago, 06 1977.
- [7] J. Arcos, R. de Mantaras, and X. Serra, "Saxex: a case-based reasoning system for generating expressive musical performances," in *Proceedings of the International Computer Music Conference*, 1997, pp. 329–336.
- [8] R. Bresin and G. Battel, "Articulation strategies in expressive piano performance. analysis of legato, staccato, and repeated notes in performances of the andante movement of mozart's sonata in g major (k. 545)," *Journal of New Music Research*, vol. 29, no. 3, pp. 211–224, 2000.
- [9] O. Ishikawa, H. Katayose, and S. Inokuchi, "Identification of music performance rules based on iterated multiple regression analysis," *Journal of IPSJ*, vol. 43, no. 2, pp. 268–276, 2002, (written in Japanese).
- [10] G. Widmer, "Learning expressive performance: The structure-level approach," *Journal of New Music Research*, vol. 25, no. 2, pp. 179–205, 1996.
- [11] K. Teramura, H. Okuma, Y. Taniguchi, S. Makimoto, and S. Maeda, "Gaussian process regression for rendering music performance," in *International Conference on Music Perception and Cognition (ICMPC)*, 2008, pp. 167–172.
- [12] G. Widmer, S. Flossmann, and M. Grachten, "Yqx plays chopin," *AI Magazine*, vol. 30, no. 3, pp. 35–48, 2009.
- [13] T. Baba, M. Hashida, and H. Katayose, "A conducting system with heuristics of the conductor virtualphilharmony," in *Proc. of New Interfaces for Musical Expression (NIME)*, 2010 (CD-ROM).
- [14] "6.0 system." [Online]. Available: http://en.wikipedia.org/wiki/6.0_system

COMPARING INERTIAL AND OPTICAL MOCAP TECHNOLOGIES FOR SYNTHESIS CONTROL

Ståle A. Skogstad and Kristian Nymoen

fourMs - Music, Mind, Motion, Machines

Department of Informatics

University of Oslo

{savskogs, krisny}@ifi.uio.no

Mats Høvin

Robotics and Intelligent Systems group

Department of Informatics

University of Oslo

matsh@ifi.uio.no

ABSTRACT

This paper compares the use of two different technologies for controlling sound synthesis in real time: the infrared marker-based motion capture system *OptiTrack* and *Xsens MVN*, an inertial sensor-based motion capture suit. We present various quantitative comparisons between the data from the two systems and results from an experiment where a musician performed simple musical tasks with the two systems. Both systems are found to have their strengths and weaknesses, which we will present and discuss.

1. INTRODUCTION

Motion capture (MoCap) has become increasingly popular among music researchers, composers and performers [1]. There is a wide range of different MoCap technologies and manufacturers, and yet few comparative studies between the technologies have been published. Where one motion capture technology may outperform another in a sterilized laboratory setup, this may not be the case if the technologies are used in a different environment. Optical motion capture systems can suffer from optical occlusion, electromagnetic systems can suffer from magnetic disturbance, and so forth. Similarly, even though one motion capture system may be better than another at making accurate MoCap recordings and preparing the motion capture for offline analysis, the system may not be as good if the task is to do accurate motion capture in real time, to be used for example in controlling a sound synthesizer.

In this paper we compare the *real-time* performance of two motion capture systems (Figure 1) based on different technologies: *Xsens MVN* which is based on inertial sensors, and *OptiTrack* which is an infrared marker-based motion capture system (IrMoCap). Some of our remarks are also relevant to other motion capture systems than the ones discussed here, though the results and discussions are directed only toward *OptiTrack* and *Xsens*.

We will return to a description of these technologies in section 3. In the next section we will give a brief overview of related work. Section 4 will present results from comparisons between the two motion capture systems, which are then discussed in section 5.

Copyright: ©2011 Skogstad et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.



Figure 1. The NaturalPoint *OptiTrack* system (left) and the *Xsens MVN* system (right).

2. RELATED WORK AND BACKGROUND

Motion capture technologies have been used in musical contexts for a long time, and during the 00's we saw several examples of using various motion capture technologies for real-time control of sound. This includes electromagnetic motion capture [2], video-based motion capture [3], optical marker-based motion capture [4] and inertial motion capture [5], to mention a few.

Several researchers have reported on differences between motion capture technologies. Most of these reports, however, have been related to offline analysis for medical or animation purposes. Cloete et al. [6] have compared the kinematic reliability of the *Xsens MVN* suit with an IrMoCap system during routine gait studies. They conclude that the *Xsens MVN* system is comparable to IrMoCap systems but with shortcomings in some angle measurements. They also point out several practical advantages with the *Xsens* suit, like its wireless capabilities and quick set-up time. Another experiment by Thies et al. [7] found comparable acceleration values from two *Xsens* sensors and an IrMoCap system, and showed that calculating acceleration from the IrMoCap position data introduced noise. One of the conclusions from this experiment was that filtering methods need to be investigated further.

Miranda and Wanderley have pointed out some strengths and weaknesses with electromagnetic and optical motion capture systems [1]: Electromagnetic systems are able to track objects, even if it is not within the direct line of sight of external cameras. On the other hand, these systems need cables which may be obtrusive. Optical systems are superior to many other systems in terms of sampling rate, since they may track markers at sampling rates of more than 1000 Hz, and systems using passive markers have no need for obtrusive cables. Still, these systems need a direct line of sight between markers and cameras, and a passive

marker system may not be able to uniquely identify each marker.

Possibilities, strengths and weaknesses for real-time motion capture in musical contexts are discussed individually for IrMoCap and full-body inertial sensor systems in [8] and [9]. In this paper we will compare the real-time abilities of the two technologies.

2.1 Initial remarks on requirements when using MoCap for real-time control of music

A musical instrument is normally controlled with excitation and modification actions [10]. We can further distinguish between two types of excitations: discrete (i.e. trigger), or continuous (like bowing a string instrument). Dobrian [11] identifies two types of control data: triggers and streams of discrete data representing a sampling of a continuous phenomenon. Following these remarks, we are looking for a system able to robustly trigger sound events with good temporal accuracy, and to continuously control a system with good spatial accuracy and little noise. Consequently, we have chosen to emphasize three properties: spatial accuracy, temporal accuracy and system robustness. We will come back to measurements and discussion of these properties in sections 4 and 5.

3. TECHNOLOGIES

3.1 NaturalPoint OptiTrack

NaturalPoint OptiTrack is an optical infrared marker-based motion capture system (IrMoCap). This technology uses several cameras, equipped with infrared light-emitting diodes. The infrared light from the cameras is reflected by reflective markers and captured by each camera as 2D point-display images. By combining several of these 2D images the system calculates the 3D position of all the markers within the capture space. A calibration process is needed beforehand to determine the position of the cameras in relationship to each other, and in relationship to a global coordinate system defined by the user.

By using a combination of several markers in a specific pattern, the software can identify rigid bodies or skeletons. A *rigid body* refers to an object that will not deform. By putting at least 3 markers on the rigid body in a unique and non-symmetric pattern, the motion capture system is able to recognize the object and determine its position and orientation. A *skeleton* is a combination of rigid bodies and/or markers, and rules for how they relate to each other. In a human skeleton model, such a rule may be that the bottom of the right thigh is connected to the top of the right calf, and that they can only rotate around a single axis. In the NaturalPoint motion capture software (Arena), there exist 2 predefined skeleton models for the human body. It is not possible to set up user-defined skeletons.

3.2 The Xsens MVN

The Xsens MVN technology can be divided into two parts: (1) the sensor and communication hardware that are responsible for collecting and transmitting the raw sensor

data, and (2) the Xsens MVN software engine, which interprets and reconstructs the data to full body motion while trying to minimize positional drift.

The Xsens MVN suit [12] consists of 17 inertial MTx sensors, which are attached to key areas of the human body. Each sensor consists of 3D gyroscopes, accelerometers and magnetometers. The raw signals from the sensors are connected to a pair of Bluetooth 2.0-based wireless transmitters, which again transmit the raw motion capture data to a pair of wireless receivers.

The data from the Xsens MVN suit is fed to the MVN software engine that uses sensor fusion algorithms to produce absolute orientation values, which are used to transform the 3D linear accelerations to global coordinates. These in turn are translated to a human body model which implements joint constraints to minimize integration drift. The Xsens MVN system outputs information about body motion by expressing body postures sampled at a rate up to 120Hz. The postures are modeled by 23 body segments interconnected with 22 joints.

4. MEASUREMENTS

We carried out two recording sessions to compare the OptiTrack and Xsens systems. In the first session, a series of simple measurements were performed recording the data with both Xsens and OptiTrack simultaneously. These recordings were made to get an indication of the differences between the data from the systems. In the second session (Section 4.5), a musician was given some simple musical tasks, using the two MoCap systems separately to control a sound synthesizer.

4.1 Data comparison

Our focus is on comparing real-time data. Therefore, rather than using the built-in offline recording functionality in the two systems, data was streamed in real-time to a separate computer where it was time-stamped and recorded. This allows us to compare the quality of the data as it would appear to a synthesizer on a separate computer. Two terminal applications for translating the native motion capture data to Open Sound Control and sending it to the remote computer via UDP were used.

We have chosen to base our plots on the unfiltered data received from the motion capture systems. This might differ from how a MoCap system would be used in a real world application, where filtering would also be applied. Using unfiltered data rather than filtered data gives an indication of how much pre-processing is necessary before the data can be used for a musical application.

The Xsens suit was put on in full-body configuration. For OptiTrack, a 34-marker skeleton was used. This skeleton model is one of the predefined ones in the Arena software. Markers were placed outside the Xsens suit, which made it necessary to adjust the position of some of the markers slightly, but this did not alter the stability of the OptiTrack system.

Both systems were carefully calibrated, but it was difficult to align their global coordinate systems perfectly. This

is because OptiTrack uses a so-called L-frame on the floor to determine the global coordinate system, whereas Xsens uses the position of the person wearing the suit during the calibration to determine the origin of the global coordinate system. For this reason, we get a bias in the data from one system compared to the other. To compensate for this, the data has been adjusted so that the mean value of the data from the two systems more or less coincide. This allows us to observe general tendencies in the data.

4.2 Positional accuracy and drift

When comparing the Xsens and the OptiTrack systems there is one immediately evident difference. OptiTrack measures absolute position, while the sensors in the Xsens MVN suit can only observe relative motion. With Xsens, we are bound to experience some positional drift even though the system has several methods to keep it to a minimum [9].

4.2.1 Positional accuracy - still study

Figure 2 shows the position of the left foot of a person sitting in a chair without moving for 80 seconds. The upper plot shows the horizontal (XY) position and the lower plot shows vertical position (Z) over time. In the plot it is evident that Xsens suffers from positional drift, even though the person is sitting with the feet stationary on the floor. Xsens reports a continuous change of data, with a total drift of more than 0.2 m during the 80 seconds capture session. Equivalent plots of other limbs show similar drift, hence there is little relative drift between body limbs.

This measurement shows that OptiTrack is better at providing accurate and precise position data in this type of clinical setup. However, for the vertical axis, we do not observe any major drift, but the Xsens data is still noisier than the OptiTrack data.

4.2.2 Positional accuracy - walking path

The left plot in Figure 3 displays the horizontal (XY) position of the head of a person walking along a rectangular path in a large motion capture area recorded with Xsens. The plot shows a horizontal positional drift of about 2 meters during the 90 seconds capture session. Xsens shows

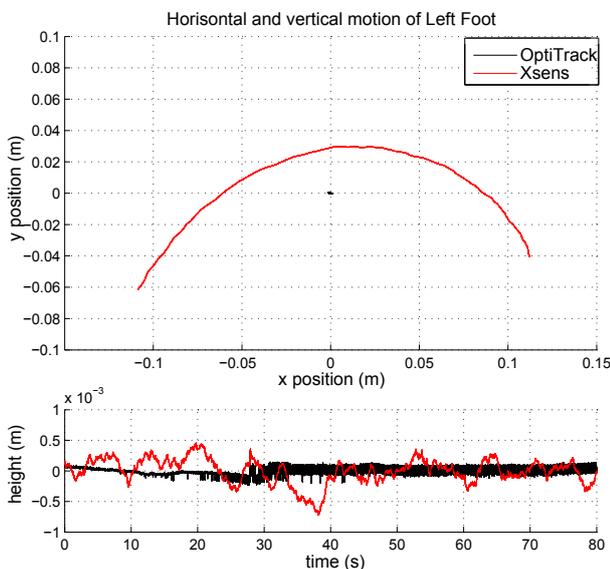


Figure 2. Horizontal and vertical plots of a stationary foot.

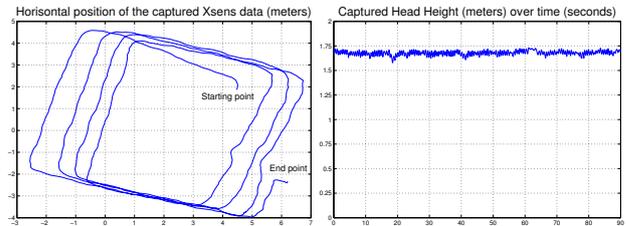


Figure 3. Recording of the horizontal (left) and vertical (right) position of the head.

no drift in the vertical direction (Z), as can be seen in the right plot. This is expected since the MVN engine maps the data to a human body model and assumes a fixed floor level. Because of the major horizontal drift we can conclude that Xsens MVN is not an ideal MoCap system if absolute horizontal position is needed.

4.2.3 Camera occlusion noise

The spatial resolution of an IrMoCap system mainly relies on the quality of the cameras and the calibration. The cameras have a certain resolution and field of view, which means that the spatial resolution of a marker is higher close to the camera than far away from the camera. The calibration quality determines how well the motion capture system copes with the transitions that happen when a marker becomes visible to a different combination of cameras. With a “perfect” calibration, there might not be a visible effect, but in a real situation we experience a clearly visible change in the data whenever one or more cameras fail to see the marker, as shown in Figure 4. When a marker is occluded from a camera, the 3D calculation will be based on a different set of 2D images.

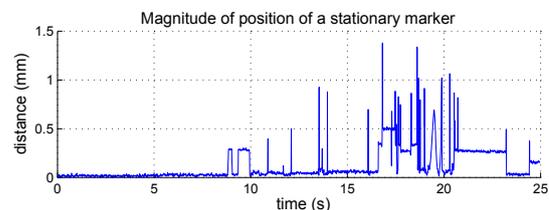


Figure 4. OptiTrack: Magnitude of the distance from the mean position of a stationary marker. The disturbances in the last part of the measurement is caused when a person moves around the marker, and thus blocks the marker in one or more cameras at a time. FrameRate 100 Hz

4.2.4 Xsens floor level change

If the motion capture area consists of different floor levels, like small elevated areas, the Xsens MVN engine will match the sensed raw data from the suit against the floor height where the suit was calibrated. This can be adjusted in post-processing, but real-time data will suffer from artifacts during floor level changes, as shown in Figure 5.

4.3 Acceleration and velocity data

In our experience, velocity and acceleration are highly usable motion features for controlling sound. High peaks in absolute acceleration can be used for triggering events, while velocity can be used for continuous excitation.

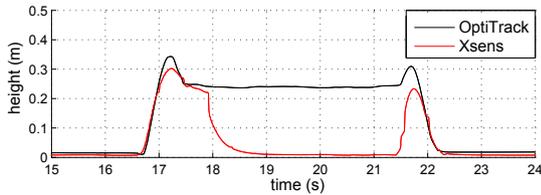


Figure 5. Recording of the vertical position of the left foot of a person, stepping onto an elevated area (around 0.25 m high). When the user plants his left foot on the object, the Xsens MVN engine will eventually map the stationary foot to floor level (18 to 19 s).

A difference between the two MoCap systems is that the Xsens system can offer velocity and acceleration data directly from the MVN engine [9]. When using the OptiTrack system we need to differentiate position data to estimate velocity and acceleration. If the positional data is noisy, the noise will be increased by differentiation (act as a high-pass filter), as we can see from Figure 6. The noise resulting from optical occlusion (see Section 4.2.3) is probably the cause for some of OptiTrack’s positional noise.

Even though the Xsens position data is less accurate, it does offer smoother velocity and, in particular, acceleration data directly. We can use filters to smooth the data from the OptiTrack system; however, this will introduce a system delay, and hence increased latency.

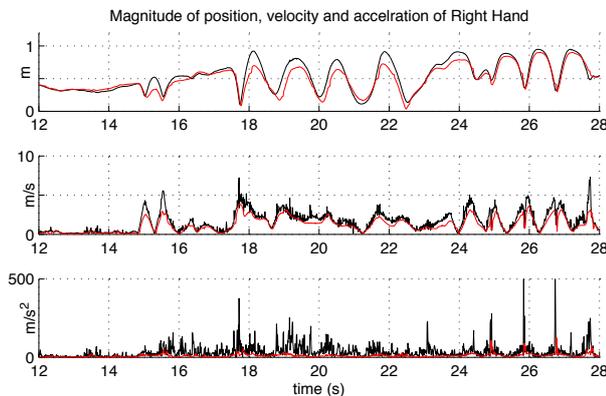


Figure 6. Velocity and acceleration data quality comparison (OptiTrack in black and Xsens in red).

4.4 Action-to-sound: latency and jitter

Low and stable latency is an important concern for *real-time* musical control [13], particularly if we want to use the system for triggering temporally accurate musical events. By *action-to-sound latency* we mean the time between the sound-producing action and the sonic reaction from the synthesizer.

To be able to measure the typical expected latency in a setup like that in Figure 7 we performed a simple experiment with an audio recorder. One computer was running one of the MoCap systems and sent OSC messages containing the MoCap information about the user’s hands. A patch in Max/MSP was made that registered hand claps

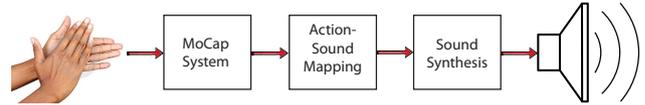


Figure 7. The acoustic hand clap and the triggered sound were recorded to measure latency of the systems.

based on MoCap data and triggered a *click* sound for each clap. The time difference between the acoustic hand clap and the triggered sound should indicate the typical expected latency for the setup.

Both MoCap systems were run on the same PC¹. The sound-producing Max/MSP patch was run on a separate Mac laptop² and received OSC messages from the MoCap systems through a direct Gbit Ethernet link. All experiments used the same firewire connected sound card, *Edirol FA-101*, as output source. The hand claps and the click output from the Max patch was recorded with a microphone. Statistical results from the time delays between hand claps and corresponding click sound in the recorded audio files are given in Table 1. The values are based on 30 claps each. In this experiment, OptiTrack had a faster sound output response and a lower standard deviation than Xsens. The standard deviation is included as an indication of the jitter performance of the MoCap systems, since lower standard deviation indicates higher temporal precision.

Higher Xsens latency and jitter values are probably partly due to its use of Bluetooth wireless links. The Xsens MVN system also offers a direct USB connection option. We performed the same latency test with this option; and the results indicate that the connection is around 10-15 milliseconds faster, and has a lower jitter performance, than the Bluetooth link.

The upper bounds for “intimate control” have been suggested to be 10ms for latency and 1ms for its variations (jitter) [13]. If we compare the bounds with our results, we see that both systems have relatively large latencies. However, in our experience, a latency of 50ms is still usable in many cases. The high jitter properties of the Xsens system are probably the most problematic, especially when one wants high temporal accuracy.

| | min | mean | max | std. dev. |
|-----------------|-----|------|-----|-----------|
| OptiTrack | 34 | 42.5 | 56 | 5.0 |
| Xsens Bluetooth | 41 | 52.2 | 83 | 8.4 |
| Xsens USB | 28 | 37.2 | 56 | 6.9 |

Table 1. Statistical results of the measured action-to-sound latency, in milliseconds.

4.5 Synthesizer control

In a second experiment, a musician was asked to perform simple music-related tasks with the two motion capture

¹ Intel 2.93 GHz i7 with 8GB RAM running Win 7

² MacBook Pro 10.6.6, 2.66 GHz Duo with 8GB RAM

systems. Three different control mappings to a sound synthesizer were prepared:

- Controlling pitch with the distance between the hands
- Triggering an impulsive sound based on high acceleration values
- Exciting a sustained sound based on the velocity of the hand

For the pitch mapping, the task was to match the pitch of one synthesizer to the pitch of another synthesizer moving in the simple melodic pattern displayed in Figure 8, which was repeated several times. This task was used to evaluate the use of position data from the two systems as the control data.

For the triggering mapping, the task was to follow a pulse by clapping the hands together. This task was given to evaluate acceleration data from the two systems as the control data, and to see if the action-to-sound latency and jitter would make it difficult to trigger events on time.

The excitation mapping was used to follow the loudness of a synthesizer, which alternated between "on" and "off" with a period of 1 second. This task was used to evaluate velocity data as control data.

The *reference sound* (the sound that the musician was supposed to follow) and the *controlled sound* (the sound that was controlled by the musician) were played through two different loudspeakers. The two sounds were also made with different timbral qualities so that it would be easy to distinguish them from each other. The musician was given some time to practice before each session. To get the best possible accuracy, both systems were used at their highest sampling rates for this experiment: Xsens at 120 Hz, and OptiTrack at 100 Hz.



Figure 8. The simple melody in the pitch-following task. This was repeated for several iterations.

4.5.1 Pitch-following results

We found no significant difference between the performances with the two systems in the pitch-following task. Figure 9 displays an excerpt of the experiment, which shows how the participant performed with both Xsens and OptiTrack. The participant found this task to be difficult, but not more difficult for one system than the other. Also, the data shows no significant difference in the performances with the two systems. This indicates that the quality of relative position values (between markers/limbs) is equally good in the two systems for this kind of task.

4.5.2 Triggering results

Table 2 shows the results of the latency between the reference sound and the controlled sound for the triggering test. They are based on 40 hand claps for each of the two

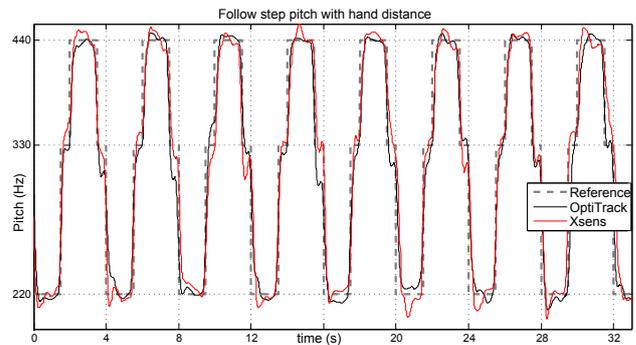


Figure 9. There was no significant difference between the two systems for the pitch-following task.

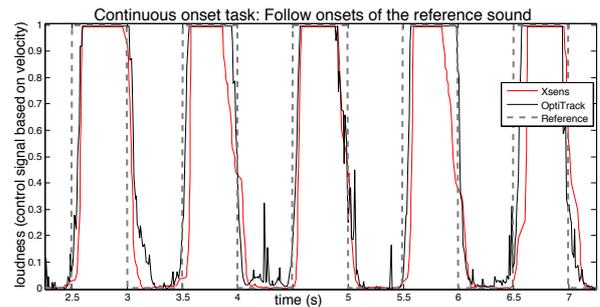


Figure 10. The major difference between the two systems in the continuous onset task was the noisy data from the OptiTrack system, which made it difficult to be quiet between the onsets. Apart from this, there was no big difference between the two systems.

MoCap systems. As we can see, the *mean* latency value is almost equal for Xsens and OptiTrack. Xsens has a higher standard deviation, which may indicate that the Xsens jitter shown in Table 1 makes it difficult for the user to make a steady trigger pulse.

| | min | mean | max | std. dev. |
|-----------|------|------|------|-----------|
| OptiTrack | 18.5 | 45.2 | 77.1 | 13.8 |
| Xsens | 2.6 | 44.7 | 96.3 | 28.3 |

Table 2. Statistical results, in milliseconds, of the measured time differences between reference signal and control signal.

4.5.3 Continuous onset results

For the continuous onset task, where the loudness of the sound was controlled by the absolute velocity of the right hand, we also observed a time delay between the onset of the reference tone and the onset of the sound played by our performer. This delay was present for both systems. In this task, the OptiTrack system suffered from noise, which was introduced when calculating the absolute velocity of the unfiltered OptiTrack data, as described in Section 4.3 (see Figure 10). The musician said that this made it more difficult to be quiet between the reference tones, and that this task was easier to perform with the Xsens system.

5. DISCUSSION

We have seen several positive and negative aspects with the quantitative measurements of the two technologies. In this section we will summarize our experiences of working with the two systems in a music-related context.

The main assets of the Xsens suit is its portability and wireless capabilities. The total weight of the suit is approximately 1.9 kg and the whole system comes in a suitcase with the total weight of 11 kg. Comparably, one could argue that a 8-camera OptiTrack setup could be portable, but this system requires tripods, which makes it more troublesome to transport and set up. OptiTrack is also wireless, in the sense that the user only wears reflective markers with no cables, but the capture area is restricted to the volume that is covered by the cameras, whereas Xsens can easily cover an area with a radius of more than 50 meters. When designing a system for real-time musical interaction based on OptiTrack, possible marker dropouts due to optical occlusion or a marker being moved out of the capture area must be taken into account. For Xsens, we have not experienced complete dropouts like this, but the Bluetooth link is vulnerable in areas with heavy wireless radio traffic, which may lead to data loss. Nevertheless, we consider Xsens to be the more robust system for on-stage performances.

OptiTrack has the benefit of costing less than most other motion capture technologies with equivalent resolution in time and space. The full Xsens suit is not comfortable to wear for a longer time period, whereas OptiTrack markers impose no or little discomfort. On the other hand, OptiTrack markers can fall off when tape is used to attach them. Also, OptiTrack's own solution for hand markers, where a plastic structure is attached to the wrist with Velcro, tends to wobble a lot, causing very noisy data for high acceleration movement, something we experienced when we set up the hand clapping tests. Xsens has a similar problem with the foot attachments of its sensors, which seems to cause positional artifacts.

Sections 4.2 to 4.5 show a number of differences between Xsens and OptiTrack. In summary, OptiTrack offers a higher positional precision than Xsens without significant drift, and seemingly also lower latency and jitter. Xsens delivers smoother data, particularly for acceleration and velocity. Our musician subject performed equally well in most of the musical tasks. However, the noisy OptiTrack data introduced some difficulties in the continuous onset task, and also made it challenging to develop a robust algorithm for the triggering task. Furthermore, Xsens jitter made the triggering task more difficult for the musician.

6. CONCLUSIONS

Both OptiTrack and Xsens offer useful MoCap data for musical interaction. They have some shared and some individual weaknesses, and in the end it is not the clinical data that matters, but the intended usage. If high positional precision is required, OptiTrack is preferable over Xsens, but if acceleration values are more important, Xsens provide less noisy data without occlusion problems. Overall, we find Xsens to be the most robust and stage-friendly Mo-

Cap system for real-time synthesis control.

7. REFERENCES

- [1] E. R. Miranda and M. Wanderley, *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard*. A-R Editions, Inc., 2006.
- [2] J. Michel Couturier and D. Arfib, "Pointing fingers: Using multiple direct interactions with visual objects to perform music," in *Proc. NIME*, 2003, pp. 184–188.
- [3] G. Castellano, R. Bresin, A. Camurri, and G. Volpe, "Expressive control of music and visual media by full-body movement," in *Proc. NIME*. New York, USA: ACM, 2007, pp. 390–391.
- [4] F. Bevilacqua, J. Ridenour, and D. J. Cuccia, "3d motion capture data: motion analysis and mapping to music," in *Proc. Workshop/Symposium SIMS*, California, Santa Barbara, 2002.
- [5] P.-J. Maes, M. Leman, M. Lesaffre, M. Demey, and D. Moelants, "From expressive gesture to sound," *Journal on Multimodal User Interfaces*, vol. 3, pp. 67–78, 2010.
- [6] T. Cloete and C. Scheffer, "Benchmarking of a full-body inertial motion capture system for clinical gait analysis," in *EMBS*, 2008, pp. 4579–4582.
- [7] S. Thies, P. Tresadern, L. Kenney, D. Howard, J. Goulermas, C. Smith, and J. Rigby, "Comparison of linear accelerations from three measurement systems during reach & grasp," *Medical Engineering & Physics*, vol. 29, no. 9, pp. 967–972, 2007.
- [8] S. A. Skogstad, A. R. Jensenius, and K. Nymoen, "Using IR optical marker based motion capture for exploring musical interaction," in *Proc. NIME*, Sydney, Australia, 2010, pp. 407–410.
- [9] S. A. Skogstad, K. Nymoen, Y. de Quay, and A. R. Jensenius, "Osc implementation and evaluation of the xsens mvn suit," in *Proc of NIME*, Oslo, Norway, 2011.
- [10] A. R. Jensenius, M. M. Wanderley, R. I. Godøy, and M. Leman, "Musical gestures: concepts and methods in research," in *Musical Gestures: Sound, Movement, and Meaning*, R. I. Godøy and M. Leman, Eds. New York: Routledge, 2010, pp. 12–35.
- [11] C. Dobrian, "Aesthetic considerations in the use of 'virtual' music instruments," in *Proc. Workshop on Current Research Directions in Computer Music*, 2001.
- [12] D. Rosenberg, H. Luinge, and P. Slycke, "Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors," *Xsens Technologies*, 2009.
- [13] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," in *Proc. NIME*, Seattle, USA, 2001.

A TOOLBOX FOR STORING AND STREAMING MUSIC-RELATED DATA

Kristian Nymo

fourMs - Music, Mind, Motion, Machines
Department of Informatics
University of Oslo
krisny@ifi.uio.no

Alexander Refsum Jensenius

fourMs - Music, Mind, Motion, Machines
Department of Musicology
University of Oslo
a.r.jensenius@imv.uio.no

ABSTRACT

Simultaneous handling and synchronisation of data related to music, such as score annotations, MIDI, video, motion descriptors, sensor data, etc. requires special tools due to the diversity of the data. We present a toolbox for recording and playback of complex music-related data. Using the Sound Description Interchange Format as a storage format and the Open Sound Control protocol as a streaming protocol simplifies exchange of data between composers and researchers.

1. INTRODUCTION

In this paper we introduce a set of tools that have been developed for working with music-related data. Our goal with this software is primarily to provide a set of tools for researchers working with music-related body motion, but we also see the potential for using the tools in other research areas. We started working on the tools in 2008, and the development has continued over the last years together with our research on music and movement [1, 2, 3]. The need for a common method of storing and sharing data related to musical movement was discussed at a panel session at the International Computer Music Conference 2007 [4], and further emphasised at a seminar in May 2010 at IRCAM, Paris, where several researchers from around the world working with music and motion, and sound spatialisation were present. A common denominator for this seminar was to come closer to a scheme for describing spatio-temporal aspects of music. The tools we are presenting were revised after this seminar with the intention of making them easy to use for the research community.

Section 2 introduces previous research and gives an overview of why these tools are needed, and what has already been done in the field. In section 3, the different types of data we are working with are discussed. Section 4 introduces the tools. Finally, in section 5, we conclude and point out the future directions of the development.

Copyright: ©2011 Nymo et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. BACKGROUND AND MOTIVATION

In our research on music-related body motion, we are often faced with situations where we want to study data from several devices at the same time. We will start this section by looking at two use cases that summarise some of the challenges in the field and the tools needed.

2.1 Use cases

a. The music researcher

A researcher is interested in studying the movement of a pianist by using an optical infrared motion capture system and record MIDI events from the piano. By themselves, the MIDI and motion capture data is trivial to record. However, synchronising the two, and being able to play them back later, or even scrubbing through the recording, keeping the MIDI-data and the motion capture data aligned, is not as trivial. Motion capture data is typically recorded at a sampling rate of 100–500 Hz, while the MIDI data stream is event driven and only needs to be stored each time a MIDI event takes place. Thus, using a common sampling rate for MIDI data and motion capture data would mean recording a lot of redundant data. The setup becomes even more complex when the researcher wants to record data from other sensors and audio/video as well.

b. The composer

A composer wants to develop a system for modifying sound in real-time. Let us say that the composer has hired a violin player who is wearing a position sensor and using a bow equipped with an accelerometer. She wants to develop a system that modifies the violin sound in real-time, based on output from the position sensor and the bow accelerometer data. Having the musician available at all times to perform can be expensive, as the musician would typically have to spend quite a lot of time waiting for the composer to make adjustments in the mapping between motion and sound. The composer would benefit from being able to record both the sound and the sensor data, and to play them back as a single synchronised performance.

Both of these examples show us that there is a need for a flexible system that is able to record different types of data from an arbitrary number of devices simultaneously. Further complexity is added when multiple representations of the same data is required. For instance, the researcher could be interested in the coordinates of the hands of a piano player in relation to a global coordinate system, but

also in relation to a coordinate frame defined by the position of the piano, or the center of mass in the pianist's body. The natural complexity of music introduces needs for various simultaneous representations of the same data.

Existing formats for working with these types of data have advantages and disadvantages, and there is no agreement between researchers on how to share music-related motion data. For motion capture data, the most widespread format is C3D.¹ Unfortunately, C3D does not allow for storing or synchronising music-related data and media. The Gesture Motion Signal² format has been developed to handle low level data in a musical context, but does not handle higher level data. The latter is handled well with the Performance Markup Language,³ but this format does not meet our requirements when it comes to audio and video synchronisation.

An approach similar to or own has been implemented in OpenMusic [5]. Bresson et al. have implemented a solution for storing and streaming sound spatialisation data in the Sound Description Interchange Format (SDIF). This seems to be a promising solution, and we hope to keep collaborating on SDIF descriptors for spatio-temporal data.

2.2 GDIF

The Gesture Description Interchange Format (GDIF) has been proposed for handling the diversity of data related to music and motion [6]. The name GDIF might be somewhat misleading, as this is neither a format per se, nor is it limited to only gesture-related data. Rather, it is a concept and an idea for how data, and particularly data related to musical movement, can be described and shared among different researchers.

This concept includes a hierarchical structure, where the raw data (i.e. the data that one receives directly from the sensor or interface) is stored at the bottom layer. Above this layer is a so-called *cooked layer*, where certain processing has taken place. This can be anything from simple filtering or transformations, to more advanced analysis. Other layers may include segmentations or chunks [7] and even higher-level descriptors such as expressivity, affect and mood.

So far, GDIF development has been concerned with conceptual issues, and it has been up to the user to define how to implement storage and streaming. Some guidelines have been suggested, one of them being the approach implemented in the system we are presenting in this paper. We are using the Sound Description Interchange Format for storing and the Open Sound Control protocol for streaming GDIF data [4]. These formats will be presented in sections 2.3 and 2.4.

2.3 SDIF

The Sound Description Interchange Format (SDIF) was proposed by researchers at IRCAM and CNMAT and has been suggested as a format for storing GDIF data [4, 8]. This file format describes a sequence of time-tagged *frames*.

¹ <http://www.c3d.org/>

² <http://acroe.imag.fr/gms/>

³ <http://www.n-ism.org/Projects/pml.php>

Each frame consists of an identifier indicating what type of frame it is, the frame size, the actual data and zero-padding to make the frame size a multiple of eight bytes [9]. The frames are further structured into *streams*. These streams are series of frames, and all streams share a common timeline. Inside each frame, the actual data is stored as strings, bytes, integers or floating point values in one or more 2D matrices.

2.4 Open Sound Control

Open Sound Control (OSC) is a protocol for real-time audio control messages [10]. Conceptually, OSC shares many similarities with the SDIF format, as it describes a way of streaming time-tagged bundles of data. Each bundle contains one or more *OSC messages*, each message containing an *OSC address* and the actual data in a list format. The OSC address contains a hierarchical structure of human readable words, separated by slashes, making it simple to work with and share data between researchers and musicians (e.g. `/mySynth/pitch 120`).

3. DATA TYPES

We are working with many different sorts of data. Part of GDIF development is to define data types that are as generic and at the same time as well defined as possible. In other words, data types in GDIF recordings must be defined in such a way that they are open enough for different use, and at the same time detailed enough to leave little or no doubt about what sort of data that is contained in a GDIF stream.

Frames and matrices in SDIF streams are identified by a four letter type tag. This introduces some challenges when it comes to describing data. By convention, the first letter should be X for non-standard SDIF streams, leaving us with three letters to define the frame type and matrix type we are working with. Although it makes sense to distinguish between the two, our current implementation makes no distinction between the frame type and the matrix type. This means that the current system only allows a single data matrix inside each frame, and the frame automatically adapts the type tag from the matrix it contains. This has been sufficient in our use so far, but it would make more sense to let the frame type identify the stream (e.g. according to input device) and the matrix types define the data within each matrix (e.g. position, orientation, etc.).

For our matrix type tags, we have chosen to let the second letter determine the main data category, e.g. "P" for position data. The third letter denotes the dimensionality of the data, e.g. "2" if we are only tracking horizontal position. The fourth letter lets us know if the stream contains delta values of the original data. This number denotes derivative level, for instance "1" if the stream is the first derivative of the original data. This means that an XP32 matrix would contain 3-dimensional data, of the second derivative from the original position stream (i.e. acceleration).

We are sometimes interested in the absolute value of a vector, i.e. the length of the vector independent of the direction. This type of matrix is denoted by replacing the

third letter in the type tag with an ‘‘A’’. To formalise, this gives us the general case:

$$XPjd[n] = XPj(d - 1)[n] - XPj(d - 1)[n - 1]$$

$$XPAd[n] = \sqrt{\sum_{i=1}^j XPjd[n][i]^2}$$

and as an example, the specific case:

$$XP31[n] = XP30[n] - XP30[n - 1]$$

$$XPA1[n] = \sqrt{\sum_{i=1}^3 XP31[n][i]^2}$$

where d denotes the derivative level, n denotes the frame index in a sequence of frames, i denotes the dimension index at frame n , and j denotes dimensionality of the stream.

In addition to streams describing position, velocity, etc., GDIF data types include everything from raw data from sensors to higher level descriptors. Table 1 displays a selection of the GDIF data types we are currently working with. A more complete list of data types can be found at the wiki that has been set up for GDIF and SpatDIF development.⁴ It should be noted that these are our suggestions, and we welcome a discussion on these data types.

| Tag | Description |
|------|--|
| XIDX | Referring to a certain event in a series of events, e.g. triggering a sound sample from a sample bank. |
| XP30 | 3-dimensional position stream. |
| XP31 | 3-dimensional position stream. 1st derivative. (i.e. velocity calculated from position data) |
| XPA1 | x-dimensional position stream. Absolute value of 1st derivative. |
| XOQ0 | Orientation stream, four quaternion values. |
| XA30 | 3D acceleration stream. Used when working with systems that provide acceleration data as raw data. |
| IMID | MIDI stream, already defined in the SDIF standard |
| XEMG | Electromyography sensor input. |
| XMQ0 | Quantity of motion stream. |
| XMA1 | Area of motion stream. First derivative. |

Table 1. A selection of GDIF data types.

The system accepts all measurement units. However, we recommend using the International System of Units (SI) whenever this is possible. This will make it easier for researchers to share GDIF recordings.

4. IMPLEMENTATION

The tools presented in this paper are based on the SDIF tools in the FTM library,⁵ mainly `ftm.sdif.write` for recording and `ftm.track` for playback [11]. They are implemented in Max as modules in the Jamoma⁶ framework. These frameworks provide solutions for OSC and SDIF. The two main modules in the toolbox are the recording module and the playback module.

⁴ http://xdif.wiki.ififi.uio.no/Data_types

⁵ <http://ftm.ircam.fr>

⁶ <http://www.jamoma.org>

The recording module, based on `ftm.sdif.write`, is designed for writing matrix-formatted data into separate streams in an SDIF file (Figure 1).



Figure 1. The record module

Different streams are separated by different OSC namespaces (e.g. `\stream\0`, `\stream\1`). The internal components of the recording module are created dynamically based on the user’s selection of streams from a drop-down menu in the GUI. The user may customise the stream types that are available in the drop-down menu by editing a simple text file. Using a script language that is specific to the Max environment, stream definition commands and data descriptions are generated dynamically and sent to the `ftm.sdif.write` object whenever the user inputs a command or selects streams. The internally used OSC-routing objects as well as the `ftm.sdif.write` object

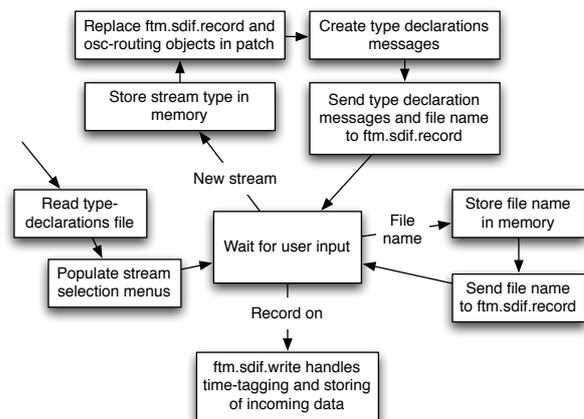


Figure 2. Simplified flowchart of the scripting system in the record module

The playback module displayed in Figure 3 is based on the `ftm.track` object. When an SDIF file is loaded into the playback module, an `ftm.track` object is created for each stream in the file. The data that is streamed from each track object is converted from the FTM float matrix format to Open Sound Control bundles using the OSC tools developed at CNMAT [10]. OSC does not support streaming matrices, hence each matrix row is separated as an instance number with its own OSC sub-address, e.g. first row gets the address `/XPOS/1`, second row `/XPOS/2`, etc. The user may set a custom buffer size for the OSC time tag to compensate for network latency and jitter. This buffer is set to a default value of 10 milliseconds.

The modules provide the user with a simple user inter-

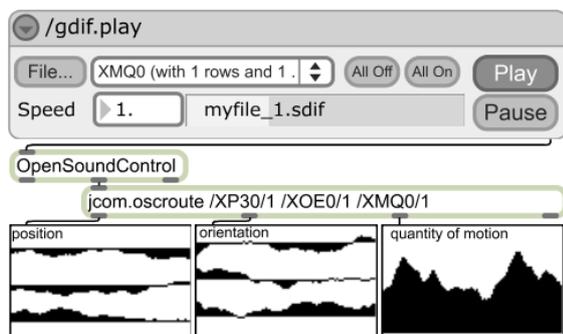


Figure 3. The playback module streaming a 3D position stream, an euler orientation stream and a quantity of motion stream

face. Rather than having to record data into separate unsynchronised buffers, the user can now record data into a single container without worrying about synchronisation. The presented tools are open source, and can be downloaded by checking out the Jamoma repository from github,⁷ or directly from the project website.⁸ Since the data is recorded as SDIF files, users may benefit from tools like EasDIF⁹ for analysis and post processing.

5. CONCLUSIONS AND FUTURE WORK

This paper has presented challenges we are facing when studying music-related body motion, and our solution to some of these problems in the form of a software toolbox. This toolbox includes a flexible module for making synchronized recordings of music-related data, and a module for playing back the data in real-time. The implementation makes the GDIF recording setup fast and easy, and makes this type of technology available to less experienced Max users.

Future development includes:

- Separating frame types as independent definitions. This will allow describing the stream type according to the device (e.g. a motion capture stream), and each frame can contain different data matrices (e.g. a position matrix and an orientation matrix).
- Human readable OSC namespace for data from the playback module (currently using the SDIF type tag).
- Integration of the Jamoma dataspaceLib for conversion between different data representations [12].
- Implementing simple data processing like automatic filtering and calculating absolute values.
- Develop a sequencer-like visual display, allowing zooming, editing, etc.
- Database for storing large collections of GDIF data.

6. ACKNOWLEDGEMENTS

Thanks to the developers of Max, Jamoma, FTM and OSC for providing a good frameworks for implementing these tools. Thanks also to the reviewers for valuable feedback.

⁷ <http://github.com/jamoma/Jamoma>

⁸ <http://www.fourms.uio.no/software/jamomagdif/>

⁹ <http://sourceforge.net/projects/sdif/>

7. REFERENCES

- [1] A. R. Jensenius, “GDIF development at McGill,” McGill University, Montreal, Canada, COST ConGAS – STSM report, 2007.
- [2] K. Nymoen, “A setup for synchronizing GDIF data using SDIF-files and FTM for Max,” McGill University, Montreal, Canada, COST SID – STSM report, 2008.
- [3] A. R. Jensenius, “Motion capture studies of action-sound couplings in sonic interaction,” KTH, Stockholm, Sweden, COST SID – STSM report, 2009.
- [4] A. R. Jensenius, A. Camurri, N. Castagne, E. Maestre, J. Malloch, D. McGilvray, D. Schwarz, and M. Wright, “Panel: the need of formats for streaming and storing music-related movement and gesture data,” in *Proceedings of the 2007 International Computer Music Conference*, Copenhagen, 2007.
- [5] J. Bresson, C. Agon, and M. Schumacher, “Représentation des données de contrôle pour la spatialisation dans openmusic,” in *Actes de Journées d’Informatique Musicale (JIM’10)*, 2010.
- [6] A. R. Jensenius, T. Kvitte, and R. I. Godøy, “Towards a gesture description interchange format,” in *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression*. Paris, France: Paris: IRCAM – Centre Pompidou, 2006, pp. 176–179.
- [7] R. I. Godøy, “Reflections on chunking in music,” in *Systematic and Comparative Musicology: Concepts, Methods, Findings. Hamburger Jahrbuch für Musikwissenschaft*. P. Lang, 2008, vol. 24, pp. 117–132.
- [8] M. Wright, A. Chaudhary, A. Freed, S. Khoury, and D. L. Wessel, “Audio applications of the sound description interchange format standard,” in *AES 107th Convention*, 1999.
- [9] M. Wright, A. Chaudhary, A. Freed, D. Wessel, X. Rodet, D. Virolle, R. Woehrmann, and X. Serra, “New applications of the sound description interchange format,” in *Proceedings of the 1998 International Computer Music Conference*, Ann Arbor, 1998, pp. 276–279.
- [10] M. Wright, A. Freed, and A. Momeni, “OpenSound Control: state of the art 2003,” in *Proceedings of the 2003 conference on New Interfaces for Musical Expression*, Montreal, Canada, 2003, pp. 153–160.
- [11] N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Müller, “FTM – complex data structures for Max,” in *Proceedings of the 2005 International Computer Music Conference*, Barcelona, 2005, pp. 9–12.
- [12] T. Place, T. Lossius, A. R. Jensenius, N. Peters, and P. Baltazar, “Addressing classes by differentiating values and properties in OSC,” in *Proceeding of the 8th International Conference on New Instruments for Musical Expression*, 2008.

AUTOMATIC CREATION OF MOOD PLAYLISTS IN THE THAYER PLANE: A METHODOLOGY AND A COMPARATIVE STUDY

Renato Panda

CISUC – DEI

University of Coimbra

panda@student.dei.uc.pt

Rui Pedro Paiva

CISUC – DEI

University of Coimbra

ruipedro@dei.uc.pt

ABSTRACT

We propose an approach for the automatic creation of mood playlists in the Thayer plane (TP). Music emotion recognition is tackled as a regression and classification problem, aiming to predict the arousal and valence (AV) values of each song in the TP, based on Yang's dataset. To this end, a high number of audio features are extracted using three frameworks: PsySound, MIR Toolbox and Marsyas. The extracted features and Yang's annotated AV values are used to train several Support Vector Regressors, each employing different feature sets. The best performance, in terms of R^2 statistics, was attained after feature selection, reaching 63% for arousal and 35.6% for valence. Based on the predicted location of each song in the TP, mood playlists can be created by specifying a point in the plane, from which the closest songs are retrieved. Using one seed song, the accuracy of the created playlists was 62.3% for 20-song playlists, 24.8% for 5-song playlists and 6.2% for the top song.

1. INTRODUCTION

Since the beginning of mankind music has always been present in our lives, serving a myriad of purposes both socially and individually. Given the major importance of music in all human societies throughout history and particularly in the digital society, music plays a relevant role in the world economy.

As a result of technological innovations in this digital era, a tremendous impulse has been given to the electronic music distribution industry. Factors like the widespread access to the Internet, bandwidth increasing in domestic accesses or the generalized use of compact audio, such as mp3, have contributed to that boom. The frenetic growth in music supply and demand uncovered the need for more powerful methods for automatically retrieving relevant songs in a given context from such huge databases. In fact, any large music database, or, generically speaking, any multimedia database, is only really useful if users can find what they are seeking in an efficient manner. Furthermore, it is also important that the organization of such a database can be performed as objectively and efficiently as possible.

Digital music repositories need, then, more advanced,

flexible and user-friendly search mechanisms, adapted to the requirements of individual users. In fact, "music's preeminent functions are social and psychological", and so "the most useful retrieval indexes are those that facilitate searching in conformity with such social and psychological functions. Typically, such indexes will focus on stylistic, mood, and similarity information." [1]. This is supported by studies on music information behavior that have identified music mood¹ as an important criterion for music retrieval and organization [2].

Besides the music industry, the range of applications of mood detection in music is wide and varied, e.g., game development, cinema, advertising or the clinical area (in the motivation to compliance to sport activities prescribed by physicians, as well as stress management).

Compared to music emotion synthesis, few works have been devoted to emotion analysis. From these, most of them deal with MIDI or symbolic representations [3]. Only a few works tackle the problem of emotion detection in audio music signals, although it has received increasing attention in recent years. Being a recent research topic, many limitations can be found and several problems are still open. In fact, the present accuracy of those systems shows there is plenty of room for improvement. In a recent comparison, the best algorithm achieved an accuracy of 65% in a task comprising 5 categories [4].

Several aspects make music emotion recognition (MER) a challenging task. On one hand, the perception of the emotions evoked by a song is inherently subjective: different people often perceive different, sometimes opposite, emotions. Besides, even when listeners agree in the kind of emotion, there's still ambiguity regarding its description (e.g., the adjectives employed). Additionally, it is not yet well-understood how and why music elements create specific emotional responses in listeners [5].

For a long time, mood and emotions has been a major subject of psychologists and so several theoretical models have been proposed over the years. Such models can be divided into two approaches: categorical models or dimensional models. Categorical models consist of several states of emotion (categories), such as anger, fear, happiness and joy. Dimensional models use several axes to map emotions into a plane. The most frequent approaches

¹ Even though mood and emotion can be defined differently, the two terms are used interchangeably in the literature and in this paper. For further details, see [4].

uses two axes (e.g. arousal-valence or energy-stress), with some cases of a third dimension (dominance).

The advantage of dimensional models is the reduced ambiguity when compared with the categorical approach. However, some ambiguity remains, since each of the four quadrants represents more than one distinct emotion (happiness and excitement are both represented by high arousal and valence for example). Given this, dimensional models can be further divided into discrete (described above) and continuous. Continuous models, unlike discrete ones, view the emotion plane as a continuous space where each point denotes a different emotional state, thus removing the ambiguity between emotional states [5].

In order to reduce ambiguity, Thayer's mood model [6] is employed. Hence, the emotion plane is regarded as a continuous space, with two axes: arousal and valence. Each point, then, denotes a different emotional state and songs are mapped to different points in the plane.

In this paper we aim to automatically generate playlists by exploiting mood similarity between songs in the Thayer plane, based only on features extracted from the audio signal. To this end, we built on Yang's work [5], where a regression solution to music emotion recognition was proposed.

Thus, our first goal is to predict AV values for each song in the set. We employed the annotated values from the dataset created by Yang [5]. From each song, a high number of audio features are extracted, with recourse to three frameworks: PsySound, MIR Toolbox and Marsyas. The extracted features and Yang's AV annotated values are used to train Support Vector Regressors (SVR), one for arousal and another for valence. Given the high number of extracted features, the feature space dimensionality is reduced via feature selection, applying two distinct algorithms: forward feature selection (FFS) [7] and RReliefF (RRF) [8]. The highest results were achieved with a subset of features from all frameworks, selected by FFS, reaching 63% for arousal and 35.6% for valence, in terms of R^2 statistics. Results with RRF were slightly lower recurring to a smaller subset of features. Compared to the results reported in [5], the prediction accuracy increased from 58.3% to 63% for arousal, and from 28.1% to 35.6% for valence, i.e., an improvement of 4.7% and 7.5%, respectively. A classification approach, using quadrants in the Thayer plane (TP) to train and prediction instead of AV values was also tested. Still, results were very similar between different feature sets, reaching 55% accuracy in terms of quadrant matching.

Our second goal is to automatically create mood-based playlists. A playlist is "a list that specifies which songs to play in which order." [9]. The sequence of songs has three important aspects: the elements, i.e., the songs in the sequence; the order in which these elements appear; and the length of the sequence. Unlike playing random songs or listening to complete albums, many times users want to listen to music according to their mood or to some activity they are involved in (e.g., relaxing or running). In this work, we select the elements in the playlist based on their distance to a seed song according to their location in the Thayer plane (Euclidean distance is calculated). In this way, the songs in the playlist are organized in increasing distance order to the seed song. Additional-

ly, the order of the songs can be specified with more flexibility by drawing a desired mood trajectory in the Thayer plane (see Section 4, Figure 3). As for the duration of the playlist, the number of songs to include is specified by the user.

The accuracy of this approach is measured by matching playlists generated with predicted AV values against playlists using the real AV values. With one seed song, the average accuracy of the created playlists is 62.3% for 20-song playlists, 24.8% for 5-song playlists and 6.2% for the top song only. We are not aware of any previous studies regarding the quantitative evaluation of mood-based playlists, so, to the best of our knowledge, this is an original contribution.

Finally, we have also built a working prototype to analyze music mood as well as to generate playlists based on a song or a mood trajectory (see Section 4, Figure 3).

This paper is organized as follows. In section 2, we describe relevant work that has been done in the area. In section 3, the feature extraction process and used frameworks are approached. Followed regression strategy and AV mood modeling is also addressed. In section 4, the quality of the ground truth is analyzed and experimental results are presented and discussed. Finally, conclusions from this study are drawn in section 5.

2. RELATED WORK

In 1989, Thayer proposed a two-dimensional mood model [6], offering a simple but effective way to represent mood. In this model, mood depends on two factors: Stress (happiness/anxiety) and Energy (calmness/energy) combined in a two-dimensional axis, forming four different quadrants: Contentment, representing calm and happy music; Depression, referring to calm and anxious music; Exuberance, referring to happy and energetic; and Anxiety, representing frantic and energetic music (see Figure 1). A key aspect of the model is that emotions are located away from the center, since closer to the center both arousal and valence have small values, thus not representing a clear, identifiable emotion. Thayer's mood model can fit in both sub-categories of dimensional models: it can be considered discrete, having four classes, but it can also be regarded as a continuous model, as approached by [5] and in this paper.

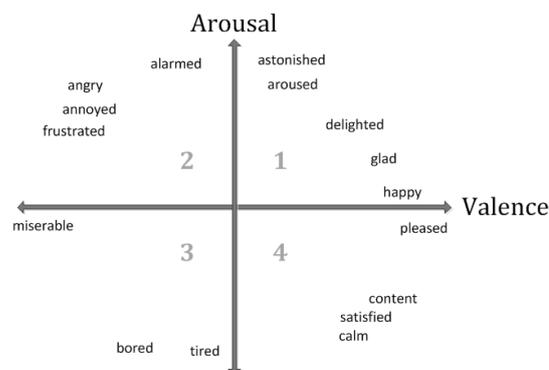


Figure 1. Thayer's model of mood (adapted from [10]).

Research on the relations between music and emotion has a long history, with initial empirical studies starting

in the 19th century [11]. This problem was studied more actively in the 20th century, when several researchers investigated the relationship between emotions and particular musical attributes such as mode, harmony, tempo, rhythm and dynamics [12]. However, only a few attempts have been made to build computational models. From these, most are devoted to emotion synthesis [13], elaborating on the relationships between emotion and music composition and music expressivity.

Only a few works addressing emotion detection in audio signals can be found. To the best of our knowledge, the first paper on mood detection in audio was published in 2003, by Feng et al. [14]. There, musical pieces are classified into 4 mood categories (happiness, sadness, anger and fear) using two musical attributes: tempo and articulation, extracted from 200 songs. These features are used to train a neural network classifier. The classifier is then validated on a test corpus of 23 pieces, with average precision and recall of 67 and 66%, respectively. This first attempt towards music mood detection had, naturally, several limitations. Namely, only two music attributes were captured and only four mood labels were employed. Regarding system validation, a reduced test corpus was utilized, making it hard to provide evidence of generality.

Most of the described limitations were still present in following research works (e.g., [13], [15], [16]). Overall, in each approach a different (and limited) set of features, mood taxonomies, number of classes and test sets are proposed. Also, some studies constrain the analysis to a particular musical style, e.g., [13], [16].

One of the most comprehensive approaches so far is proposed by Lu et al. [13]. The system is based on Thayer's model of mood, employing its 4 music moods and using features of intensity, timbre and rhythm. Mood is then detected with recourse to a hierarchical framework based on Gaussian Mixture Models and feature decorrelation via the Karhunen-Loeve Transform. The algorithm was evaluated on a test set of 800 songs, reaching 86.3% average accuracy. This value should be regarded with caution, since the system was only evaluated on a corpus of classical music using only 4 classes. Its main limitations are the absence of important mood-related features, such as mode and articulation, and its short number of mood categories. Some interesting points are the usage of a hierarchical framework and different weights for each feature, according to their.

Contrasting to most approaches, based on categorical mood models, Yang et al. [5] maps each music clip to a point in Thayer's arousal-valence plane. The authors evaluated their system with recourse to R^2 statistics, having achieved 58.3% accuracy for arousal and 28.1% for valence. We base our approach in this work.

In a recent evaluation that took place in MIREX'2010 [4], the accuracy of several algorithms in a 5-class mood classification task was compared. The best algorithm achieved 65% accuracy. For a more comprehensive survey on MER see [10]. To sum up, we can see, from the lack of accuracy and generality of the current approaches, that there is plenty of room for improvement.

Regarding automatic playlist generation (APG), most current approaches are based on the specification of one or more seed songs, creating playlists based on the dis-

tance between the seed and remaining songs, according to some distance function, e.g., [17-19]. Playlist ordering is usually defined according to the distance to the seed.

Other approaches rely on the usage of user-specified constraints based on metadata, e.g., [9], [20], [21]. Those constraints usually include criteria such as balance (e.g., don't allow two consecutive songs of the same artist) or progress (e.g., increase tempo or change genre at some point) among others [9]. Besides metadata-based constraints, audio similarity constraints can also be employed (e.g. timbre continuity through a playlist) [22].

In this paper, we follow the first approach, i.e., a seed song is specified and the playlist is created according to distances of the songs in the dataset to this seed song. Additionally, the order of the songs can also be specified by drawing a desired mood trajectory in the Thayer plane.

3. FEATURE EXTRACTION AND AV MOOD MODELING

3.1 Feature Extraction

Several authors have studied the most relevant musical attributes for mood analysis. Namely, it was found that major modes are frequently related to emotional states such as happiness or solemnity, whereas minor modes are associated with sadness or anger [23]. Simple, consonant, harmonies are usually happy, pleasant or relaxed. On the contrary, complex, dissonant, harmonies relate to emotions such as excitement, tension or sadness, as they create instability in a musical piece [23]. In a recent overview, Friberg [12] lists and describes the following features: timing, dynamics, articulation, timbre, pitch, interval, melody, harmony, tonality and rhythm. Other common features not included in that list are, for example, mode, loudness or musical form [23]. Several of these features have already been studied in the MIDI domain, e.g., [24]. The following list contains many of the relevant features for music mood analysis:

- Timing: Tempo, tempo variation, duration contrast
- Dynamics: overall level, crescendo/decrescendo, accents
- Articulation: overall (staccato/legato), variability
- Timbre: Spectral richness, onset velocity, harmonic richness
- Pitch (high/low)
- Interval (small/large)
- Melody: range (small/large), direction (up/down)
- Harmony (consonant/complex-dissonant)
- Tonality (chromatic-atonal/key-oriented)
- Rhythm (regular-smooth/firm/flowing-fluent/irregular-rough)
- Mode (major/minor)
- Loudness (high/low)
- Musical form (complexity, repetition, new ideas, disruption)

However, many of the previous features are often difficult to extract from audio signals. Also, several of them require further study from a psychological perspective. Therefore, it is common to apply low-level audio descrip-

tors (LLDs), studied in other contexts (e.g., genre classification, speech recognition), directly to mood detection. Such descriptors aim to represent attributes of audio like pitch, harmony, loudness, timbre, rhythm, tempo and so forth. LLDs are generally computed from the short-time spectra of the audio waveform, e.g., spectral shape features such as centroid, spread, skewness, kurtosis, slope, decrease, rolloff, flux, contrast or MFCCs [25]. Other methods have been studied to detect tempo and tonality.

To extract the referred features, an audio framework is normally used. The main differences between frameworks are the number and type of features available, stability, ease of use, performance and the system resources they require. In this work, features from PsySound, MIR Toolbox and Marsyas were used, measuring the relevance of each one in MER. Although PsySound is cited in some literature [5] as having several relevant features to emotion, there is no known comparison between this and other frameworks.

In his work [5], Yang used PsySound2 to extract a total of 44 features. At the time, PsySound was available only for Mac PowerPC computers. Since then, the program was rewritten in MATLAB, resulting in PsySound3. Still, the current version contains inconsistencies and lacks features present in the previous version, making it impossible to replicate Yang feature set and thus compare the results between PsySound2 and 3. For this reason, we employ the exact same PsySound2 features extracted and kindly provided by Yang. From PsySound, a set of 15 features are said to be particularly relevant to emotion analysis [26]. Therefore, another feature set was defined by Yang [5], containing these 15 features. This set is denoted as Psy15 hereafter, while the full PsySound, Marsyas and Music Information Retrieval (MIR) Toolbox feature sets will be denoted as Psy44, MAR and MIR respectively.

The MIR Toolbox is an integrated set of functions written in MATLAB, that are specific to the extraction of musical features such as pitch, timbre, tonality and others [27]. A high number of both low and high-level audio features are available.

Marsyas (Music Analysis, Retrieval and Synthesis for Audio Signals) is a framework developed for audio processing with specific emphasis on MIR applications. Marsyas has been used for a variety of projects in both academia and industry, and it is known to be computationally efficient, due in part to the fact of being written in highly optimized C++ code. On the less bright side, it lacks some features considered relevant to MER.

A brief summary of the extracted features and their respective framework is given in Table 1. Regarding Marsyas and MIR Toolbox, the analysis window size used for frame-level features is 23 ms, later transformed to song-level features by the MeanVar model [28], which represents the feature by mean and variance. All extracted features were normalized to the [0, 1] interval. A total of 12 features extracted with Marsyas returned the same (zero) value for all songs, thus not being used in the experiment.

| <i>Framework (features)</i> | <i>Description</i> |
|-----------------------------|--|
| PsySound2 (44) | Loudness, sharpness, volume, spectral centroid, timbral width, pitch multiplicity, dissonance, tonality and chord, based on psycho acoustic models. |
| MIR Toolbox (177) | Among others: root mean square (RMS) energy, rhythmic fluctuation, tempo, attack time and slope, zero crossing rate, rolloff, flux, high frequency energy, Mel frequency cepstral coefficients (MFCCs), roughness, spectral peaks variability (irregularity), inharmonicity, pitch, mode, harmonic change and key. |
| Marsyas (237) | Spectral centroid, rolloff, flux, zero cross rate, linear spectral pair, linear prediction cepstral coefficients (LPCCs), spectral flatness measure (SFM), spectral crest factor (SCF), stereo panning spectrum features, MFCCs, chroma, beat histograms and tempo. |

Table 1. Frameworks used and respective features.

3.2 AV Mood Modeling

A wide range of supervised learning methods are available and have been used in MER problems before. From those, we opted for regression algorithms as a solution, similarly to what was done by Yang. The idea behind regression is to predict a real value, based on a previous set of training examples, which proved to be a fast and reliable solution [29].

Since we employ Thayer’s model as a continuous representation of mood, a regression algorithm is used to train two distinct models – one for arousal and another for valence. To this end, the algorithm is fed with each song feature vector, as well as the AV values, previously annotated in Yang’s study. The created models can then be used to predict AV values for a given feature vector.

Support Vector Regression (SVR) was the chosen algorithm, since it achieved the best results in Yang’s study [5], when compared with Multiple Linear Regression (MLR) and AdaBoost.RT. We used the libSVM library [30], a fast and reliable implementation of SVR and classification (SVC). A grid parameter search was also carried out to discover the best SVR parameters.

To reduce the dimensionality of the feature space while increasing prediction accuracy, achieving a subset of features that are better suited to our problem, we tested two feature selection algorithms: Forward Feature Selection (FFS) [7] and RReliefF [8]. FFS is a simple algorithm, starting with an empty “ranked” set of features. All the remaining features are tested one at a time, moving the best performing one to the “ranked” set. The procedure continues iteratively, with one feature being added to the “ranked” set in each iteration, until no more features are left. One of its main limitations in FFS is the fact that it does not take into consideration the relation

that might exist between groups of features, resulting in big subsets of features. RRelief is another algorithm to measure features' importance. Unlike FFS, RRF does not assume feature independence. In addition, it also provides a weight to each feature in the problem under analysis. Since the algorithm uses k-nearest neighbors (KNN), a proper value of K is of major importance. Using a small value may give unreliable results. On the other hand, if K is high it may fail to highlight important features. Taking this into consideration, several values of K for each feature set were tested to obtain better results. Given the differences of each feature selection algorithm, it may be interesting to compare each ranking and respective performance.

The dimensionality of the feature space can also be reduced with recourse to Principal Component Analysis (PCA) [31]. This is a widely used technique whose basic idea is to project the computed feature matrix into an orthogonal basis that best expresses the original data set. Moreover, the resulting projected data is decorrelated. As for the selection of the principal components, we kept the ones that retained 90% of the variance. Regarding implementation, we made use of the PCA MATLAB code provided in the Netlab toolbox [32].

In order to measure performance of the regression models we used the R^2 statistics, "which is the standard way for measuring the goodness of fit for regression models" [5]. Moreover, we want a direct comparison between our results and Yang's. R^2 is defined as follows, (1):

$$R^2 = 1 - \frac{SSE}{SST}, \quad (1)$$

where SSE represents the sum square error (SSE) and SST the total sum of squares (SST). SSE measures the total deviation of the predicted values from the original annotations (2).

$$SSE = \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (2)$$

where y_i is the annotation and \hat{y}_i the predicted value. The SST is used to measure the deviation of each annotation to the mean value of the annotations (3).

$$SST = \sum_{i=1}^N (y_i - \bar{y})^2, \quad (3)$$

where y_i is the specific annotation and \bar{y} the average of all annotation values. An R^2 of 1 means the model fits the data perfectly, while negative values indicate that the model is even worse than simply taking the sample mean.

4. EVALUATION

4.1 Ground Truth Analysis

As previously mentioned, we employ the dataset and AV annotations kindly provided by Yang and used in his work [5]. The AV annotations are fundamental to the

results, since they are used in the regressor training process and to measure the playlist results. According to Yang, the dataset is made of 25 seconds clips, of various genres, that better expressed the emotion present on each song, for a total of 195 songs, balanced between quadrants. The ground truth was created using 253 volunteers with different backgrounds, in a subjective test, with each song being labeled by at least 10 different subjects. The volunteers were asked to annotate the evoking emotion in AV values, between [-1, 1]. Details on the subjective test can be found in [5].

There are several problems with the ground truth that may have a negative influence on the results. One of them is the proximity of the AV values with the origin of the graph. Thayer's model places the emotions far from the center, where the reference values are relevant, with a high positive or negative valence and arousal. However, most of the annotations are near the center, as shown in Figure 2, where 70% are at a distance smaller than 0.5. In it, the position of each point represents the average AV value given by annotators, while the marker type represents the expected quadrant for each song by Yang. One possible reason for this is the fact that the AV annotations result from averaging several annotations by different subjects, which can vary greatly, once again showing the subjectivity existent in emotions perception.

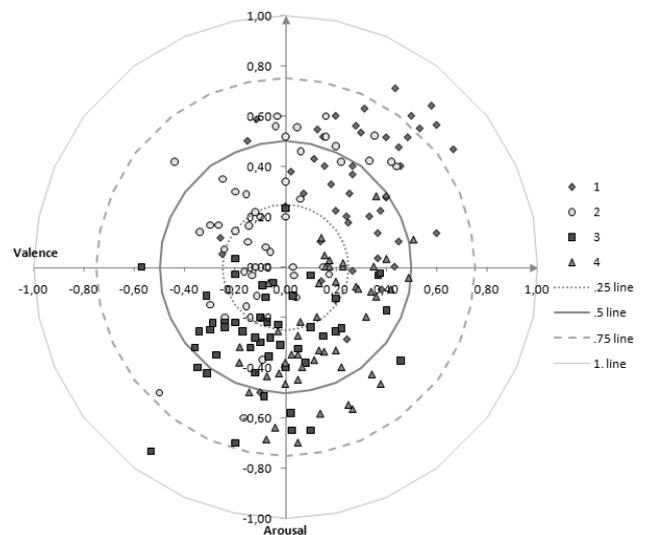


Figure 2. Yang annotations placed on Thayer's model

Another issue is the unbalanced song distribution. The original balance was disrupted since the AV values given by the subjects often placed the songs in a different quadrant than initially predicted by Yang. This affected specially the second quadrant, having only 12% of the songs.

Finally, a few inconsistencies were found between the provided data. Originally, the article [5] mentions 195 songs. However, due to some mismatches between arousal and valence annotations in the data provided by the author, we could only employ 189 songs. In any case, the difference is not significant.

4.2 Experimental Results

4.2.1 Mood Regression

In the regression tests, 20 repetitions of 10-fold cross validation were run, ensuring that all songs are used in different groups for training and testing.

Various tests were run in order to perceive the importance of each framework and its features on mood detection. From these tests, the best results were obtained with FFS, using a combination of all feature sets, reaching 63% for arousal and 35.6% for valence, using a total of 53 and 80 features respectively. The number of used features was high, in part due to the FFS working mode. Although RRF results were lower, they were in many cases obtained resorting to less features, helping us to identify the most important features for both problems (AV). For instance, using only the first ten features selected with RRF resulted in 31.5% for arousal and 15.2% for valence. On the other hand, FFS achieved only 0.8% and 2.0% for arousal and valence respectively..

The remaining tests highlighted MIR Toolbox features as achieving better results, especially on valence with R^2 attaining 25.7%. PsySound followed, with a valence accuracy of 21% and Marsyas scored the lowest, only 4.6%, proving to be quite ineffective for valence prediction. In terms of arousal, all the frameworks had a close score, ranging from 56% (Marsyas) to 60.3% (MIR Toolbox). A summary of the results is presented in Table 2 (values refer to average \pm variance). For some unknown reason, we were unable to replicate Yang results [5], using either the Psy15 features or the list of features resulting from the feature selection algorithm². We also conducted the same tests with PCA, normally used to reduce correlation between variables, without any noticeable improvement in results but actually leading to lower R^2 values.

| | <i>All features</i> | | <i>FFS</i> | | <i>RReliefF</i> | |
|-----------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | A | V | A | V | A | V |
| Psy15 | 58.7% ± 15.6 | 12.7% ± 18.4 | 60.3% ± 14.7 | 21.0% ± 15.4 | 60.1% ± 16.0 | 21.1% ± 16.4 |
| Psy44 | 57.3% ± 15.9 | 7.9% ± 14.0 | 57.3% ± 15.6 | 19.1% ± 13.4 | 60.5% ± 15.2 | 16.3% ± 15.0 |
| MIR | 58.2% ± 14.2 | 8.5% ± 19.5 | 58.7% ± 13.3 | 25.7% ± 18.9 | 62.1% ± 9.9 | 23.3% ± 15.7 |
| MAR | 52.9% ± 16.2 | 3.7% ± 14.9 | 56.0% ± 14.6 | 4.6% ± 20.2 | 60.0% ± 12.4 | 10.4% ± 10.7 |
| ALL + PCA | 56.5% ± 13.6 | 23.4% ± 18.2 | 61.8% ± 11.0 | 27.2% ± 22.5 | 61.4% ± 16.2 | 17.0% ± 20.6 |
| ALL | 57.4% ± 15.6 | 19.4% ± 12.3 | 62.9% ± 8.8 | 35.6% ± 14.7 | 62.6% ± 13.7 | 24.5% ± 14.3 |

Table 2. Results of the regression and classification tests.

A list of the top ten features for both arousal and valence is presented on Table 3. The list was obtained by

² It is worth mentioned that, in order to try to replicate Yang’s results, we employed the SVR parameters mentioned in his web page: <http://mpac.ee.ntu.edu.tw/~yihshuan/MER/taslp08/>.

running the RReliefF algorithm on the combined feature set of all frameworks (referred as “ALL” in Table 3).

| <i>Arousal</i> | | | <i>Valence</i> | | |
|------------------------------|-------|--------|--------------------------|-------|--------|
| Feature | Set | Weight | Feature | Set | Weight |
| SFM19 (std) | MAR | 0.0186 | spectral diss (S) | Psy15 | 0.0255 |
| RMS energy (kurtosis) | MIR | 0.0153 | tonality | Psy15 | 0.0239 |
| key strength minor (max) | MIR | 0.0139 | key strength major (max) | MIR | 0.0210 |
| MFCC2 (kurtosis) | MIR | 0.0136 | key clarity | MIR | 0.0158 |
| pulse clarity | MIR | 0.0135 | fluctuation (kurtosis) | MIR | 0.0147 |
| spectral kurtosis (skw) | MIR | 0.0129 | MFCC6 (skw) | MIR | 0.0132 |
| LAMin | Psy44 | 0.0128 | fluctuation (skw) | MIR | 0.0129 |
| spectral skewness (kurtosis) | MIR | 0.0126 | pulse clarity | MIR | 0.0118 |
| Nmin | Psy44 | 0.0112 | tonal centroid 1 (std) | MIR | 0.0118 |
| chroma (kurtosis) | MIR | 0.0110 | key strength major (std) | MIR | 0.0117 |

Table 3. Top ten features selected by RRF (using the combined feature set from the three frameworks).

4.2.2 Playlist Generation

As mentioned before, for playlist quality evaluation we tested a regressor-based distance strategy. In this method, distances are calculated using the predicted AV values returned by the regression models. The predicted distances were compared to the reference distances resulting from the real AV annotations.

To this end, the dataset was randomly divided in two groups, balanced in terms of quadrants. The first, representing 75% of the dataset was used to train the regressor. Next, the resulting model was used to predict AV values for the remaining 25% songs³. From this test dataset, a song is selected and serves as the seed for automatic playlist generation. Using the seed’s attributes, similarity against other songs is calculated. This originates two playlists ordered by distance to the seed, one based on the predicted and another on the annotated AV values. The annotations playlist is then used to calculate the accuracy of the predicted list, by matching the top 1, 5 and 20 songs. Here, we only count how many songs in each top are the same (e.g., for top5, a match of 60% means that the same three songs are present in both lists). The entire process is repeated 500 times, averaging the results.

Results obtained for playlist generation were very similar between the three audio frameworks. Several tests were run using all the combinations of features referred before. The similarity ranking was calculated using predicted values from the regressor. The best results were

³ This 75-25 division was necessary so that the validation set was not too short, as we want to evaluate playlists containing up to 20 songs. On the other hand, the 90-10 division was employed before for the sake of comparison with Yang’s results

accomplished using FFS for the combined feature set of all frameworks, with a matching percentage of 6.2% for top1, 24.8% for top5 and 62.3% for top20. Detailed results are presented in Table 4. The lower results in smaller playlists are mostly caused by the lack of precision when predicting valence. Still, best results are obtained with longer playlists, as normally used in a real scenario.

| | | Psy15 | Psy44 | MIR | MAR | ALL |
|-------|-----|----------------|----------------|----------------|----------------|----------------|
| Top1 | All | 4.2 ± 20.7 | 4.1 ± 18.6 | 3.6 ± 22.0 | 4.0 ± 20.7 | 4.2 ± 20.9 |
| | FFS | 5.6 ± 21.0 | 3.8 ± 18.6 | 5.2 ± 23.6 | 4.4 ± 19.8 | 6.2 ± 20.7 |
| | RRF | 5.1 ± 22.0 | 4.6 ± 19.0 | 5.6 ± 22.0 | 4.6 ± 22.6 | 5.2 ± 20.6 |
| Top5 | All | 21.1 ± 18.1 | 20.9 ± 17.1 | 22.8 ± 19.0 | 18.1 ± 17.6 | 21.0 ± 17.8 |
| | FFS | 21.5 ± 18.3 | 21.2 ± 17.9 | 22.0 ± 19.3 | 19.8 ± 18.5 | 24.8 ± 18.3 |
| | RRF | 21.9 ± 18.1 | 22.1 ± 17.9 | 23.3 ± 18.4 | 18.7 ± 17.8 | 23.3 ± 18.4 |
| Top20 | All | 61.9 ± 11.6 | 60.5 ± 12.3 | 62.7 ± 14.1 | 58.5 ± 13.6 | 60.7 ± 14.1 |
| | FFS | 62.0 ± 11.9 | 61.9 ± 12.4 | 62.5 ± 13.9 | 60.0 ± 13.6 | 62.3 ± 13.6 |
| | RRF | 61.0 ± 12.2 | 60.8 ± 12.8 | 61.7 ± 13.7 | 57.4 ± 13.0 | 61.6 ± 13.8 |

Table 4. Regression-based APG results (in %)

Finally, we have also built a working prototype to analyze music mood as well as to generate playlists based on a song or a mood trajectory. This is illustrated in Figure 3, where a desired mood trajectory was specified by drawing in the Thayer plane (black dots), giving rise to the playlist represented by the larger colored circles.

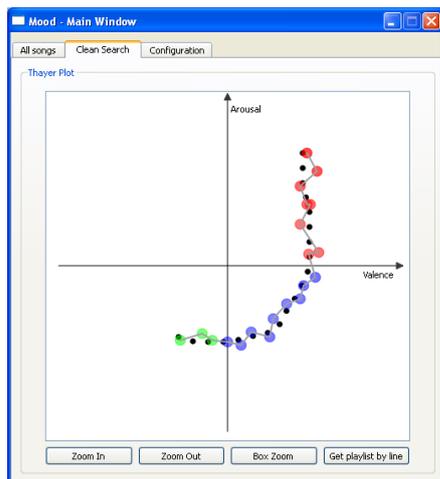


Figure 3. Automatic playlist generation prototype.

5. CONCLUSIONS

In this paper, we proposed an approach for the automatic creation of mood playlists in the Thayer plane, based on previous work by Yang [5] on continuous mood modeling.

Regarding AV prediction accuracy, we were able to outperform Yang’s previous results using forward feature

selection on a set of features extracted from three frameworks (PsySound, MIR Toolbox and Marsyas), reaching 63% average accuracy for arousal and 35.6% for valence, in terms of R^2 statistics. RReliefF was also important to highlight the most interesting features to the problem.

Regarding the playlist generation and similarity analysis, matching for top1 was low, averaging 5% between all frameworks, with top20 presenting some reasonable results, of around 60%. From all the tests, a slightly higher accuracy was attained using the FFS selection of features from the combination of all frameworks, with 6.2%, 24.8% and 62.3% for top1, top5 and top20 respectively. Still, the results are very similar between feature selection algorithms to classify one as better suited. The same is verified in relation to frameworks, with MIR Toolbox having a slight advantage.

In both cases, to decrease the influence that the outliers may have in the results we pretend to repeat the tests using median values instead of the current arithmetic mean. Despite the achieved improvements, we can see, from the lack of accuracy and generality of both our and other current approaches, that there is plenty of room for improvement. Also, several key open problems can be identified, namely in terms of extraction, selection and evaluation of meaningful features in the context of mood detection in audio music, extraction of knowledge from computational models (as all known approaches are black-box) and the tracking of mood variations throughout a song. In order to tackle the current limitations, we believe the most important problem to address is the development of novel acoustic features able to capture the relevant musical attributes identified in the literature, namely features better correlated to valence.

As stated in previous studies [10], the lyrical part of a song can have a great influence in the transmitted mood. The emotional response to the lyrics, obtained through natural language processing and commonsense reasoning, contributes to both the context and mood classification of the song [25].

As for playlist creation, it would be interesting to add some constraints regarding song ordering, for example, in terms of balance and progression.

Acknowledgments

This work was supported by the MOODetector project (PTDC/EIA-EIA/102185/2008), financed by the Fundação para Ciência e Tecnologia - Portugal.

6. REFERENCES

- [1] T. Fritz et al., “Universal Recognition of Three Basic Emotions in Music,” *Current Biology*, vol. 19, no. 7, pp. 573-6, Apr. 2009.
- [2] K. Hevner, “Experimental Studies of the Elements of Expression in Music,” *American Journal of Psychology*, vol. 48, no. 2, pp. 246-268, 1936.

- [3] M. Zentner, D. Grandjean, and K. R. Scherer, "Emotions Evoked by the Sound of Music: Characterization, Classification and Measurement," *Emotion*, vol. 8, no. 4, pp. 494-521, Aug. 2008.
- [4] J. S. Downie, "2010: MIREX2010 Results," 2010. [Online]. Available: http://www.music-ir.org/mirex/wiki/2010:MIREX2010_Results. [Accessed: 19-May-2011].
- [5] Y.-H. Yang, Y.-C. Lin, Y.-F. Su, and H. H. Chen, "A Regression Approach to Music Emotion Recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 448-457, Feb. 2008.
- [6] R. E. Thayer, *The Biopsychology of Mood and Arousal*. Oxford University Press, USA, 1989, p. 256.
- [7] S. L. Chiu, "Selecting input variables for fuzzy models," *Journal of Intelligent and Fuzzy Systems*, vol. 4, no. 4, pp. 243-256, 1996.
- [8] M. Robnik-Šikonja and I. Kononenko, "Theoretical and Empirical Analysis of ReliefF and RReliefF," *Machine Learning*, vol. 53, no. 1-2, pp. 23-69, 2003.
- [9] M. P. H. Vossen, "Local Search for Automatic Playlist Generation," M.S. thesis, Technische Universiteit Eindhoven, 2005.
- [10] Y. E. Kim et al., "Music Emotion Recognition: A State of the Art Review," in *Proc. 11th Int. Society for Music Information Retrieval Conf.*, 2010, pp. 255-266.
- [11] A. Gabrielsson and E. Lindström, "The Influence of Musical Structure on Emotional Expression," in *Music and Emotion*, vol. 8, Oxford University Press, 2001, pp. 223-248.
- [12] A. Friberg, "Digital Audio Emotions - An Overview of Computer Analysis and Synthesis of Emotional Expression in Music," in *Proc. 11th Int. Conf. on Digital Audio Effects*, 2008, pp. 1-6.
- [13] L. Lu, D. Liu, and H.-J. Zhang, "Automatic Mood Detection and Tracking of Music Audio Signals," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 5-18, Jan. 2006.
- [14] Y. Feng, Y. Zhuang, and Y. Pan, "Popular Music Retrieval by Detecting Mood," *Proc. 26th Annu. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, vol. 2, no. 2, p. 375, 2003.
- [15] D. Yang and W. Lee, "Disambiguating Music Emotion Using Software Agents," in *Proc. 5th Int. Conf. on Music Information Retrieval*, 2004, p. 52-58.
- [16] D. Liu and L. Lu, "Automatic Mood Detection from Acoustic Music Data," *Int. J. on the Biology of Stress*, vol. 8, no. 6, pp. 359-377, 2003.
- [17] B. Logan, "Music Recommendation from Song Sets," in *Proc. 5th Int. Conf. on Music Information Retrieval*, 2004, pp. 425-428.
- [18] E. Pampalk, T. Pohle, and G. Widmer, "Dynamic Playlist Generation Based on Skipping Behavior," in *Proc. 6th Int. Conf. on Music Information Retrieval*, 2005, pp. 634-637.
- [19] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer, "Playlist Generation Using Start and End Songs," in *Proc. 9th Int. Conf. of Music Information Retrieval*, 2008, pp. 173-178.
- [20] J. J. Aucouturier and F. Pachet, "Scaling Up Music Playlist Generation," in *Proc. 2002 IEEE Int. Conf. Multimedia and Expo*, 2002, vol. 1, p. 105-108.
- [21] S. Pauws, W. Verhaegh, and M. Vossen, "Fast Generation of Optimal Music Playlists Using Local Search," in *Proc. 6th Int. Conf. on Music Information Retrieval*, 2006, pp. 138-143.
- [22] J.-J. Aucouturier and F. Pachet, "Finding Songs that Sound the Same," in *Proc. IEEE Benelux Workshop on Model-Based Processing and Coding of Audio*, 2002, pp. 91-98.
- [23] C. Laurier, M. Sordo, J. Serrà, and P. Herrera, "Music Mood Representations from Social Tags," in *Proc. 10th Int. Society for Music Information Conf.*, 2009, pp. 381-386.
- [24] Z. Cataltepe, Y. Tsuchihashi, and H. Katayose, "Music Genre Classification Using MIDI and Audio Features," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, pp. 275-279, 2007.
- [25] O. C. Meyers, "A mood-based music classification and exploration system," M.S. thesis, Massachusetts Institute of Technology, 2007.
- [26] E. Schubert, "Measurement and Time Series Analysis of Emotion in Music," *Emotion*, vol. 1, 1999.
- [27] O. Lartillot and P. Toiviainen, "A Matlab Toolbox for Musical Feature Extraction from Audio," in *Proc. 10th Int. Conf. on Digital Audio Effects*, 2007, p. 237-244.
- [28] A. Meng, P. Ahrendt, J. Larsen, and L. K. Hansen, "Temporal Feature Integration for Music Genre Classification," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, no. 5, pp. 275-9, 2007.
- [29] A. K. Sen and M. S. Srivastava, *Regression Analysis: Theory, Methods and Applications*. Springer, 1990, p. 362.
- [30] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *Computer*, pp. 1-30, 2001.
- [31] C. M. Bishop, "Neural Networks for Pattern Recognition," *Journal of the American Statistical Association*, vol. 92, no. 440, p. 1642, 1995.
- [32] I. Nabney and C. Bishop, "Netlab Neural Network Software," *Pattern Recognition*. 1997.

TOWARDS A PERSONALIZED TECHNICAL EAR TRAINING PROGRAM: AN INVESTIGATION OF THE EFFECT OF ADAPTIVE FEEDBACK

Teruaki Kaniwa¹, Sungyoung Kim², Hiroko Terasawa¹, Masahiro Ikeda²,
Takeshi Yamada¹, Shoji Makino¹

¹University of Tsukuba, ²Yamaha Corporation

kaniwa@mmlab.cs.tsukuba.ac.jp, sungyoung@beat.yamaha.co.jp,
terasawa@tara.tsukuba.ac.jp, masahiro_ikeda@gmx.yamaha.com,
takeshi@cs.tsukuba.ac.jp, maki@tara.tsukuba.ac.jp

ABSTRACT

Technical ear training aims to improve the listening of sound engineers so that they can skillfully modify and edit the structure of sound. To provide non-professionals such as amateur sound engineers and students with this technical ear training, we have developed a simple yet personalized ear training program. The most distinct feature of this system is that it adaptively controls the training task based on the trainee's previous performance. In detail, this system estimates a trainee's weakness, and generates a training routine that provides drills focusing on the weakness, so that the trainee can effectively receive technical ear training without an instructor. We subsequently investigated the effect of the new training program with a one-month training experiment involving eight subjects. The result showed that the score of the group assigned to the proposed training system improved more than that of the group assigned to conventional training.

1. INTRODUCTION

Technical ear training aims to improve the ability to systematically discriminate and identify sonic differences. In the audio and music production/reproduction business, this ability has been regarded as essential for sound processing "in order to create the desired quality of sound" as Miskiewicz suggests [1]. Since Retowski's initial work [2], many institutions have developed systematic training programs for their junior employees or students [1-8].

Although the specific purposes of training programs vary according to the educational goal, the fundamental training method is to have a trainee compare a reference signal with its sonically modified version, comprehend the difference, and then repeat such comparisons until the trainee can reliably identify the sonic difference without a reference. In most cases, an instructor interactively guides the trainee; as in a music lesson, the trainee practices this method by observing the instructor's demon-

stration, accomplishing the given task, and receiving guidance or feedback to allow progress to the next training stage. In contrast, this method may be followed independently with existing static training materials such as the Ear-training audio CD [3].

Recent ear training programs [4-8] often utilize computer software that assists the instructor in the teaching task by, for example, administering the training schedule, and displaying the training scores to the trainee through the real-time visualization of electronically stored data. By incorporating these additional features from computer-assisted training programs, an instructor can effectively guide a trainee and offer appropriate advice to make the training faster than purely empirical acquisition.

The next step for the ear-training software that will take it beyond being a mere teaching aid is to guide the trainee in the same manner as a private lesson, so that a trainee can practice the training without an instructor. The related research field to this project is Intelligent Tutoring Systems(ITS), in which artificial intelligence systems provide customized instructions or discussions to students, i.e. without the intervention of human beings [9]. Topics of ITS includes to select problems at a level of difficulty appropriate to the student's overall performance, and such systems have been called "adaptive" and their sophistication lay in the task-selection algorithms. Moreover, many conferences of ITS have been held internationally, e.g. ITS2010 in Pittsburgh [10]. In recent e-learning studies, this adaptive and interactive feature is regarded as essential for maintaining a trainee's motivation even in a self-learning scenario [11].

In this study, we propose a new system for a computer-based ear training program that adaptively creates personal training routines based on an individual's training record. This system estimates the trainee's weakness, and generates a training routine, which provides drills focusing on the weakness, so that the trainee can effectively study technical listening without an instructor. We conducted a series of evaluation tests, and report the effect of this training system.

Copyright: © 2011 Kaniwa et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. EAR TRAINING PROGRAM

As mentioned in Section 1, various types of ear training program have been proposed. The research group at McGill University in Canada proposed three types of training namely, “matching”, “removing”, and “absolute identification.”[5]. All three types adopt the training method introduced in the previous section and control the spectrum of a sound using the three parameters of a parametric equalizer [12] (center frequency, Q, and gain). Matching and removing training involves asking a trainee to modify the spectrum of a sound and make it equal to the given modification. On the other hand, the goal of an absolute identification task is to increase a trainee’s ability to identify a modified spectrum, describe it in terms of technical parameters (center frequency, Q, and gain), and eventually build a long-term memory of the internal reference on which a trainee will rely for future identification tasks.

Of these three types of training, we employed “absolute identification” for this study because it enables entry-level listeners to learn the identification quickly in a limited training period.

During our informal preliminary study, we found that the inherent ability to identify the modified center frequencies was not identical for all trainees. For example, a trainee may find it difficult to discriminate subtle differences between two adjacent low frequency bands such as 250 and 125 Hz, while another trainee shows similar confusion in the high frequency area. To the best of our understanding, the conventional training programs for self-study do not control such individual differences, which could make it more difficult for a trainee to overcome his/her weak points. Therefore, we hypothesized that a new system would assist a trainee more effectively if it could analyze a trainee’s previous training record and adaptively provide more training in the area in which he or she performed poorly. And we created a computer-based, personalized technical ear-training program, which manages individual training records with a database and provides personal training routines adaptively based on the record. Figure 1 shows a block chart of the new training system.

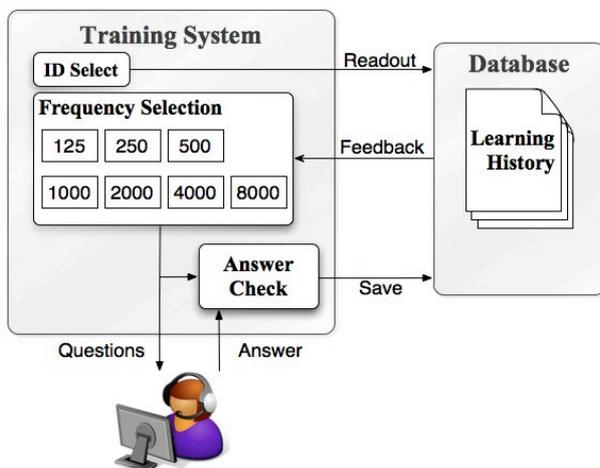


Figure 1. Block chart of new training system

3. GENERATION OF PERSONALIZED PROBLEM SET

As stated above, we proposed a new system that varies the composition of training questions depending on the individual study record. In our proposed system, the software reads the previous correct answer rate of the current trainee (i.e. how precisely the trainee identified the modified spectrum), and adjusts the probability of question appearance. The new probability of question appearance is calculated by dividing the relative weight of each frequency bandwidth by the entire weight as follows:

$$P_n = \frac{W_n}{W} \quad (1)$$

where n is the index number of the filter in this study as shown in Table 1, P_n is the probability of question appearance for bandwidth n , W_n is the relative weight of bandwidth n . W is the entire weight denoted as follows:

$$W = \sum_{n=1}^7 W_n \quad (2)$$

W_n is calculated as follows:

$$W_n = 100 - \frac{100 - L}{100} \cdot R_n \quad (3)$$

where L is the minimum weight, and R_n is the correct answer rate for bandwidth n . W_n decreases monotonously as R_n increases. To confirm whether a trainee has achieved a high correct rate by chance, the program needs to test even for a high-score bandwidth by assigning a non-zero value to L . Currently L is set at 25 after several heuristic trials, resulting in a W_n value range of 25 to 100. With this weighting process, the system generates more questions to the low-score bandwidth, and fewer to the high-score bandwidth. In addition, this process is dynamic so that if the correct answer rate changes, the system automatically adjusts the question appearance accordingly. The consequent question was whether or not this “interactive and smart randomization” would provide self-trainees with more effective learning. To investigate this question, we conducted an experiment that compared the influence of the proposed system on learning the absolute identification of spectral modification.

| N | Center frequency [Hz] |
|-----|-----------------------|
| 1 | 125 |
| 2 | 250 |
| 3 | 500 |
| 4 | 1000 |
| 5 | 2000 |
| 6 | 4000 |
| 7 | 8000 |

Table 1. Center frequency of the filter

4. EXPERIMENT TO EVALUATE PROPOSED SYSTEM

To evaluate the proposed system, we formed two groups of trainees: the **Conventional group** and the **Proposal group**. For the **Conventional group**, we set the software to generate questions using a non-weighted random function, while for the **Proposal group**, it generated questions using the weighted random function dynamically updated according to the trainee’s previous training scores as described in Section 3.

In total, eight subjects participated in the experiment. Before the main experiment, we conducted a preliminary test of absolute identification (25 questions with +12 dB boosted pink noise) and divided the subjects into two groups so that the initial condition of each group was as close as possible. As shown in Table 2, the descriptive statistics of the preliminary test scores indicated that these two groups were similar to each other. Furthermore, we conducted a preliminary F-test to test the equal variance, followed by a two-tailed t-test (two independent samples with equal sample size and equal variance), which confirmed that there were no statistically significant differences between the two groups as shown in Tables 3 and 4.

| Group | Conventional | Proposal |
|--------------|--------------|----------|
| <i>M</i> [%] | 64 | 63 |
| <i>SD</i> | 4.89 | 5.19 |

Table 2. Grouping of subjects

| | |
|-----------------|------|
| <i>F</i> (1, 6) | 47.5 |
| <i>p</i> | 0.39 |

Table 3. F-test of initial test (Conventional group vs. Proposal group)

| | |
|--------------|------|
| <i>t</i> (6) | 2.45 |
| <i>p</i> | 0.63 |

Table 4. t-test of initial test (Conventional group vs. Proposal group)

The experiment took place in a recording studio at the University of Tsukuba using three types of sound files namely, “Pink Noise”, “Orchestra (Symphony No. 3 by C. Saint-Saens)”, and “Piano (Piano Sonata #2, Op. 36 by S. Rachmaninov)”. We selected a portion of each sound file that contained the all target bands required for the training. Table 5 shows the equipment used for the experiment.

After an initial tutorial session on the operation of the ear training program, the subjects practiced with the training system by themselves twice a week for about 30 minutes each time, according to the training curriculum shown in Table 6. The training curriculum is designed so that the subjects find the task challenging and motivating with a level of difficulty that increases as the training proceeds. The total duration of the training was four weeks. In addition to actual training, the subjects could freely practice with the content shown in Table 7 on the first training

day of each week to familiarize themselves with the kind of training task they would perform in that week.

| | |
|----------------------|--------------------|
| Audio Interface | MOTU UltraLite-mk3 |
| Headphone | SENNHEISER HD 650 |
| Programming Language | Max MSP ver.5.1.5 |

Table 5. Equipment used for the experiment

| Week | dB | Sound File |
|------|------------------|------------------------------|
| 1 | +12 dB, +6 dB | Pink Noise, Orchestra, Piano |
| 2 | +6 dB, +3dB | Pink Noise, Orchestra, Piano |
| 3 | − 12 dB | Pink Noise, Orchestra |
| 4 | ± 12 dB, − 12 dB | Pink Noise, Orchestra |

Table 6. Training curriculum

| Week | dB | Sound File |
|------|---------|------------|
| 1 | +12 dB | Pink Noise |
| 2 | +6 dB | Pink Noise |
| 3 | − 12 dB | Pink Noise |
| 4 | ± 12 dB | Pink Noise |

Table 7. Contents of practice

5. RESULTS

5.1 Analysis by Mean and Standard Deviation

We first analyzed the mean value and the standard deviation of the correct answer rate to investigate the effect of the proposed system. Figure 2 shows the overall average correct answer rate for each sound. From this, it was found that the average correct answer rate of the **Proposal group** was higher than that of the **Conventional group**. This result suggests that the proposed system raises the correct answer rate based on the fact that the initial average (before training) was about the same for the two groups.

Next, we analyzed the results for each sound. Figures 3, 4 and 5 show the mean value and the standard deviation of the correct answer rate for Pink Noise, Orchestra and Piano respectively. The analysis results show that the mean correct answer rate of the **Proposal group** is consistently higher than that of the **Conventional group** for all sounds. After verifying the equal variance with a preliminary F-test, we conducted a one-tailed t-test (independent two-sample, with equal sample size and equal variance) to evaluate the effectiveness of the **Proposal group** compared with that of the **Conventional group**. Table 8 shows the results of the t-test for each sound. We found a significant difference with Orchestra and Piano sounds but not with Pink Noise. This might be because the temporal variation of musical sounds (Orchestra and Piano) made it harder for a trainee to identify the spectral modification than Pink Noise. This result suggests that the proposed system is more efficient in terms of training for realistic and difficult tasks than the conventional training system.

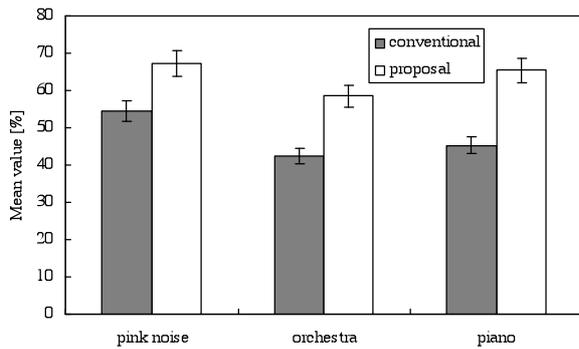


Figure 2. Average of each sound

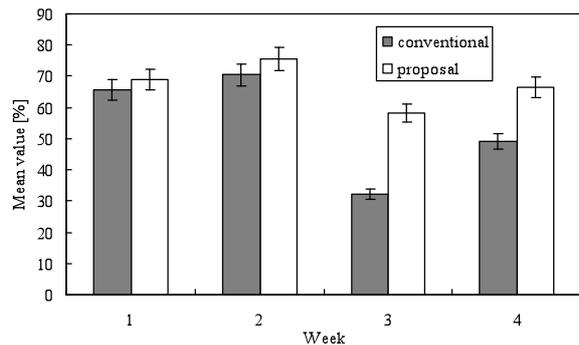


Figure 3. Result for Pink Noise

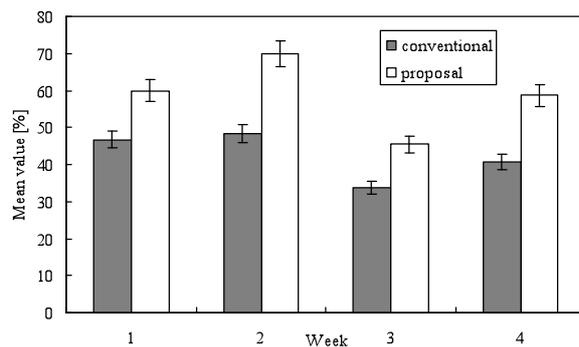


Figure 4. Result for Orchestra

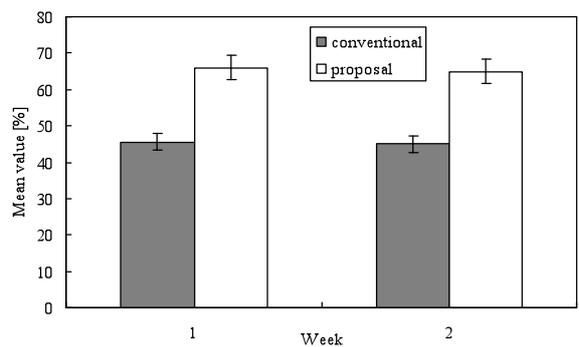


Figure 5. Result for Piano

| Sound | Pink Noise | Orchestra | Piano |
|---------------------|------------|-----------|---------|
| <i>df</i> | 6 | 6 | 2 |
| <i>t</i> (absolute) | 1.36 | 2.67 | 36.2 |
| <i>p</i> | 0.11 | 0.018 | 0.00038 |

Table 8. t-test of training data

5.2 Analysis of Improvement for Low-Score Bands

In addition to the analysis in Section 5.1, we investigated the improvement for the low-score band, to confirm that use of the proposed system increased the correct answer rate in low-score bands as intended.

In this analysis, we counted the direction of change (i.e. positive, negative, and unchanged) in the correct answer rate in low-score bands, and compared the counts of the Proposal group and Conventional group. For each subject's data, we extracted three low-score bands that marked the lowest correct answer rates for every week, and analyzed the way in which the correct answer rates in those bands changed in the following week. The indices of change were “+ (improvement)”, “- (decrease)”, and “0 (no change)”. Figure 6 shows the frequency distribution of the score change directions. The “-” element was smaller and the “+” element was larger for the Proposal group than for the Conventional group. To compare the frequencies of the score change direction quantitatively, we counted the sum of the score change by substituting each “-” element with “-1”, each “+” element with “+1”, and the no-change element with “0”. The sum of the score change for the Proposal group was -3 whereas that for the Conventional group was -13, revealing greater improvements for the low-score band with the proposed system. (This sum naturally tends to be negative because the training was designed to become harder.)

Figure 7 shows the frequency distributions for the difference in the correct answer rates for the low-score bands. The Proposal group showed a higher frequency of positive score difference between 0% and 30% than the Conventional group. In contrast, the Proposal group showed a lower frequency of negative score difference between -20% and -10%. In other words, the Proposal group showed a strong tendency to improve their listening in low-score bands. This means that the Proposal group exhibited better progress in the low-score bands, which indicates that they conducted self-training more efficiently than the Conventional group.

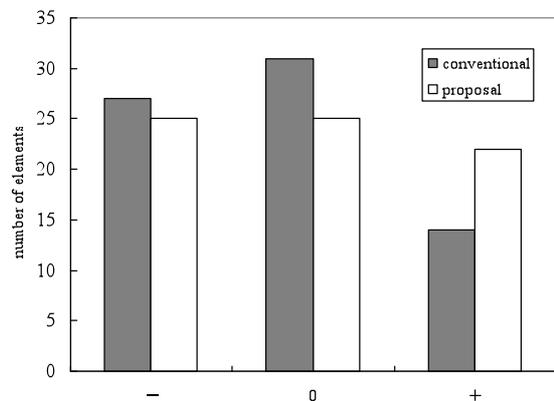


Figure 6. Result of analyzing with the index of the changed direction of the correct answer rate

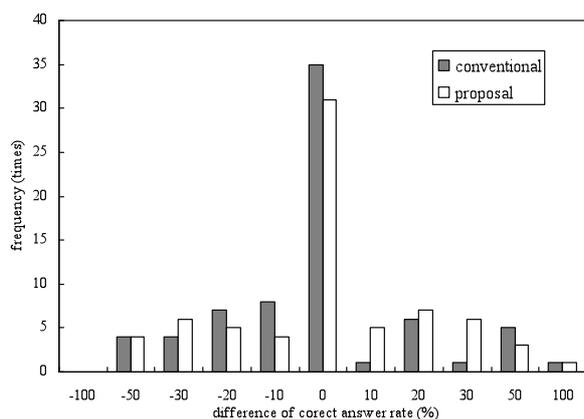


Figure 7. Frequency distribution

6. CONCLUSIONS

To support self-learners who are training their technical listening ability, we proposed a new training program that adapts to individual differences and provides more training with regard to a trainee’s weak points. However, the proposed system exhibited different levels of effectiveness for each type of sounds, and that suggests interactions between the types of the test-sounds. In our future work, it would be desirable to reconsider the experimental design, so that analysis of variance can be conducted to test the effect of interactions. Even though this kind of software will not replace a human instructor, the results from the experimental test showed that our proposed system could assist the trainees to conduct more effective training by themselves, especially with realistic, therefore more difficult, identification tasks. To make this program more interactive and educative, we are investigating additional features that could offer a self-trainee fast and entertaining training, such as implementing the training program in a portable device.

7. REFERENCES

[1] A. Miskiewicz, “Timbre Solfège: A Course in Technical Listening for Sound Engineers,” in *J. Audio Eng. Soc.*, 1922, 40(7/8): pp. 621 – 625.

[2] T. Letowski, “Development of technical listening skills: Timbre solfeggio,” in *J. Audio Eng. Soc.*, 1985, 33(4): pp. 240 – 244.

[3] D. Moulton, *The Golden Ears Audio Eartraining Program*. Audio CD published by KIQ Productions.

[4] S. Iwamiya, Y. Nakajima, K. Ueda, K. Kawahara, and M. Takada, “Technical Listening Training: Improvement of sound sensitivity for acoustic engineers and sound designers,” in *Acoustical Science and Technology*, January 2003, 24(1): pp.27 – 31.

[5] R. Quesnel, “Timbral Ear Trainer: Adaptive, interactive training of listening skills for evaluation of timbre,” in *Proc. Audio Engineering Society 100th*

Int. Conv., Copenhagen, Denmark, 1996. AES. Preprint 4241.

[6] S. E. Olive, “A new listener training software application,” in *Proc. Audio Engineering Society 110th Int. Conv.*, Amsterdam, Netherlands, May 2001, AES. Preprint 5384.

[7] J. Corey, *Audio Production and Critical Listening*. Focal Press, 2010.

[8] N. Akira, “Auditory training system that uses TCP/IP networks and WWW browsers,” in *Japanese Acoustical Society of Japan (J)*, 2006, 62(3): pp. 208 – 213.

[9] D. Sleeman and J. S. Brown, *Intelligent Tutoring Systems*. Academic Press, 1982.

[10] V. Aleven, J. Kay, and J. Mostow, *Intelligent Tutoring Systems 10th International Conference, ITS 2010*, Pittsburgh, PA, USA Springer-Verlag, 2010.

[11] R. Garris, R. Ahlers, and J. E. Driskell, “Games, motivation, and learning: A research and practice model,” in *Simulation & Gaming*, 2002, 33(4): pp. 441 – 467.

[12] G. Massenburg, “PARAMETRIC EQUALIZATION,” in *Proc. Audio Engineering Society 42nd Int. Conv.*, Los Angeles, USA, May 1972. AES.

EXTRACTION OF SOUND LOCALIZATION CUE UTILIZING PITCH CUE FOR MODELLING AUDITORY SYSTEM

Takatoshi Okuno, Thomas M. McGinnity, Liam P. Maguire

Intelligent Systems Research Centre, University of Ulster, Derry, BT48 7JL UK.

t.okuno@ulster.ac.uk

ABSTRACT

This paper presents a simple model for the extraction of a sound localization cue utilizing pitch cues in the auditory system. In particular, the extraction of the interaural time difference (ITD) as the azimuth localization cue, rather than the interaural intensity difference (IID), is constructed using a conventional signal processing scheme. The new configuration in this model is motivated by psychoacoustical and physiological findings, suggesting that the ITD can be controlled by the pitch cue in the simultaneous grouping of auditory cues. The localization cues are extracted at the superior olivary complex (SOC) while the pitch cue may be extracted at a higher stage of the auditory pathway. To explore this idea in the extraction of ITD, a system is introduced to feed back information on the pitch cue to control and/or modify the ITD for each frequency channel.

1. INTRODUCTION

Computational modelling of the auditory system has been recently investigated at the neuronal level. In particular, in a model for the extraction of auditory cues related to sound localization, such as ITD and IID, spiking neural networks (SNN) are utilized [1]. Such modelling can be established by defining which part of the auditory pathway functions to process each auditory cue. However, it is unclear where other cues such as pitch are processed in the auditory pathway. Wrigley et al. [2] states that the neurophysiological mechanisms underlying auditory stream formation are poorly understood and it is not fully known how groups of features are coded and communicate within the auditory system.

In physiological studies, it is understood that the auditory cues for sound localization are extracted at the medial superior olive (MSO) and lateral superior olive (LSO) in the SOC, then integrated at the inferior colliculus (IC) to extract representations of positions in space [3]. However, the extraction process of *pitch*, which is recognized as one of the most primitive cues among the auditory cues, is not well identified. According to some recent papers, the extraction of pitch may be processed at the IC by the existence of some neurons responding to the sinusoidally am-

plitude modulated sound within a restricted range [3]; or it may be processed at the SOC by the existence of Huggins pitch known as a result of the binaural interaction of noise stimuli [4], [5]; or that there may exist an extraction process of pitch at the brainstem and thalamus. The decision process of pitch may occur at lateral Heschl's gyrus in the auditory cortex through the analysis of fMRI [6]. Therefore, it is not currently possible to identify exactly which part of the auditory pathway has a particular role for extracting pitch. Assuming that the decision process of pitch ends at the auditory cortex, it may be possible to have the decision process of pitch performed after the extraction process of the sound localization cues.

In psychoacoustical studies, there have been extensive findings about sound localization, pitch and other cues, that have been summarized in the research framework referred to as auditory scene analysis (ASA) [7]. Treating ASA with a computational approach (computational ASA: CASA) to resolve certain engineering problems such as signal separation issues has enabled many computational models for ASA systems to be undertaken [8].

Recently, sound localization cues were used for sequential organization. This means that auditory objects from the same spatial direction can be organized as one auditory stream, even if those auditory objects are isolated from each other in terms of time, although the sound localization cues have been regarded as one of the primitive cues for simultaneous organization in ASA [9, 10]. Darwin [11] states that ITDs are remarkably ineffective at segregating simultaneous sounds despite the dominance of ITDs in the region around 500 Hz [12]. Culling also mentioned that harmonicity contributes to the grouping of sounds across the frequency integration of ITD, according to experimental results by Hill et al. [13]. Furthermore, the relationship between ITD and pitch indicated that the formation of auditory objects precedes decisions on their location so that a model would allow pooling of location information across frequency channels in order to reduce the variability found in individual channels and so produce a percept with a stable location.

This paper proposes a simple model using the ITD and pitch cues, that considers the interaction between the two cues while they are being extracted. This is not a conventional approach in CASA, that permits the individual extraction of auditory cues independently [8]. Considering the contradiction between the physiological view (the extraction of sound localization cues may precede the decision process of pitch) and psychoacoustical view (the for-

mation of auditory objects including the use of pitch may precede the decision of sound localization), the model is proposed as a feedback system with the extraction of ITD before that of pitch so that the model can be biologically-plausible. The proposed model differs from the frame based method [10] in that it concerns the order of the process as a feedback system and the frequency dependence of ITDs. The model is constructed at the level of conventional signal processing, incorporating the use of an auditory periphery model and a correlation based calculation as this will allow the model to be reconstructed by an SNN in the future in terms of biological plausibility.

2. A PROPOSED MODEL

The proposed model is described by the systematic configuration shown in Fig.1 and each calculation method is explained in turn as follows. The signals presented here are speech signals and white noise for simplicity and quantification. The ratio between the levels of the two signals is controlled by the signal to noise ratio (SNR) at the origin of the signals. By convolving each signal with head-related transfer functions (HRTFs), it can convey the information of sound location. The signals for the left and right ears are added to yield the binaural signal. HRTFs utilized here are produced from the MIT media lab, they are bilaterally symmetric measurement data sets using a dummy head with the same size pinnae for both ears in an anechoic chamber [14, 15]. Since all HRTFs are prepared at 44.1 kHz sampling frequency, computer simulation performed later on is undertaken using the same sampling frequency. For the directions of sound, a range of $\pm 90^\circ$ azimuth with the midline as the centre (5° intervals) is considered.

Each binaural signal is decomposed into frequency channels by applying a Gammatone filter bank which models the filtering at the cochlea [16]. The frequency decomposition by the cochlear filtering is performed in 64 channels covering 50-8000 Hz, which should be a sufficient number of channels in terms of the equivalent rectangular bandwidth (ERB) rate [17]. The frequency range covered by the filter bank should be appropriate even though ITD and pitch frequencies processed at a later stage are taken into account. The output of the filter bank is used as the input to calculate the summary cross correlation function (SCCF) in order to obtain the ITD. From this stage, the calculation is processed on a frame by frame basis. If SCCF is calculated in the range of the lag time between ± 1 ms, it covers the range of the azimuth between $\pm 90^\circ$. However, one frame length is set as 30 ms here because of the stability of the pitch extraction algorithm, which is performed at a later stage.

Different ways to obtain the signal to be used for the pitch extraction from the binaural signal are proposed in the literature. In [9], the signal used for the pitch extraction is named as the *better ear signal* which has a better SNR determined from the signal before adding the signals convolved with the HRTFs. In [10], a signal produced by averaging the left and right ear signals are used as the better ear signal. Considering the head-shadow effect [18] and the diffraction wave, however, averaging the left and right

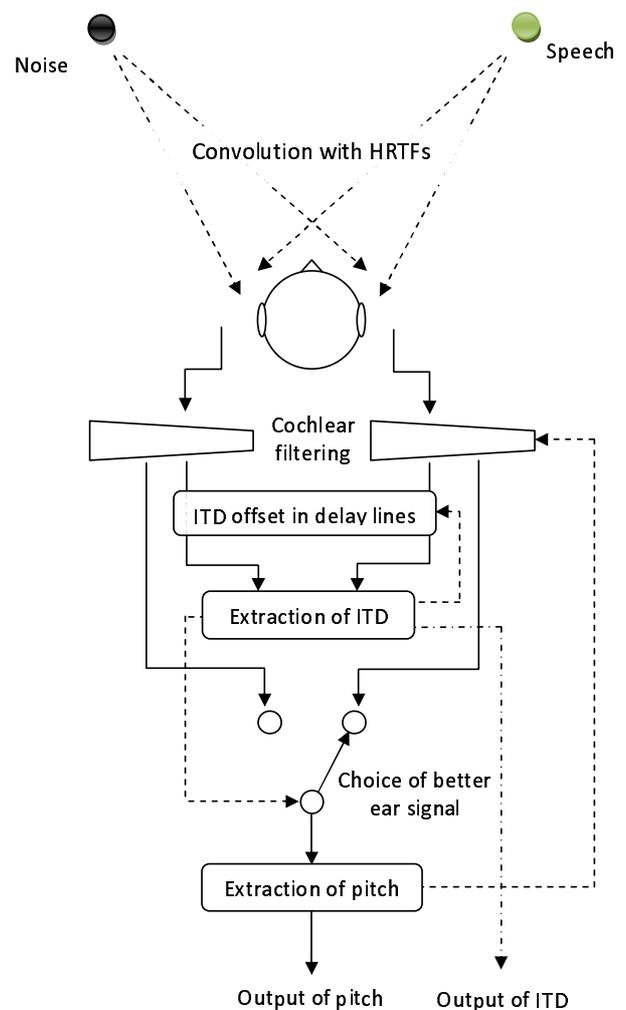


Figure 1. A systematic configuration for a proposed model

ear signals would make the better ear signal complicated. Therefore, in each frame, the better ear signal is obtained by choosing the left or right ear signal, which is based on the value of ITD such as the left for the negative value of ITD or the right for the positive value of ITD, calculated at the earlier stage. This function is depicted by a symbolic switch in Fig.1 and denoted as "Choice of better ear signal". This assumes that the ITD extracted is calculated for a dominant signal in the binaural signal or for the second dominant signal after the dominant signal is removed.

The pitch extraction algorithm is implemented by the minor updated algorithm of enhanced summary auto correlation function (ESACF), based on the summary auto correlation function (SACF) [19]. Practically, the better ear signal obtained in the frequency channels is half-rectified and used to calculate the auto correlation functions, which are then summarized over the frequency channels. Considering pitch extraction under noisy conditions, the minor update is conducted by subtracting the mean value from the SACF, then re-performing half-rectification in order to make the sensitivity of the pitch extraction higher. In ESACF, the minor updated SACF is interpolated on the basis of harmonics, then the signal is subtracted from the

minor updated SACF and half-rectified again. This process is repeated to estimate the fundamental frequency (F_0) within a frame. This interpolation is especially performed by considering the second, third and fifth harmonics. The summation over frequencies in SACF is calculated for 25 channels from 99 Hz to 961 Hz, which corresponds to the centre frequencies of the Gammatone filter bank. However, the limitations for the possible frequency range of the F_0 s are defined from 100 Hz up to 450 Hz because the pitch extraction is basically designed for a speech signal. In addition, to avoid the mis-extraction during silent intervals of speech (when clean speech is the only input), the pitch extraction algorithm can be turned on or off with the estimation of the power of the signal such as comparing the lag-zero values of SACF with a certain threshold.

The accuracy of the estimated F_0 s is dependent on the capability of the algorithm and the characteristics of the input signals. This means that it is not always possible to extract the true F_0 s. Therefore, the *harmonic stream* which is composed of a F_0 up to third harmonics, is defined by classifying the estimated F_0 s. In the n -th frame, a candidate of F_0 extracted by the ESACF algorithm is defined as $F_0(n)$, and three frequencies of the harmonic stream are defined as $f_0(n)$, $f_1(n)$ and $f_2(n)$ respectively. Then the classification is performed by the following three equations:

$$\text{if } F_0(n) < 200, \begin{cases} f_0(n) = 2 \times F_0(n) \\ f_1(n) = 3 \times F_0(n) \\ f_2(n) = 0 \end{cases} \quad (1)$$

$$\text{if } 200 \leq F_0(n) \leq 350, \begin{cases} f_0(n) = F_0(n) \\ f_1(n) = 2 \times F_0(n) \\ f_2(n) = 3 \times F_0(n) \end{cases} \quad (2)$$

$$\text{if } F_0(n) > 350, \begin{cases} f_0(n) = 2 \times 0.5 \times F_0(n) \\ f_1(n) = 3 \times 0.5 \times F_0(n) \\ f_2(n) = 0 \end{cases} \quad (3)$$

Owing to the limitation of Eq.(1), the $F_0(n)$ from 100 to 200 Hz, which is generally said to be the F_0 of speech, is not used for any of the following processes. This is because the ITDs for the frequencies lower than 200 Hz are not reliable due to the phase analysis of HRTFs between left and right ears. In addition, since the frequencies higher than 300 Hz are dominant for the pitch sensation [20], it would be reasonable to construct the harmonic stream higher than 200 Hz. $f_2(n)$ in Eq.(1) and (3) is set to zero since the 4th harmonics of the expected F_0 is not used to construct the harmonic stream here.

The harmonic stream is then replaced to the nearest centre frequencies of the Gammatone filter bank, and the chosen frequency channels are utilized to calculate the ITD at the next frame. However, due to the effect of the diffraction wave around the dummy head in the measurement of HRTFs, it is known that ITD and IID are changed depending on the frequencies, the direction of sound source and the physical size of the dummy head [21]. This is because there is an object between two microphones (like a dummy head) which is not negligible when the human head is considered. Especially, in the sampling frequency that just covers the audible range such as 44.1 kHz, a small num-

ber of sample difference would affect the accuracy in estimation of the sound direction. In [18], the fact that their proposed system does not work due to the diffraction wave in the case where there is an object between two microphones in their 2ch microphone array system, is discussed. They indicate that it can be interpreted as filtering with the transfer function characteristic of the diffraction. This filter can be also incorporated into the delay units of the dual delay line. Here, before the summation over frequencies in SCCF, the offset of ITD is applied to correct the estimated angles or ITDs. It is proposed that a matrix of the offset as a function of the sound direction and frequency is prepared, and the offset selected by the estimated angle in the previous frame is applied. With the sampling frequency f_s , the angular frequency ω , the phase difference of HRTFs between left and right ear $\varphi_\theta(\omega)$ at the measured angle θ of HRTFs, the offset is calculated as the difference between the phase delay (in samples) in Eq.(4) and the sample difference for the whole frequency range obtained by the cross correlation of HRTFs:

$$\text{Phase delay } (\theta) := \frac{-f_s \varphi_\theta(\omega)}{\omega} \quad (\text{in samples}) \quad (4)$$

Fig.2 shows the offsets as a function of the centre frequencies of Gammatone filter bank, for $\theta = 30^\circ, 60^\circ, 90^\circ$. These values are rounded to the nearest integer for the ITD offset axis. It is noted that the offset is applied even if the

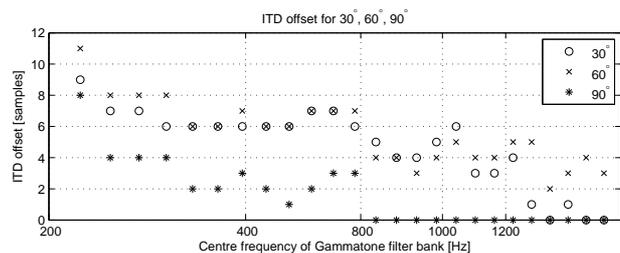


Figure 2. Examples of the offset in case of $\theta = 30^\circ, 60^\circ, 90^\circ$

estimated angle is wrong or changed significantly from the previous frame since this offset is utilized based on the results of the estimation in the previous frame. Therefore, for example, if the sound direction changes from -60° to $+60^\circ$ drastically, the offset gives the error to the estimates of $+60^\circ$ since the offset is applied based on the estimates for -60° .

Using the geometric approximation often used for a 2ch microphone array system, the estimated ITDs in the samples can be converted to azimuthal angles. However, this conversion cannot keep the linearity for angles close to $\pm 90^\circ$ because of the diffraction of the dummy head. Therefore, by calculating the cross correlation function of HRTFs between both ears for all azimuth, and then interpolating the obtained ITDs linearly, the angles when HRTFs are measured are linked with estimated ITDs in samples. Since the azimuth between -90° and 0° is symmetric to the azimuth between 0° and 90° , the relationship between the estimated ITDs and the angles of HRTFs is shown in Fig.3.

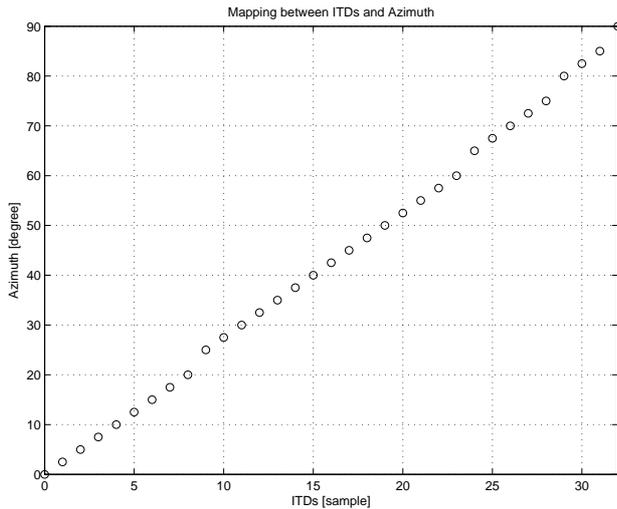


Figure 3. The relationship between the estimated ITDs and the azimuth

3. SIMULATION RESULTS

To examine the performance of the proposed system, a computer simulation was conducted for a speech signal and a directional white noise. Five seconds of female speech sample was utilized and convolved with the HRTF for the 30° angle. Similarly, white noise was generated and convolved with the HRTF for -30° , which becomes a directional white noise. Both signals were added in order to generate the binaural signal. The right ear signal, left ear signal and better ear signal are shown in Fig.4(a), (b) and (c) respectively. At the location of sound sources, the SNR is controlled as 10 dB. Comparing (a) with (b), it appears there is a difference in SNR because of the head-shadow effect. The solid and dotted lines in Fig.4(c) are the results of "Choice of better ear signal" from both (a) and (b). Namely, when there is a solid line with a value $+0.5$, which means the intervals of the positive value of ITD, which indicates that the right ear signal is used. Conversely, when there is a dotted line with a value -0.5 , this means the intervals of the negative value of ITD, which indicates that the left ear signal is used. Therefore, (c) is a combination of the signals from both (a) and (b).

As mentioned before, a modified ESACF method is employed for pitch extraction. The extraction results are shown in Fig.5 with circles on the spectrogram which uses a log frequency axis (100-2000 Hz) for the vertical axis. Fig.5(a) shows the extracted $F0(n)$ on the spectrogram. Although the detection of the silent intervals for speech is performed at the same time, it does not work since there is a directional white noise in the intervals. However, it can be confirmed that the pitch extraction algorithm does not pick up the wrong $F0(n)$ for the most of the intervals. Even though the pitch extraction algorithm is not evaluated here, it could be concluded that the lowest components of speech in the spectrogram, namely the F0s are fol-

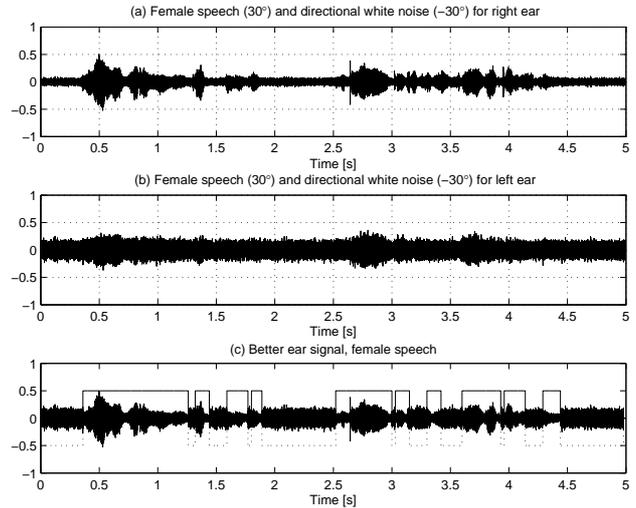


Figure 4. (a) Right ear signal, (b) left ear signal, (c) better ear signal and the results of "Choice of better ear signal" (SNR:10 dB at the location of sound sources)

lowed by the algorithm. Applying the extracted $F0(n)$ for Eq.(1)-(3), the harmonic stream is constructed as shown in Fig.5(b). $f0(n)$, $f1(n)$ and $f2(n)$ are indicated by the symbols shown in Fig.5(b). It can be seen that there are some $F0(n)$ discarded under 200 Hz intentionally, based on Eq.(1)-(3).

The frequency channels are selected by feeding back

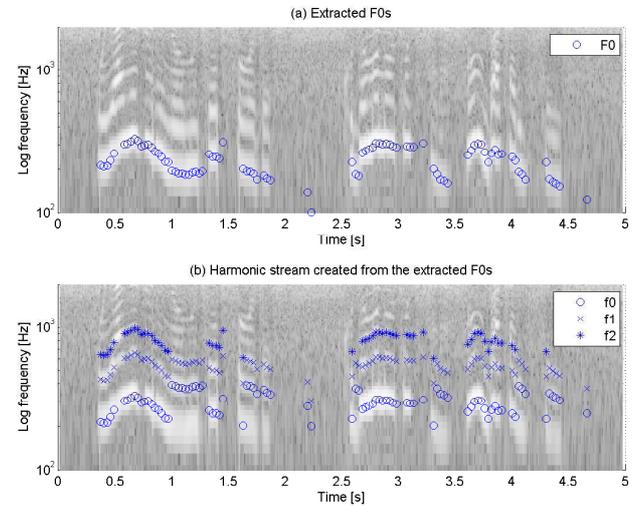


Figure 5. The results of pitch extraction and the harmonic stream based on Eq.(1)-(3)

the harmonic stream in Fig.5(b) at the next frame in order to calculate the ITD. Similarly, the offset is applied at the next frame according to the estimated angle. In Fig.6, the estimated angles are shown in the case of four different conditions, "none", "pitch", "offset" and "both"; "none" has no feedback of pitch and no offset, "pitch" has feedback of pitch but no offset, "offset" has no feedback of pitch but has the offset, and "both" has the feedback of

both pitch and the offset. There are no differences for the four conditions between the speech intervals and the silent intervals in terms of time. The most important point of the evaluation is the accuracy of the estimated angles for *speech*, such as the estimated angles for 30° . "none" and "pitch", to which the offset is not applied, are far from 30° and change around 40° - 50° . However, "offset" and "both", to which the offset is applied, the 30° angle is achieved, as expected. Therefore, applying the offset is very important and inevitable to building this system. It is noted that the estimated angles change to 30° after the estimation overshoot when the estimated angles changed from -30° to $+30^\circ$ rapidly since the offset is applied at one frame later. As for the estimated angles for the directional white noise, the estimated angles stay around -30° . This seems to be because the power of the white noise is constant over the frequency range, and ITDs in the higher frequency channels dominate the estimation of ITD by SCCF.

To evaluate the results more quantitatively, a correct an-

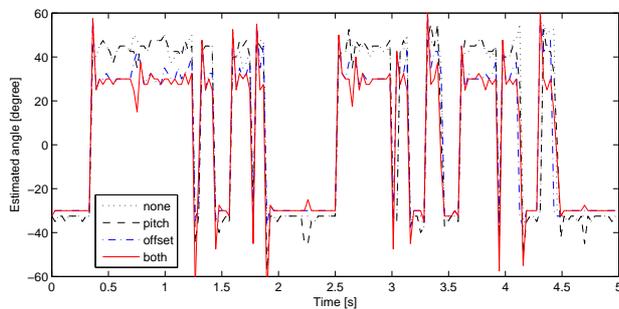


Figure 6. The temporal change of the estimated angles under four conditions)

swer for the estimated angles, $\theta(n)$, is prepared for 30° when the SNR is 20 dB, since the whole intervals of the better ear signal become 30° if a clean speech is used. To prepare the correct answer, the intervals when the right ear signal is chosen like the solid lines in Fig.4(c) are calculated, and then it is assumed as if the obtained intervals were all 30° . Here, accuracy (*Acc*) in [10] is quoted as the evaluation function. If the estimated angle is defined as $\hat{\theta}(n)$, *Acc* is defined as

$$Acc = \frac{1}{N} \sum_{n=1}^N \delta(\theta(n), \hat{\theta}(n)) \quad (5)$$

where N is the number of frames. $\delta(a, b)$ is defined as

$$\delta(a, b) = \begin{cases} 1, & \text{if } |a - b| < \beta \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where β is the tolerance and is fixed as 3° as [10] did. In Fig.7, *Acc* is shown under four conditions when SNR is set up as 10 and 0 dB. *Acc* of "none" and "pitch", in which the offset is not applied is very low as expected from Fig.6, and that means that the system is basically not working. In the case that the SNR is 10 dB, it is remarkable that *Acc* for "both" has 10% better accuracy than that for "offset" despite using only a few frequency channels. In the case

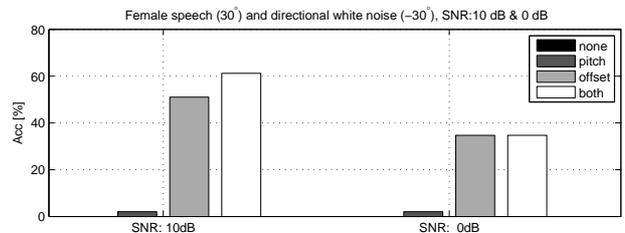


Figure 7. *Acc* under four conditions when SNR is set up as 10 and 0 dB

of SNR of 0 dB, *Acc* is wholly decreased. It is known that white noise as a masker affects the accuracy in the estimation of sound direction [22]. The reason why *Acc* of "offset" and "both" became the same accuracy might be the reduction of the accuracy in SCCF itself rather than the the reduction of the accuracy in the pitch extraction. It would be necessary to have a further investigation between the accuracy in the pitch extraction and SNR to determine the reason.

4. DISCUSSION AND FURTHER WORK

When the estimated angle moved from the angle of the directional white noise to that of the speech, overshoots were observed at "offset" and "both" in Fig.6. This might be understandable intuitively, in that it might take a certain time for the auditory system to estimate the accurate sound location if the location moved rapidly. In the proposed system, the frame size to calculate ITD was fixed at 30 ms, which is the same length as the frame size to calculate F0s. However, ± 1 ms is good enough to obtain the angles for the azimuth. Therefore, if the frame size is fixed at 1 ms for ITD and 30 ms for F0s, it might be possible to reduce the overshoots assuming a smoothing function. In addition, the frame size could be crucial to the temporal boundary between simultaneous and sequential organization. It is necessary to consider the compatibility with the findings of psychoacoustics.

In this paper, although the number of streams for the pitch extraction is defined as 1 such as speech or white noise, the multi-pitch algorithm is utilized in [9]. It is important for the model to investigate how many streams should be extracted at the same time in the simultaneous and sequential organization.

Although the importance of biological plausibility has been discussed since then, the pitch extraction algorithm is still too complicated to be implemented at neuronal level. Also, the method to create the better ear signal is indefinite in terms of the physiological view. Compatibility between the physiological and psychological view with the engineering tool requires more investigation.

5. CONCLUSIONS

In this paper, a simple model integrating ITD as a sound localization cue with the pitch cue is proposed. Considering the order that the decision process of pitch could be per-

formed after the extraction process of sound localization cues, a feedback system is employed. Although only one speech sample was utilized in the simulation, the integration shows 10% improvement of accuracy in the extraction of ITD. Integrating other auditory cues including IID for elevation is planned future work.

Acknowledgments

This research is supported under the Centre of Excellence in Intelligent Systems (CoEIS) project, funded by the Northern Ireland Integrated Development Fund and InvestNI.

6. REFERENCES

- [1] B. Glackin, J. A. Wall, T. M. McGinnity, L. P. Maguire, and L. J. McDaid, "A spiking neural network model of the medial superior olive using spike timing dependent plasticity for sound localization," *Front. Comput. Neurosci.*, vol. 4, no. 18, pp. 1–16, 2010.
- [2] S. N. Wrigley and G. J. Brown, "A computational model of auditory selective attention," *IEEE Trans. Neural Network*, vol. 15, no. 5, pp. 1151–1163, 2004.
- [3] J. K. Bizley and K. M. M. Walker, "Sensitivity and selectivity of neurons in auditory cortex to the pitch, timbre, and location of sounds," *The Neuroscientist*, vol. 16, no. 4, pp. 453–469, 2010.
- [4] E. M. Cramer and W. H. Huggins, "Creation of pitch through binaural interaction," *J. Acoust. Soc. Am.*, vol. 30, no. 5, pp. 413–417, 1958.
- [5] K. M. M. Walker, J. K. Bizley, A. J. King, and J. W. H. Schnupp, "Cortical encoding of pitch: Recent results and open questions," *Hear Res.*, vol. 271, pp. 74–87, 2011.
- [6] R. D. Patterson, S. Uppenkamp, I. S. Johnsrude, and T. D. Griffiths, "The processing of temporal pitch and melody information in auditory cortex," *Neuron*, vol. 36, pp. 767–776, 2002.
- [7] A. S. Bregman, *Auditory Scene Analysis*. Cambridge, MA: MIT Press, 1990.
- [8] E. D. L. Wang and G. J. Brown, *Computational Auditory Scene Analysis*. Wiley/IEEE Press., 2006.
- [9] J. Woodruff and D. L. Wang, "Sequential organization of speech in reverberant environments by integrating monaural grouping and binaural localization," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 7, pp. 1856–1866, 2010.
- [10] H. Christensen, N. Ma, S. N. Wrigley, and J. Barker, "Integrating pitch and localisation cues at a speech fragment level," in *Proc. of INTERSPEECH 2007*, pp. 2769–2772, 2007.
- [11] C. J. Darwin, E. W. A. Yost, A. N. Popper, and R. R. Fay, *Spatial hearing and perceiving sources in Auditory Perception of Sound Sources*. Springer, 2007, ch. 8, pp. 215–232.
- [12] J. F. Culling and Q. Summerfield, "Perceptual segregation of concurrent speech sounds: absence of across-frequency grouping by common interaural delay," *J. Acoust. Soc. Am.*, vol. 98, pp. 785–797, 1995.
- [13] N. I. Hill and C. J. Darwin, "Effects of onset asynchrony and of mistuning on the lateralization of a pure tone embedded in a harmonic complex," *J. Acoust. Soc. Am.*, vol. 93, no. 4, pp. 2307–2308, 1993.
- [14] W. Gardner and K. Martin, "HRTF measurements of a KEMAR dummy-head microphone," MIT Media Lab Perceptual Computing, Tech. Rep. 280, 1994.
- [15] W. G. Gardner, *3-D Audio Using Loudspeakers*. Boston: Kluwer Academic, 1998.
- [16] R. D. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, "An efficient auditory filterbank based on the gammatone function," *Cambridge, UK., Tech. Rep., MRC Applied Psychology Unit*, 1988.
- [17] B. R. Glasberg and B. C. J. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hear Res.*, vol. 47, pp. 103–138, 1990.
- [18] C. Liu, B. C. Wheeler, W. D. O'Brien, R. C. Bilger, C. R. Lansing, and A. S. Feng, "Localization of multiple sound sources with two microphones," *J. Acoust. Soc. Am.*, vol. 108, no. 4, pp. 1888–1905, 2000.
- [19] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 708–716, 2000.
- [20] R. J. Ritsma, "Frequencies dominant in the perception of the pitch of complex sounds," *J. Acoust. Soc. Am.*, vol. 42, pp. 191–198, 1967.
- [21] G. F. Kuhn, "Model for the interaural time differences in the azimuthal plane," *J. Acoust. Soc. Am.*, vol. 62, pp. 157–167, 1977.
- [22] C. Lorenzi, S. Gatehouse, and C. Lever, "Sound localization in noise in normal-hearing listeners," *J. Acoust. Soc. Am.*, vol. 105, no. 3, pp. 1810–1820, 1999.

SUPPORT FOR LEARNING SYNTHESISER PROGRAMMING

Mateusz Dykiert

University College London
m.dykiert@cs.ucl.ac.uk

Nicolas Gold

University College London
n.gold@ucl.ac.uk

ABSTRACT

When learning an instrument, students often like to emulate the sound and style of their favourite performers. The learning process takes many years of study and practice. In the case of synthesisers the vast parameter space involved can be daunting and unintuitive to the novice making it hard to define their desired sound and difficult to understand how it was achieved. Previous research has produced methods for automatically determining an appropriate parameter set to produce a desired sound but this can still require many parameters and does not explain or demonstrate the effect of particular parameters on the resulting sound. As a first step to solving this problem, this paper presents a new approach to searching the synthesiser parameter space to find a sound, reformulating it as a multi-objective optimisation problem (MOOP) where two competing objectives (closeness of perceived sonic match and number of parameters) are considered. As a proof-of-concept a pareto-optimal search algorithm (NSGA-II) is applied to CSound patches of varying complexity to generate a pareto front of non-dominating (i.e. "equally good") solutions. The results offer insight into the extent to which the size and nature of parameter sets can be reduced whilst still retaining an acceptable degree of perceived sonic match between target and candidate sound.

1. INTRODUCTION

When learning an instrument, students often like to emulate the sound and style of their favourite performers. The learning process takes many years of study and practice and yet for software synthesisers, the user must deal with complexities involved in specification of the timbre [1]. The vast parameter space makes it challenging for the inexperienced user to achieve a desired sound as a small change in one parameter can alter the result quite significantly. Several authors describe approaches to automatically determine a set of synthesiser parameters that can produce a desired target sound (see for example Horner [2], Johnson [3], Lai [4], and McDermott [5]). However, achieving a target sound does not necessarily help a user to learn the effect of particular parameters. What is required is both a way of achieving desired sounds, and also guiding a user in meaningful experimentation around that sound to aid

learning. This will require the development of underlying technology for automatic parameter derivation and the development of appropriate pedagogy and supporting tools.

As a first step towards this goal, this paper presents a reformulation of the synthesiser sound-matching problem as a Multi-Objective Optimisation Problem (MOOP) where two objectives are considered: closeness of sonic match, and number of parameters used to achieve the match. Maintaining an acceptable sonic match to the target sound while reducing the number of parameters required to attain it should aid an inexperienced user since the apparent complexity of the synthesiser is reduced. Determining the appropriate balance between the size of the parameter set and the closeness of match is a difficult problem as the trade-off may not be easy to judge (for example, it might be that there are parameter sets that are considerably smaller than the original but that do not substantially affect the sound, or in other cases it may be that a single parameter can have a major impact on the timbral quality). To gain insight into these trade-offs and inspired by an analogous problem in software engineering (see [6]), we have applied a multi-objective pareto-optimal search algorithm (NSGA-II). Such algorithms generate a pareto front of non-dominating (i.e. equally-good) solutions. Plotting these on a graph then allows the trade-offs between the size of parameter set and sonic match to be more easily seen.

The paper makes the following contributions:

1. A reformulation of the search for synthesiser parameter sets as a multi-objective search problem.
2. A solution representation for encoding parameters and fitness for use by the NSGA-II algorithm.
3. The results of an empirical study of the NSGA-II algorithm applied to a number of CSound synthesisers.

The remainder of the paper is organised as follows. Section 2 presents the reformulation of parameter search as a MOOP and introduces relevant theory. Section 3 describes the solution representations. Section 4 describes the experimental configuration that led to the results discussed in Section 5. Section 6 presents related work and Section 7 concludes.

2. RELATED WORK

There are several applications of genetic algorithms to the generation of synthesiser parameters to produce a sounds similar to a given target.

Yee-King and Roth [7] present a software synthesiser programmer. Their Java-based system is capable of automatically programming any VSTi compatible software synthesiser. A genetic algorithm provides the core of this system with parameters encoded as real numbers from 0 to 1 into a chromosome. The system uses proportionate roulette wheel selection and uniform random crossover as genetic operators. Mutation is achieved by adding a gaussian random variable. The fitness function is based on MFCCs compared between target and candidate solution to produce Square Root Mean Error. The system has been evaluated using expert users and produces close perceptually matching sounds. We adopt the same fitness approach for the calculation of sonic distance but our approach differs in the consideration of the second fitness objective (number of parameters), and the GA operators.

McDermott et al. [5] also use a genetic algorithm to deduce the set of synthesiser parameters. A comprehensive study of different fitness functions is presented, including pointwise metric, perceptual metric (made up from centroid, harmonicity and attack time), discrete Fourier transform metric and composite metric derived from simpler measures. They observe that all of these fitness functions perform well on simple target, but their performance is highly diminished when working on targets containing many partials. Their conclusion is that the perceptual measure is the most suitable for such computation. Our approach follows this finding in using a perceptually driven metric (although we use MFCC rather than the composite metric suggested) but differs in that our aim is to reduce the size of the parameter set.

Lai et al. [4] present another solution based on genetic algorithms for FM synthesis. The system is evaluated using different fitness functions. The sonic match is determined using spectral centroid and spectral norm derived from short time Fourier transform. Spectral centroid is found to perform better than spectral norm used alone, but much better results are obtained if these two functions are combined together. The system is evaluated against a piano tone generated by a Yamaha MA3 FM synthesiser with known pitch. Closely matched sounds are reported to be generated. Our work shares a similar overall approach but again, we differ in that we are undertaking a multi-objective approach and are currently focusing on additive synthesis (although FM would be an interesting avenue of future work for our approach).

3. PROBLEM DEFINITION

This section defines the sonic matching problem more formally as a MOOP. Multi-objective optimisation problems have two or more equally important objectives to be independently maximised (or minimised) simultaneously [8].

A synthesiser \mathcal{S} , uses a set \mathcal{P} of parameters to generate a sound. Each parameter n_p is an integer in the range 0-127 (corresponding to the possible MIDI CC values sent, for example, from a control surface). Although this restricts the possible values available for each parameter, the combination is still a large search space. If the parameters were

not restricted, many more iterations would be required before the algorithm converged on a set of good solutions.

An audio fragment \mathcal{A} is a 0.5 second fragment of audio of constant amplitude and pitch. The distance d between two sounds \mathcal{A}_1 and \mathcal{A}_2 is defined thus:

$$d = \sqrt{\frac{\sum_{i=0}^N (x_{1,i} - x_{2,i})^2}{n}}, \quad (1)$$

where n is the number of parameters, x_1 is the \mathcal{A}_1 's MFCC and x_2 the \mathcal{A}_2 's MFCC value.

The problem of sound matching with fewer parameters can therefore be seen as one of minimising d while simultaneously minimising the arity of \mathcal{P} .

3.1 Genetic algorithms

Genetic Algorithms (GAs) were first introduced by Holland in 1975 [9]. They are inspired by biological processes of evolution - selection, mutation, crossover, for optimising and solving complex problems. Genetic algorithms generate a random initial population which is then evolved through reproduction, forming new populations until a stopping condition is reached e.g. a specific number of function evaluations or the solution of required fitness is found.

3.1.1 Representation

The candidate solution is encoded into a chromosome. The way that the values are encoded in a chromosome is implementation dependent. Usually values are encoded as a binary string.

3.1.2 Selection

Selection is used for determining parents used in crossover. There exist many selection operators, such as roulette wheel and tournament [9].

3.1.3 Crossover

Crossover exchanges selected genes from parents obtained in selection. New offspring are created in this step [9].

3.1.4 Mutation

In order to maintain diversity of solutions within generations, the mutation operator randomly changes a gene of newly created offspring during crossover [9].

4. SOLUTION

This section describes the algorithms used and provides more detailed insight into our approach.

4.1 Parameter Encoding

Synthesiser parameters can have different types and value ranges. This introduces complexities for genetic algorithms due to the requirement of different type encodings within the chromosome. The complexity is carried forward to mutation and crossover operators, which need to make sure that the particular operation does not create invalid (out of range) solutions. Following certain standards can reduce

this complexity. For example, Yee-King and Roth [7] use a VST compatible backend, hence parameters are encoded as real numbers from 0 to 1. We follow a similar idea, adopting the MIDI specification where most continuous controllers conform to a range of integers between 0 and 127. This uniform way of storing values ensures that the parameter is always valid and within the range specified. A secondary list of parameters is maintained for lookup of maximum and minimum values for the further value projection required for generating candidate solutions.

In the case of the CSound patches used, we need to project the value into a real parameter range. In order to achieve this we use the following equation:

$$t = \frac{(a - b)}{128} \cdot v + b, \quad (2)$$

where t is the actual parameter value, a is the actual maximum parameter value, b is the actual minimum parameter value and v is the value stored in a chromosome.

4.2 Fitness Assignment

With the transformation of the problem into multi-objective optimisation problem, two fitness functions must be defined. This section describes the two fitness functions used in the test system.

4.2.1 Sonic Match

The first objective needs to be evaluated to classify how similar the candidate and target sounds are. In order to achieve this, we use Mel-Frequency Cepstral Coefficients (MFCC). The concept was originally introduced as measures for speech recognition [10] but its application in music has been growing ever since MFCCs provide perceptually meaningful means for the classification of audio signals.

MFCCs are computed in four steps [11]:

1. Applying the discrete Fourier transform on a windowed sound segment.
2. Mapping powers from resulting spectrum onto Mel scale.
3. Convolution of warped power spectrum with triangular band-pass filter and taking natural logarithm of the result.
4. Final computation of MFCC using Equation 3.

Davis and Mermelstein in [12] express the computation of MFCCs as:

$$MFCC_i = \sum_{k=0}^{20} X_k \cos\left\{i\left(k - \frac{1}{2}\right)\frac{\pi}{20}\right\}, \quad k = 1, 2, \dots, M, \quad (3)$$

where M is the number of cepstrum coefficients and X_k is the low-energy output of k th filter.

In a similar approach to the work of Yee-King and Roth [7], the fitness is expressed as the RMS Error of the MFCCs.

4.2.2 Number of Parameters

The second objective function must determine when a particular parameter has no effect on the sound generated during computation (in other words, for all practical purposes, that parameter is redundant). Each parameter has a value which causes no effect on the sound. These values are specific to the individual parameters, for example, a high-pass filter will not have any effect if the cut-off frequency is below the level at which humans can perceive sound. To calculate the number of parameters which actually have any effect, we sum all parameters for which the no-effect condition is false.

4.3 NSGA-II

Algorithm 1 shows the NSGA-II algorithm defined by Deb et al. [13].

Initially, a random parent population P_0 is created. Solutions are evaluated for fitness. Offspring population Q_t is created by selection using binary tournament, crossover and mutation.

Algorithm 1 NSGA-II - main loop

```

while stopping condition not met do
     $R_t = P_t \cup Q_t$ 
     $\mathcal{F} = \text{fast-non-dominated-sort}(R_t)$ 
     $P_{t+1} = \emptyset$  and  $i = 1$ 
    while  $|P_{t+1}| + |\mathcal{F}_i| \leq N$  do
        crowding-distance-assignment( $\mathcal{F}_i$ )
         $P_{i+1} = P_{t+1} \cup \mathcal{F}_i$ 
         $i = i + 1$ 
    end while
    Sort( $\mathcal{F}$ ,  $\prec_n$ )
     $P_{t+1} = P_{t+1} \cup \mathcal{F}[1 : (N - |P_{t+1}|)]$ 
     $Q_{t+1} = \text{make-new-pop}(P_{t+1})$ 
     $t = t + 1$ 
end while

```

NSGA-II introduces elitism to maintain the best solutions found so far. Each individual in the population (i.e. a set of parameter values) is assessed for the number of effective parameters and sonic match closeness. A fast non-dominated sorting algorithm is used to sort the population into different fronts. Each front has a different non-dominated rank, hence NSGA-II algorithm assigns a special fitness value to each solution according to the front on which it is located. Crowding distance is used as a second internal fitness value assigned according to the magnitude of the distance. Crowding distance is said to be an density estimate of the front.

4.4 Random Search

To provide a comparison with NSGA-II, we have also run the experiments using the Random Search (RS) algorithm in jMetal [14]. This technique was used to check the validity of the formulated problem, as NSGA-II should have no problem outperforming random search.

| Name | # Params | Filter 1 | Filter 2 | Filter 3 |
|------|----------|-----------|-----------|-----------|
| S1 | 20 | - | - | - |
| S2 | 21 | Low-Pass | - | - |
| S3 | 21 | High-Pass | - | - |
| S4 | 22 | Band-Pass | - | - |
| S5 | 22 | Low-Pass | High-Pass | - |
| S6 | 24 | Low-Pass | High-Pass | Band-Pass |
| S7 | 24 | Band-Pass | Low-Pass | High-Pass |

Table 1. This table describes filters used within CSound patches and the number of parameters which the patch accepts.

5. EXPERIMENTAL SETUP

Each experiment used a different target sound which was randomly generated using the same CSound patch. Each experiment was executed 8 times for each CSound patch for both NSGA-II and Random Search. The length of the target and all generated sounds was limited to 0.5s. The amplitude and fundamental frequency are constant.

In our system we are using MFCC implementation in Java from Comirva project [15] and the NSGA-II implementation of jMetal [14].

The CSound patch used in the experiments is made up from two oscillators which use composite waveforms generated by functions using GEN10 routines. Each function specifies 10 relative strengths of each fixed harmonic partial. The composite waveforms created are added together to produce a sound. This represents a CSound patch called S1. The Table 1 outlines the different orchestra files used by the system.

5.1 EA Parameters and Operators

Each experiment has been run for 5000 fitness function evaluations. The initial population size was set to 100, and similarly, the archive population was also set to 100. All experiments used the same operators: single point crossover, bit flip mutation and tournament selection. All experiments were run with crossover probability $P_c = 0.7$ and mutation probability to $P_m = 1/n$ (where n is the number of parameters). Synthesiser parameters were encoded as integers, in range from 0 to 127, to provide uniform and MIDI compatible representation in a chromosome.

6. RESULTS

In this study we investigate the feasibility of our concept to reduce the number of parameters necessary to achieve a desired sound. In order to achieve this, we use the seven different CSound patches outlined in Table 1. The patches have a similar base structure, but they differ in complexity. Patches S6 and S7 contain the same components, but different arrangements and signal routing.

The results are shown in Figure 1. Each subfigure shows the NSGA-II obtained non-dominated front with all solutions generated by Random Search algorithm (the point cloud in each diagram). The lines joining points on the

pareto front are not themselves meaningful but are simply included to show the front more clearly. Subfigures a-g represent typical runs of the experiment (5000 function evaluations) on seven patches described. Figure 2 presents results for the S1 patch over 20000 function evaluations. Clearly, the increased number of function evaluations increased the closeness of the match and minimised the number of parameters more than an experiment with 5000 function evaluations. This situation is not always the case with other patches - sometimes the sonic match cannot be better, and the number of parameters required is not minimised further.

The results obtained show that the NSGA-II algorithm usually obtains a solution with a closer sonic match to the target than Random Search. NSGA-II is also better in minimising the number of parameters having an effect on the generated sound, finding such solutions more quickly and with more success. The lower extreme of the parameter number of parameter space is explored much further than with Random Search, converging on more suitable solutions. On average NSGA-II finds a solution requiring around two parameters less than equivalent solution obtained by RS. Subfigure Figure 1(a) clearly shows a good example of this; comparing similar sonic match values, the best solution obtained by NSGA-II algorithm has 17 parameters whereas the Random Search solution requires 20.

The repeated runs of the same experiment are fairly consistent and follow a similar trend.

The NSGA-II solution will scale better, judging by experiments involving S6 and S7 as it handles parameter optimisation better in more complicated patches. Both S6 and S7 have 24 parameters and the number of parameters is reduced the most in comparison to other simpler experiments involving 18 parameters.

In rare cases, the random search has produced a singular very, very good solution with a large number of parameters.

Extended runs of the experiments, including 10k and 20k function evaluations, have shown that the NSGA-II algorithm outperforms Random Search. With increased number of iterations, RS has improved slightly, and some solutions with reduced number of parameters are obtained.

The performance of the test system is fairly good and consistent. The time it takes to perform an experiment with 5000 iterations is around 6 minutes on a machine with Intel Core i3 2.4GHz processor with 4GB RAM. The running time increases linearly to 13 minutes for 10k and 27 minutes for 20k function evaluations.

7. CONCLUSIONS AND FUTURE WORK

This paper has presented a reformulation of the search for synthesiser parameters as a multi-objective optimisation problem to reduce the complexity of synthesiser programming. Two objectives were considered: the closeness of sonic match (in common with other previous methods in this area), and the number of effective parameters in the

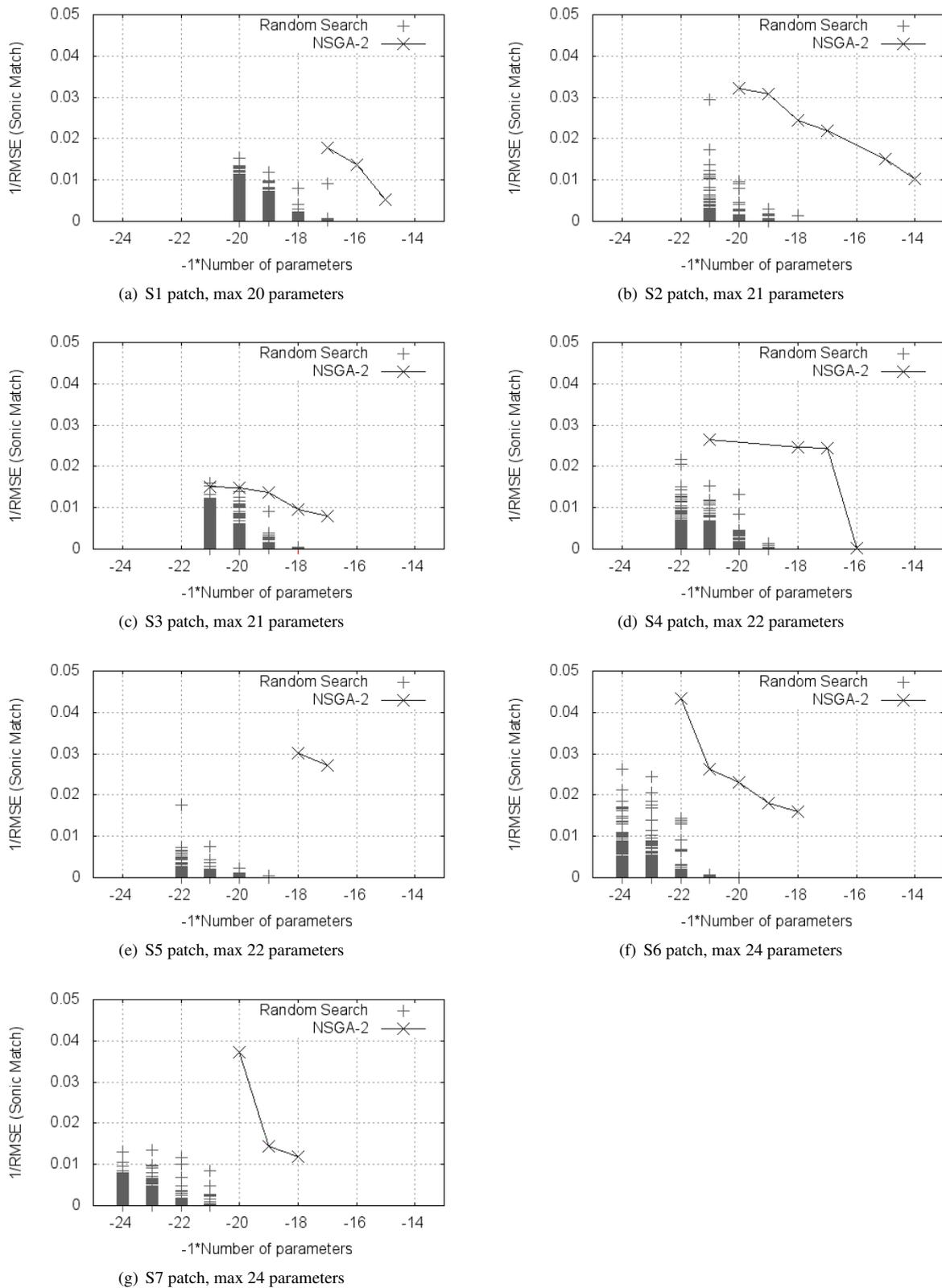


Figure 1. Graphs showing pareto front obtained by NSGA-II vs solutions generated by Random Search algorithm

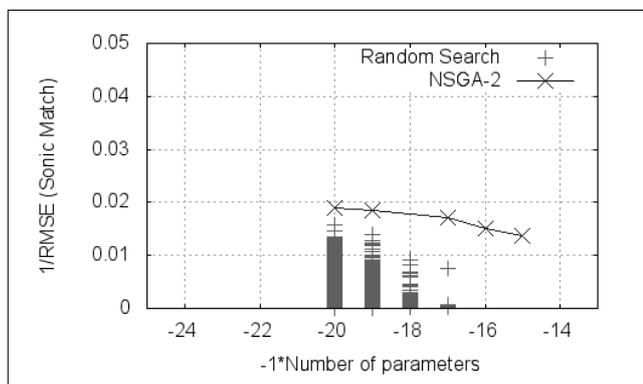


Figure 2. S1 experiment, 20k function evaluations.

generating set. A state of the art pareto-optimal search algorithm (NSGA-II) was applied to generate a pareto front of non-dominant solutions. The results were also compared to Random Search.

The results indicate that it is possible to reduce the size of the parameter set whilst still maintaining a good match to the target sound. Using a pareto-optimal search algorithm to generate pareto fronts allows insight into the trade-off between the size of parameter set and the closeness of match, showing where further reductions in the size of the set make large differences to the match distance. The use of a single synthesiser (Csound) means that it is possible that results may not generalise to others. This is counter-balanced by the fact that the synthesis method used here is standard additive synthesis and the parameters are expressed in a MIDI compatible form, amenable to use with other synthesisers.

There are a number of directions for future work. Further investigation of the content of patches on the pareto front is expected to offer more comprehensive insight into the parameters that are most easily eliminated. Other standard search algorithms will be tried to improve the efficiency of the pareto front generation. It is possible that a hybrid co-evolutionary approach of genetic programming and genetic algorithms will offer the possibility of further simplification by modifying the architecture as well as the parameter set. More complex synthesiser architectures will also be investigated (for example, the DX7 implementation in Csound) and subsequently non-Csound synthesisers programmed via MIDI.

Acknowledgments

We thank the members of CREST (in particular Shin Yoo and Mark Harman) for useful discussions and are grateful to the Computer Science Department at University College London for supporting this work.

8. REFERENCES

- [1] R. Boulanger and J. Ffitch, "Teaching software synthesis through Csound's new modelling opcodes," in *Proceedings of the International Computer Music Conference*, 1998.
- [2] A. Horner, J. Beauchamp, and L. Haken, "Machine tongues XVI: Genetic algorithms and their application to FM matching synthesis," *Computer Music Journal*, vol. 17, no. 4, pp. 17–29, 1993.
- [3] C. Johnson, "Exploring the sound-space of synthesis algorithms using interactive genetic algorithms," in *Proceedings of the AISB Workshop on Artificial Intelligence and Musical Creativity*, Edinburgh, 1999.
- [4] Y. Lai, S. Jeng, D. Liu, and Y. Liu, "Automated optimization of parameters for FM sound synthesis with genetic algorithms," in *International Workshop on Computer Music and Audio Technology*, 2006.
- [5] J. McDermott, N. Griffith, and M. O'Neill, "Toward user-directed evolution of sound synthesis parameters," *Applications on Evolutionary Computing*, pp. 517–526, 2005.
- [6] Y. Zhang, M. Harman, and S. Mansouri, "The multi-objective next release problem," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM, 2007, pp. 1129–1137.
- [7] M. Yee-King and M. Roth, "Synthbot: An unsupervised software synthesizer programmer," in *Proceedings of the International Computer Music Conference*, 2008.
- [8] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Wiley, 2001.
- [9] J. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. The University of Michigan Press, Ann Arbor, 1975.
- [10] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," *Pattern Recognition and Artificial Intelligence*, vol. 116, 1976.
- [11] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of MFCC," *Journal of Computer Science and Technology*, vol. 16, no. 6, pp. 582–589, 2001.
- [12] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.
- [13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [14] J. Durillo, A. Nebro, and E. Alba, "The jMetal framework for multi-objective optimization: design and architecture," in *IEEE Congress on Evolutionary Computation (CEC)*, 2010, pp. 4138–4325.

- [15] M. Schedl, "The CoMIRVA Toolkit for Visualizing Music-Related Data," Department of Computational Perception, Johannes Kepler University Linz, Tech. Rep., June 2006.

LEECH: BITTORRENT AND MUSIC PIRACY SONIFICATION

Curtis McKinney

Bournemouth University
cmckinney@bournemouth.ac.uk

Alain Renaud

Bournemouth University
arenaud@bournemouth.ac.uk

ABSTRACT

This paper provides an overview of a multi-media composition, *Leech*, which aurally and visually renders BitTorrent traffic. The nature and usage of BitTorrent networking is discussed, including the implications of wide-spread music piracy. The traditional usage of borrowed musical material as a compositional resource is discussed and expanded upon by including the actual procurement of the musical material as part of the performance of the piece.

The technology and tools required to produce this work, and the roles that they serve, are presented. Eight distinct streams of data are targeted for visualization and sonification: Torrent progress, download/ upload rate, file name/ size, number of peers, peer download progress, peer location, packet transfer detection, and the music being pirated. An overview of the methods used for sonifying and and visualizing this data in an artistic manner is presented.

1. INTRODUCTION

The internet has altered the manner in which we as individuals live our daily lives. Like any technology, the internet can be used for both positive and negative applications. Global communications allows families living in far off lands to connect and bond over vast distances. However, that same capability may also be used by nefarious forces to coordinate violent activities. One prominent ramification of networking technologies is music piracy. According to a recent survey commissioned by NBC Universal as much as 23.76% of all internet traffic infringes upon copyright [1]. This represents a rather substantial proportion, throwing into question just what exactly our society deems as acceptable behavior. The combination of MP3s, ipods, the internet, and the push for "singles" by record companies have all led to a commoditization of music. With this commoditization has come a reevaluation in our society of the monetary value of musical content.

One of the catalysts that has led to such a high level of piracy is a peer-to-peer networking technology known as BitTorrent [2]. This technology was not initially intended to be used for exchanging pirated content; BitTorrent is indeed used in the distribution of many completely legal downloads, including open source software and commer-

cial video games. However, according to the previously mentioned study it is estimated that approximately 63.7% of all BitTorrent traffic infringes copyright.

To help illustrate the moral and physical dynamics of music piracy a new multi-media composition, entitled *Leech*, has been constructed. *Leech* includes components of sonification and music composition, using the actual mechanisms that enable BitTorrent downloads as mined data for real-time algorithmic sound production. Network data and structure is mapped in musically and visually meaningful ways to produce an experience that embodies the look and sound of piracy. Furthermore, the actual music being pirated is itself used as a resource for audio processing and music composition. Performed in real-time, the composition provides multi-factorial insight into the world of music piracy as it happens that very moment. For reference, a video capture of a performance of *Leech* can be found at <http://vimeo.com/21603631> [3].

2. GOALS AND AESTHETICS

There are two main goals that drove the development of *Leech*. The first goal is to be sensuously arresting and interesting. *Leech* is conceptualized as a multi-media composition, not just a simple sonification. Thus it should be able to stand on its own as a composition in and of itself. The second goal of *Leech* is to make people think. No judgement is passed on the activities, legal or not, of the pirates, record companies, or musicians involved. Rather, the goal is to illuminate an often overlooked and dismissed part of modern day culture, and to spark a dialog about the nature and value of music and sharing.

Central to these goals is the often conflicting relationship between sonification and composition. If executed well, a visually and sonically interesting composition can draw attention to a subject being sonified. If done poorly, these elements may distract or even put off. Thus a strong compositional hand is required to make the experience enjoyable and thought provoking. However, if there is too strong an aesthetic influence then the data being sonified can be lost completely among the intuitive parameters of the piece.

Thus the reasoning for mapping particular pieces of data to different musical and visual parameters becomes a compromise between intuition and transparency. Table 1 shows an overview of how the different data types in a BitTorrent download are sonically and visually mapped in *Leech*. This will be covered more in depth in Section 6 and Section 7.

3. HISTORICAL PRECEDENT

3.1 Borrowed Sounds

The concept of using copyrighted audio as a musical resource for quotation and variation is not a new one. Early electronic music composers such as Edgard Varese and Iannis Xenakis borrowed musical material using analog tape. James Tenny used samples of Elvis Presley's "Blue Suede Shoes" in his composition *Collage #1* in 1961 [4]. With the advent of digital samplers in the 80's, using copyrighted material became even more commonplace. John Oswald explored the usage of popular music as a compositional resource in his album *Plunderphonics* [5]. Hip-hop artists such as Public Enemy and Biz Markie embraced the practice of sampling, and were given almost free-reign until the once underground musical style started to become profitable for record labels. Since then record companies have formed whole departments dedicated to finding copyright infringing samples in newly released songs [6].

Leech attempts to take this idea of borrowed musical material a step forward, to make transparent not only the usage of the material, but also the procurement of it as well. John Cage's works *Radio Music* and *Imaginary Landscapes #4* are especially relevant in this regard. These works explore capturing radio signals and noises from the electromagnetic spectrum, using these sounds as musical materials in real-time. This technique is quite similar to the system used by *Leech*, which captures sounds and data from packet streams on a peer to peer network. However, *Leech* has the added foil of explicit illegality. This is a rather appropriate attribute in a day and age where illegally sharing musical experiences is commonplace.

3.2 Sonification

Sonification is often understood to be a scientific activity, primarily aimed at finding another means for understanding a complex data set. In his book *Auditory Display: Sonification, Audification, and Auditory Interfaces* Gregory Kramer describes sonification as "the mapping of numerically represented relations in some domain under study to relations in an acoustic domain for the purpose of interpreting, understanding or communicating relations in the domain under study" [7]. However many composers have approached sonification as a more artistic activity, not necessarily devoted to attempting to provide purely intellectual clarity to a data set, but instead using it as a musical resource to drive a composition.

Early examples of this include *Reunion* by John Cage, which uses a chess board as an audio mixer, and *Music for Solo Performer* by Alvin Lucier, which involves amplifying brainwaves to the point of acoustically activating percussion instruments. Marty Quinn investigates sonification of natural forces in multiple works, including *The Climate Symphony* which generates gamelan-esque rhythmic music based on the pulsating climatological history of the earth, and *Rain*, which converts the intensity of ice melting over time into pitch and rhythm for percussive sounds. Bob Sturm uses the undulations of the ocean's waves in his piece *Music from the Ocean* to drive electronic music,

| Mined Data | Mapping |
|-----------------------------------|----------------------|
| Torrent Progress(%) | Timbral Complexity |
| Download/UploadRate(kB/s) | Envelope Attack time |
| File Names/Sizes(mB) | Visual presentation |
| Number of Peers(int) | Visual presentation |
| Leecher vs. Seeder(%) | Synthesis Type |
| Peer Location(ϕ/λ) | Pitch/Timbre |
| Packet Transfer(ϕ/λ) | Pitch/Timbre |
| MP3 | Processed Playback |

Table 1. Mined data and a brief overview of how they are mapped.

creating 34 different data mappings to produce individual musical tracks [8]. Especially pertinent to *Leech* is the piece *Network Sonification* by Zach Layton, which crawls websites, examining their link and data structures, converting this data into a kind of aural snapshot of their network topology [9].

4. TECHNOLOGICAL OVERVIEW

Leech involves several interlocking open source technologies. The visuals and logical systems are developed with the Java programming language [10]. The BitTorrent transfers are accomplished using the OSX application Transmission [11]. Analysis of transfer traffic is executed with the Java library Jpcap [12]. Geographic placement of peers is derived using the freely-distributed version of Max Mind's GeoLite City [13].

Visual representation and GUI elements are developed with the Processing programming language, used as a library from within Java [14]. Sound is produced with the real-time sound synthesis programming language SuperCollider [15]. Communications between Processing and SuperCollider is accomplished with the Open Sound Control (OSC) protocol [16]. LAME is used to convert partially completed MP3 downloads and load them into SuperCollider for audio processing [17].

5. DATA MINING

The basis for all of the visual and musical content in *Leech* is derived from data-mining. Therefore it is the data-mining technologies that are the core engine of the whole system, driving the flow of the entire experience. There are three distinct modules that act in coordination to derive information about the BitTorrent transfer:

5.1 Torrent Control

The first module is a Remote Procedure Calls (RPC) communication layer that controls and queries the Transmission BitTorrent Client. Through individual calls to Transmission the module can control the torrent download by starting and stopping the transfer, altering the number of peers to download from, and increasing or decreasing transfer speed. Data can be requested about the download, in-

| Mined Data | Module |
|-----------------------------------|------------------------|
| Torrent Progress(%) | Torrent Client |
| Download/UploadRate(kB/s) | Torrent Client |
| File Names/Sizes(mB) | Torrent Client |
| Number of Peers(int) | Torrent Client |
| Leecher vs. Seeder(%) | Torrent Client |
| Peer Location(ϕ/λ) | Torrent Client/GeoLite |
| Packet Transfer(ϕ/λ) | GeoLite/JPCap |
| MP3 | Torrent Client/Lame |

Table 2. Mined data and the modules used to derive them

cluding ip addresses of peers, name and size of the torrent files, download rate, and progress of download.

5.2 Geolocation

The second module utilizes the IP address database GeoLite City. This database contains the geographic location of most of the distributed IP addresses on the internet. Regularly updated, the freely distributed version is accurate to the city level in most cases, which is more than adequate for the purposes of this piece. By cross referencing this database with the IP addresses obtained from Transmission, it is possible to geographically place the peers that are transferring pirated audio.

5.3 Packet Capture

The third data-mining module monitors internet traffic on the local machine, capturing each packet of information that is being transferred to and from the localhost. From these packets of information it is possible to derive the sending and receiving IP address and payload information. By cross referencing this module with the previous two modules it is possible to derive when a packet of pirated BitTorrent information is being transferred between the localhost and particular peer. This information may then be depicted geographically, and sonically rendered.

6. MAPPING DATA

Leech is a multi-media composition, and thus it is not merely enough to derive the characteristics of a torrent download. Mapping this information in a visually and musically meaningful way is the challenge of the entire composition. The basic visual backdrop is a vectorized world map, upon which all other mined information is depicted(See Appendix A).

6.1 Peer Mapping

Using the three data-mining modules it is possible to derive several characteristics of a peer. A peer's geographic location, download progress, and when they are sharing pirated information can all be derived. Using Processing, the geographic location of a peer is rendered visually as a pulsating ellipse placed geographically on a vectorized world map. The color of the ellipse denotes the progress of the peer's own torrent download. A peer with less than 100%

downloaded is represented with a white ellipse, and is referred to as a "leecher". A peer that has finished downloading and is currently only uploading data is represented with a green ellipse and is referred to as a "seeder". Currently there is no static sonification of a peer's geographic position, or when a peer is added to the system. Instead these parameters are sonified in conjunction with other mapping systems described later.

6.2 Transfer Progress Mapping

The overall progress of the BitTorrent download and the individual progress of each MP3 transfer are also mapped visually and sonically. On the left hand side of the screen a series of bright blue bars are shown extending horizontally towards the center. As the transfer progresses to completion these bars extend further out. The names of each MP3 being downloaded is displayed over their respective bar to show their respective download progress. Sonically, these values are mapped much more directly than the packet transfer sounds, and it is much more easily cognizable to hear the effect of the download on these sounds.

One synth is produced for each individual MP3 being downloaded, usually in the range of 10 to 15 depending on the size of the album being downloaded. These sounds undulate as a kind of ambient background to the piece. As the download progresses from the beginning to completion several characteristics of the sound are modulated. High frequency content, undulation speed, feedback amount, and general timbral complexity all increase as the download progresses.

Figure 1 shows a snippet of SuperCollider code mapping file transfer data to synthesis parameters. Depicted is a Gendy stochastic oscillator, a concept conceived by composer Xenakis in his treatise *Formalized Music* [18]. Unlike periodic oscillators that oscillate linearly, this oscillates based upon a given distribution of probabilities. The oscillator is being deployed in sinus mode, which means that it is sampling an outside oscillator to provide a constantly shifting probabilistic distribution. The third and fourth inputs are the external oscillators being sampled, which are themselves Gendy oscillators(not depicted here). Inputs five and six determine the frequency of the oscillator. Very simply, as the download progresses, the pitch goes up. The final slot depicts the number of control points sampled during one period of oscillation. As the download progresses, the amount of control points sampled per period increases, thus increasing high frequency content and timbral complexity. This demonstrates a very direct influence of the download being exerted on the sound. The staggered progression of each file transmission produces a heterophonic texture that moves as a loosely connected cloud from relative timbral simplicity to more intense and complex tonal emissions. This is useful in giving an overall form and shape to the piece.

6.3 Packet Capture Mapping

Each time a packet of information is identified as being part of the torrent download, the system identifies the parties sending and receiving the pirated information. This

```

osc3 = Gendy4.ar{
  6, //Sinus Mode
  6, //Sinus Mode
  osc1, //Sampled Oscillator
  osc2, //Sampled Oscillator
  fileProgress.linlin(0,1,520,47000), //Min Freq
  fileProgress.linlin(0,1,520,47000), //Max Freq
  initCPs: 100, //Initialized Control Points
  knum:fileProgress.linlin(0,1,40,100).round(5)
};
    
```

Figure 1. File transfer sonification code in SuperCollider.

determines whether or not the localhost is downloading or uploading information, and to whom they are uploading to or downloading from. Furthermore, by cross referencing against the attributes of the peer involved, it is possible to depict whether the transfer involves a seeder or a leecher. Using these parameters the system organizes packet transfers into four subtypes: downloads from leechers (DL), uploads to leechers, (UL) downloads from seeders (DS), and uploads to seeders (US).

Whenever a packet transfer is identified it is rendered visually as a colored curve stretching from the localhost to the peer involved. The orientation and color of the curve depict what type of transfer it is. A DL transfer is a white line curving upwards. UL transfers are white and curve downwards. DS transfers are green and curve upwards. US transfers are blue and curve downwards. This information is also passed to SuperCollider via OSC to be rendered sonically.

In SuperCollider there are four types of synthesized sounds that are produced based upon the four packet transfers types. Characteristics of the packet transfer are also used to further modulate the characteristics of these sounds. Download rate, local transfer progress, peer transfer progress, and peer latitude and longitude are all characteristics that influence the synthesized sounds. Due to the large quantity of packet transfers throughout the course of a twenty minute performance, emphasis is placed more on variety of results rather than on simplistic sonifications of values. Thus it is difficult to briefly summarize how these values are mapped in each synthesized sound. Instead of attempting to dissect a large amount of sonification code, a small example of one line of code is provided to give some idea of the techniques used to sonify the packet data.

Figure 2 depicts a snippet of code near the end of a packet capture sonifying a synthesizer in SuperCollider. This code depicts a delay line that is processing an earlier synthesized audio signal. The input signal is being modified by two nested single pole band-pass filters. These filters' resonant frequencies are modulated by the geographic location of the peer that the packet of information is being transferred to or from. Thus the further west a peer is the more high frequencies in the first filter. This is fed into the second filter which filters out more low frequencies the further south the peer is located. The progress of the peer's download determines the delay time of the delay line. Peer progress ranges from 0.0 to 1.0, however here that value is being wrapped at a modulus of 0.5. Thus, as peer progress advances from 0.0 to 1.0, the delay time of the delay line will

```

BufDelayC.ar{
  LocalBuf(44100*0.5),
  OnePole.ar{
    OnePole.ar{
      synth,
      lat.linexp(-150,150,-0.99,0.99)
    },
    lon.linexp(-150,150,-0.99,0.99)
  },
  (nodeProg%0.5),
  0.75,
  synth*0.75
},
    
```

Figure 2. Packet capture sonification code in SuperCollider.

start at 0.0 seconds and reach a peak of 0.5 seconds at the mid point, then return to 0.0 and increase to another peak of 0.5 at the completion of the peer's download. Finally the original signal is summed with the delay line and fed into a feedback loop(not shown) to produce a recursively filtered echo effect.

This is one part of a much more complicated and interwoven whole, with each mapped parameter serving many purposes throughout the whole sound. This produces the desired effect: an intricate and constantly evolving sound with a wide array of variety to sonify the many different characteristics of the thousands of packets of pirated information that are transferred throughout the performance.

6.4 Pirated Music Playback

The final system manages the actual audio that is being pirated. This system is not so much mapping as it is resource collection. This system also addresses the real goal of pirating MP3's, which is to actually *listen* to them. Thus it seems technically and musically logical to provide a system for playing back these stolen sounds. By using the keyboard the performer may move a red rectangle between the MP3 progress bars. Pressing certain buttons will convert the selected MP3 into a WAV file and load it into SuperCollider. If the file is incompletely transferred, it creates a WAV file that skips missing audio data, providing a shorter audio file with sharp jump cuts. Then the system employs one of several playback synths that alter the audio in different manners. The goal with these synths is to playback the audio in heavily altered yet still somewhat recognizable fashion.

Figure 3 shows an example of SuperCollider code that plays back pirated audio data. This system uses Fast Fourier Transformation (FFT) processes initialized with very large buffer sizes. Using a spectral buffer playback system, this plays the audio data at 3% of its original speed while maintaining the same pitch. Next the audio's spectral data is squeezed into half the space it normally fills. A brick wall filter is placed upon the signal to discard most of the high spectrum and leave the low end data. The low frequency data is then spectrally enhanced, placing three new harmonics above each frequency in the spectrum. Lastly it is once again squeezed into half of the spectral field. This produces a rich and slowly evolving low end drone sound

```

bufnum2 = LocalBuf.new(1024*16, 1);
chain = PV_PlayBuf(bufnum2, recBuf, 0.03, 0, 1);
chain = PV_BinShift(chain, 0.5);
chain = PV_BrickWall(chain, -0.95);
chain = PV_SpectralEnhance(chain, 3, 2, 5);
chain = PV_BinShift(chain, 0.5);

```

Figure 3. Pirated MP3 playback code in SuperCollider.

that is heavily influenced by the bass drum and bass lines of a pirated song. It is thoroughly altered, however given familiarity with a song it is actually rather easy to detect a slow moving distorted version of the bass present in a song. While the other two systems are (more or less) tuned, this system consciously makes no effort to alter the tonality of the original song. A combination of these three distinct layers, the droning file transfer mapping, the percussive packet capture sounds, and the processed songs, produces a kaleidoscopic polytonal morass.

7. ARTISTIC CONSIDERATIONS

7.1 Non-linearity and Pseudo-linearity

The network data being mapped in *Leech* may be categorized by the manner in which it traverses its range. Pseudo-linear data moves in one direction, never skipping forward or backwards. This includes the overall download progress, progress for each individual torrented file, and number of peers that have connected to the system. This data is not strictly linear however, as the time span it takes to traverse the range of this data is not predetermined and differs for each performance and for each datum. Other data traverses its range non-linearly, skipping forward and backward at differing rates of speed. This includes download/upload rate, peer locations and peer download progress.

Having these two different types of data present is quite useful for creating a musical composition. Linear data allows the piece to have an overall form and shape, and to create a sense of tension, much like a normal non-realtime precomposed piece. *Leech* will always start off with quiet drones in the beginning, with the download progress at zero. As the piece advances, the download progress reaches closer to 100%, the drones increase in amplitude and complexity, creating a long build in tension. However, the non-linear data serves to provide variety in the piece. While pseudo-linear data tends to have an effect on the top most scale of the piece, being the form, the non-linear data provides unpredictable embellishment at the note scale. Download and upload rate are in constant flux, and each peer that a packet is transferred to will have a different and unpredictable geographic coordinate and download completion. These constant variations on a smaller time scale produce different tonal and timbral figures and patterns and add unpredictability from moment to moment. In combination these two forces give the system a sense of direction and life.

7.2 Transparency

Transparency in presenting the music being pirated is central to the piece. In compositions that focus on the concept of borrowed material, such as Luciano Berio's *Sinfonia Mvt. 3*, it can be at times difficult to identify exactly what is being borrowed and manipulated. *Leech* attempts to balance creative musical modification with transparency. Audio effects that maintain cognizable portions of the sonic material are purposefully employed.

One example of this is FFT based speed reduction, which create long evolving drones while maintaining identifiable pitch material. This audio transparency is accentuated by the use of visuals in the piece. The name of the artist, album, and each individual MP3 is clearly displayed in the visuals to inform the audience of exactly which material is being downloaded. Whenever a song is selected to be played back in modified form, it is hi-lighted on the screen to inform the audience exactly what song they are hearing being processed.

7.3 Collaboration

Collaboration is key to the mechanisms and compositional underpinnings of *Leech*. The process of illegally obtaining music is in fact a social and communal activity. Peer-to-peer networks such as BitTorrent require that a group of users proliferate information between each other in a mutually beneficial structure. The visuals in *Leech* attempt to demonstrate that the act of piracy brings together people from across the globe (though with less frequency in places such as Africa and China where free internet usage is restricted or unavailable). These peers are from many different cultures and societies, setting aside any differences to collaboratively share music.

The sounds themselves are also a collaboration. Two composers are involved in the artistic production of the piece, Curtis McKinney and Chad McKinney, twin brothers who have been collaborating for years on musical compositions. These composers also collaborate with the artist's whose works are being sonically manipulated. Furthermore, the actual choice of what to pirate for performance of the piece is determined by popularity on the BitTorrent search engine <http://isohunt.com>. Using this selection process popular artists such as Rhianna and Lady Gaga have been used for the piece in recent past.

8. FUTURE WORK

Several new capabilities will be added to *Leech* in the coming future. As it stands the main interaction a performer has with the system is simply through starting and stopping the different pirated songs being downloaded. In the future more performative controls will be added to make performance more gratifying and emotive. These control will be primarily aimed at changing the manner in which the sonification occurs through a performance. One example of this would be to introduce a parameter that may alter the range of pitches that the location of a peer causes to occur. The capability to make changes such as this dur-

ing performance would add more variety and musicality to the system.

Creating a version of *Leech* meant for presentation as an installation piece is also under review. This would involve streamlining setup, creating an auto-resetting capability, and changing the system so that participants are able to download whatever song they choose through a search prompt and keyboard.

Plans are under way for future pieces that utilize sonification of illicit network streams. A new piece entitled *Panopticon* is being constructed which will visualize and sonify a local area hacking technique known as arp poisoning. Using this method it is possible to monitor the network traffic of an entire local area network, peering into the private lives of individuals, and capturing private information. Viewing this private information in real time it is possible to see and hear what websites various users are browsing, what videos and music they are streaming, and any password or bank information they enter while on the network. This piece will serve to explore the ramifications of the dawning age of free information and loss of privacy that technology is thrusting upon us.

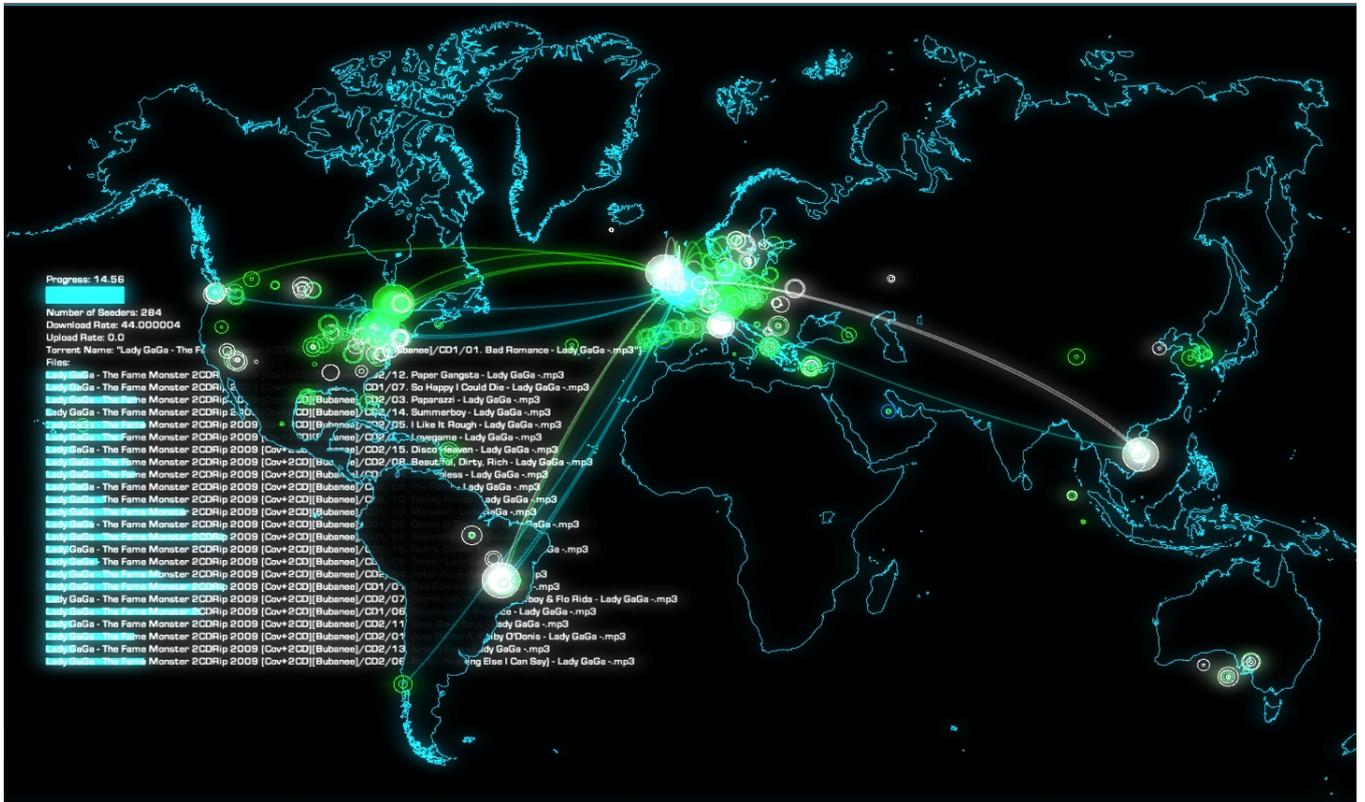
9. CONCLUSION

The overall goal of *Leech* is to produce a kaleidoscopic visual and sonic insight into what many in our society deem worthy enough to share and listen to, but not to buy. This instrument lays bare not only the hidden mechanisms that computers use to communicate, a fascinating display in itself, but also the often times contradictory relationships that our culture has with music, community, and commerce. Given the ubiquity of piracy it cannot be simply swept under the rug as a symptom of irresponsible individuals. In a typical performance of *Leech* approximately 200-400 individuals may be observed as being involved from the world over. The sounds that emanate from the piece are derivative of these individuals' choice to partake in a communal illegal activity, the mechanisms used to enable this choice, and the music that they have chosen to steal. It is the sound of piracy itself.

10. REFERENCES

- [1] Envisional, 2011, available from: documents.envisional.com/docs/Envisional-Internet_Usage-Jan2011.pdf [Accessed 5 March 2011].
- [2] BitTorrent, 2011, bitTorrent protocol specification. Available from: http://bittorrent.org/beps/bep_0003.html [Accessed 4 March 2011].
- [3] C. McKinney and C. McKinney, "Leech," 2011, available from: <http://vimeo.com/21603631> [Accessed 18 May 2011].
- [4] B. Sturm, "Concatenative sound synthesis and intellectual property: An analysis of the legal issues surrounding the synthesis of novel sounds from copyright-protected work," *Journal of New Music Research*, vol. 35, no. 1, pp. 23–33, 2006.
- [5] J. Oswald, "Plunderphonics, or audio piracy as a compositional prerogative," *Music Works*, vol. 34, 1986.
- [6] K. McLeod, *Freedom of Expression: Overzealous Copyright Bozos and Other Enemies of Creativity*. Doubleday, 2005.
- [7] G. Kramer, *Auditory Display: Sonification, Audification, and Auditory Interfaces*. Perseus, 1993.
- [8] B. Strum, "Pulse of an ocean: Sonification of ocean buoy data," *Leonardo Music Journal*, vol. 38, no. 2, pp. 143–149, 2005.
- [9] Z. Layton, 2007, available from: http://www.turbulence.org/Works/net_sonification/ [Accessed 17 May 2011].
- [10] Java, 2011, available from: www.oracle.com [Accessed 5 March 2011].
- [11] Transmission, 2011, available from: <http://www.transmissionbt.com/> [Accessed 5 March 2011].
- [12] Jpcap, 2011, available from: <http://netresearch.ics.uci.edu/kfujii/Jpcap/doc/> [Accessed 5 March 2011].
- [13] GeoLite, 2011, available from: <http://www.maxmind.com/app/geolitecity> [Accessed 5 March 2011].
- [14] Processing, 2011, available from: <http://processing.org/> [Accessed 5 March 2011].
- [15] SuperCollider, 2010, available from: <http://supercollider.sourceforge.net/> [Accessed 2 May 2010].
- [16] M. Wright, 2002, open sound control 1.0 specification. Available from: <http://opensoundcontrol.org/spec-1.0> [Accessed 2 May 2010].
- [17] LAME, 2011, available from: <http://lame.sourceforge.net/> [Accessed 5 March 2011].
- [18] I. Xenakis, *Formalized Music: Thought and Mathematics in Composition (Harmonologia Series, No 6)*. Pendragon Pr., 2001.

A. GRAPHICS EXAMPLE



Screen capture from a performance of *Leech*.

SONIK SPRING

Tomás Henriques

Music Department - Buffalo State College, NY USA
henriqjt@buffalostate.edu

ABSTRACT

This paper presents a new digital musical instrument that focuses on the issue of *feedback* in interface design as a condition to achieve a highly responsive and highly expressive performance tool. The Sonik Spring emphasizes the relationship between kinesthetic feedback and sound production while linking visual and gestural motion to the auditory experience and musical outcome. The interface consists of a 15-inch spring that is held and controlled using both hands. The spring exhibits unique stiffness and flexibility characteristics that allow many degrees of variation of its shape and length. The design of the instrument is described and its unique features discussed. Three distinct performance modes are also detailed highlighting the instrument's expressive potential and wide range functionality.

Keywords

Kinesthetic and visual feedback. Gestural control of sound. Interface for Sound and Music.

1. INTRODUCTION

A spring can be considered a universal symbol for oscillatory motion and vibration. Its simplicity and powerfulness stems from being a tangible object whose shape, length, motion and especially vibrating kinetic energy, can be easily *felt* and modified through simple hand manipulation. This is clearly understood when one thinks about toy-like devices based on a coil, such as the immensely popular SLINKY™.

Throughout time, philosophers and composers have been fascinated by the direct relationship between sound and vibration. K. Stockhausen whose musical works often include devices that link the boundaries of pitch, rhythm and vibration, took this discussion to a greater level, speaking eloquently about vibration as the common denominator of all things in the universe and relating sound to life itself [1].

Building a new instrument whose interface is simultaneously the symbol of vibration and the actual mechanism that triggers the production and modification of sound was thus very appealing.

Copyright: © 2011 Tomás Henriques et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The Sonik Spring uses primarily kinesthetic feedback as means of transmitting cognitive input to its user. Because it is handheld and controlled by spatial and gestural motions of the arms, wrists and fingers, the interface provides many degrees of complex muscular response and sensory stimulation.

One of the most common and pertinent criticisms about the performance capabilities and expressive potential of new electronic music instruments has been their lack of feedback response, frequently of kinesthetic nature. This insufficiency lessens the musical experience and hinders the new instrument from attaining the status of a “real,” acoustic-like, performance savvy instrument [2] [3] [4].



Figure 1. The Sonik Spring

The Sonik Spring was built from the ground up with the goal of creating an instrument that offers full immediate kinesthetic feedback. This is accomplished by virtue of the coil's resistance, which directly offers a strong sense of connectedness with the interface. Holding and manipulating the Sonik Spring is meant to feel like holding and shaping sound with one's own hands! Much in the way a sculptor works, the player of the Sonik Spring massages the sound, making it a clay-like material that is in constant metamorphosis. The Sonik Spring takes an approach to sound production, sound processing and music performance that empowers a musician to fully control sound in real time.

2. RELATED WORK

Research in kinesthetic based perception reveals force feedback as a stimulus deeply grounded into the human cognitive system [5] [6]. In the recent past, efforts have been made to introduce force feedback into the realm of digital controllers. One of the earliest experiments was done by Michel Waisvisz with the Belly-Web, a wire lattice similar to a spider's web [7]. In this interface the user's simple and intuitive finger movements pushing on the wires is made to alter their tension, which is detected by resistive sensors. The resulting changes are then translated into a set of control variables. Another such

experiment was the Harmonic Driving, one of the controllers that was a part of the Brain Opera. It consisted of a large compression spring attached to a bike's steering gear, which was used to control/drive musical events [8]. The spring's bending angles are measured using capacitive sensors that detect the relative displacement between two adjacent coils while torsion is obtained with a potentiometer that rotates as a function of the relative angle between the top and bottom of the spring. More recently other controllers have been introduced that address the issue of force feedback, such as the Sonic Banana [9] and the G-Spring [10]. The Sonic Banana uses four bend sensors linearly attached to a 2-foot long flexible rubber tube. When bent it maps the data from the sensors to sound synthesis parameters. Due to the relative softness of the rubber tube this controller offers limited feedback when compared to the G-Spring, which measures bend as well. It features a heavy 25-inch close-coil expansion spring, and uses light-dependent resistors to measure the varying amount of light that slips through the coils as a function of the amount of bend. Variations in bend are then mapped to synthesis parameters.



Figure 2. Expanding the spring's length

Unlike the controllers above described, the Sonik Spring uses accelerometers and gyroscopes to measure complex spatial motion. As an interface, it physically offers greater flexibility since it can be compressed, expanded, twisted or bent, in any direction, allowing the user to combine different types of intricate manipulation. Also, because the Sonik Spring is portable, wireless and comfortably played/held using both hands, it allows a higher degree of control and it looks and feels like a performable, "human-scaled" instrument.

3. DESIGN

3.1 The Interface

Choosing a spring with the right force feedback resistance was paramount to this project. The goal was to get a spring that could be *both compressed and extended* and that could provide an ideal amount of force feedback pressure when changing its length. By ideal I mean a

feedback force that was strong, enabling the user to feel and "fight" the resistance offered by the spring, while at the same time, allowing it to be fully compressed and freely extended to various lengths.

The Sonik Spring features a coil with a diameter of 3 inches and an unstrained length of 15 inches. The spring is attached at both ends to hand controller units made out of plexiglass. These consist of circular shaped plates designed to being comfortably grasped while allowing the user's fingers to move freely. The plates connect to a structure that houses and conceals most of the electronic components. Each hand controller contains sensors that detect spatial motion in three dimensions as well as five push buttons.



Figure 3. Left hand controller, 4 push buttons displayed

The spring can be extended to a maximum length of 30 inches and compressed down to 7 inches when fully collapsed. It therefore allows a length variation ranging from approximately half its size to exactly twice the length. These proportions, covering a 4:1 ratio, prove to be uniquely useful and intuitive when applying mappings of the spring's varying length to simple linear changes in musical parameters that are perceptually immediate.



Fig. 4. LH unit: accelerometer, thumb and index switches

The relevance of a string with the characteristics described above, becomes apparent when one considers the possibility to not only compress and extend its length, but to be able to bend and twist it as well, and doing so simultaneously. This remarkable flexibility allows the user to perform many different types of shape and length manipulations that can be mapped to sound and music parameters.

Working in tandem with the primary kinesthetic feedback of the spring is the important visual feedback component [11] [12], directly linking the amount of force exerted on the coil with a gestural/spatial representation of that effort. This dual quality emphasizes the uniqueness of the interface.

3.2 Sensing complex motion

The Sonik Spring senses variations in spatial motion and orientation using a combination of accelerometers and gyroscopes. Three groups of 2-axis accelerometers coupled with 1-axis gyroscopes were devised and placed in three strategic locations within the interface: one group at each end of the spring and one group at its exact middle. This is so to fully capture the very many possibilities of spatial motion, especially those related to various types of torsion and bending. Variations of motion in the lateral, longitudinal and vertical angles of rotation will be described in this paper using the terms pitch, roll and yaw, borrowed from flight dynamics.

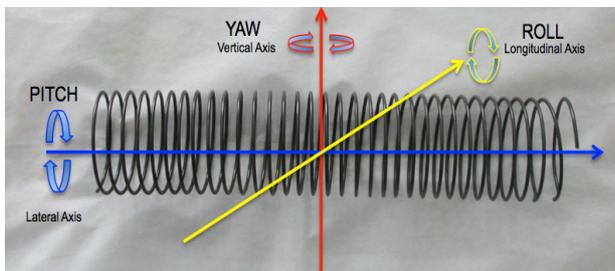


Figure 5. Spring's three axes of rotation

The simplest way to explore changes in the spring's orientation is accomplished by using *both hands* synchronously to perform the *very same* type of *wrist driven* rotating actions, doing so for each one of the 3-spatial dimensions. In this scenario the sensors at both ends and middle of the spring would have similar readings since they would be moving in exact parallel motion. If on the contrary, a performer bends and twists each hand independently, using different force amounts, such as shown in figure 6, complex shapes in the spring are created requiring all sensing elements to be separately analyzed. In this case, the fluidity of the spring's shape makes the acquisition of sensor data to have to rely on the combined result of their readings.

Changes in the spring's length are measured using the data from one axis of a small joystick. The joystick is built into the right hand controller and its shaft is

connected to a long necked hook, attached to a nearby and carefully chosen ring of the spring. When the spring changes its length, that ring along with all others gets displaced, and the distance it covers drags the shaft with it giving an accurate measurement of the spring's overall change in length. This simple solution has proven to be very reliable for the purpose it serves unlike previous experiments done with different sensors. Those included an hall-effect sensor placed at one end of the spring and actuated by a small magnet attached to a nearby ring, and a 10-turn potentiometer attached to the right hand controller, driven by a retractable wire attached to the opposite end of the spring on the left hand unit.



Figure 6. Bending the interface in a complex way

The results of these experiments revealed to be impractical. The hall-effect sensor provided inconsistent readings and the retractable wire would occasionally get entangled in the rings of the spring.



Figure 7. Measuring length variation with a joystick

The hand controller units contain five push buttons each. They are strategically placed for the fingers to rest comfortably on them. Each button is meant to be triggered by a single specific finger. One of the major roles of the buttons consists in enabling or muting the readings of the spatial sensors allowing the data to be properly routed and processed.

3.3 Sonik-Spring: A Two-Spring Mass System

The Sonik Spring is most often used pushing or pulling both ends in opposite directions, continuously varying the distance between them. Conversely, the user can manipulate the spring by keeping both arms at the very same distance while rotating the interface within the three spatial axes. But there is yet another way to explore the unique physics of this interface, given the particularities of its construction.

Since a group of sensors were placed in the center of the spring they make up for a small weight behaving as a mass in a classic spring-mass system. This arrangement offers the possibility to generate oscillatory motion of this center mass by shaking the spring either longitudinally or transversely, with different force amounts, and whilst keeping both arms/hand units at the same distance.

In the Sonik Spring the center weight acts upon both halves of the spring, turning the interface into a two-spring mass system, with both halves having similar spring constants. Figure 9 shows the housing of those sensors and also depicts a group of 10 rings that were compressed and linked together so as to mechanically facilitate to secure the sensors in place, thus further contributing to the definition of a center mass.

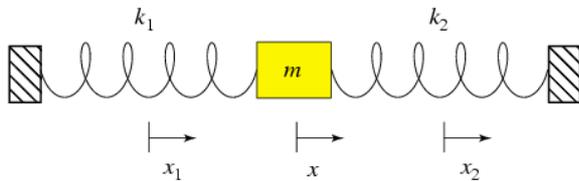


Figure 8. Two-spring mass system

When the mass m is displaced by a distance x , it makes the “first” spring to lengthen by a distance x_1 (pulling with a force in the $-x$ direction) while the “second” spring is compressed by a distance x_2 (pushing with the same force in the $-x$ direction too). Knowing that both halves of the Sonik Spring share the same spring constant, that is, $k_1=k_2$ with the amount of extension x_1 equaling the compression x_2 , the equation of motion and the frequency of the mass oscillation can be calculated as follows:

$$ma = F \quad ma = -kx$$

$$ma = -k_1x - k_2x = -(k_1 + k_2)x$$

$$k_1 = k_2$$

$$ma = -2kx$$

$$a = -(2kx)/m$$

$$\omega = \sqrt{2k/m}$$

$$T = 2\pi\sqrt{m/2k} \Rightarrow f = 1/2\pi\sqrt{m/2k}$$

The accelerometer and gyroscope placed in the center of the spring are used to measure the rate of oscillation of the mass of the system. The displacement of this mass and the cyclic way the rings compress and extend is visually very apparent. This quality suits the interface to being used rhythmically, in a very tangible way, to generate events such as short percussive sounds, etc, whose nature can be made to evolve as a function of the oscillatory energy of the interface. The rate of oscillation can also be mapped to more subtle parameters such as the frequency of an oscillator driving an amplitude modulation algorithm, etc.



Figure 9. Accelerometer and gyro at coil’s middle point

3.4 Channeling the sensor data

The Sonik Spring uses a MIDItron wireless sensor interface to collect the information acquired by the ten analog sensors and ten digital switches [13]. The analog sensor data is formatted as MIDI continuous controller messages and the on off states of the switches as MIDI note on and note off messages. This information is sent to a computer running the MaxMSP software which does all the data processing. Working with a wireless sensor interface has proven to be invaluable since it allows the spring to be completely and freely manipulated.

4. PLAYING THE SONIK SPRING

The Sonik Spring can be used in different ways. Three relevant ‘performance modes’ have been identified. These are: Instrument mode, Sound Processing Mode and Cognitive Mode.

4.1 Instrument Mode

In “*Instrument mode*” the Sonik-Spring is played as a virtual concertina, using the gestural motions commonly associated with playing this instrument while adding new performance nuances unique to the physical characteristics of the spring. In its current implementation the instrument can either use a MaxMSP patch that controls the generation of sounds based on a physical model of an air-driven vibrating reed [14] [15] [16], or it can process the sensor data sending it via MIDI to commercial hardware and software synthesizers.

To play the Sonik Spring the performer holds it horizontally, with both hands, comfortably grabbing the instrument. The sensors of the left hand unit trigger the generation of chords while those of the right hand generate melodic material.

The motion of pulling and pushing the spring emulates the presses and draws of virtual bellows using the tone generation technique of an English concertina. The amplitude of those gestures is mapped to the loudness of the sound.

The accelerometer and the five push buttons of the right hand unit are combined to generate the melodic material. This is accomplished using fingers index through pinky, to access 4 buttons that borrow the pitch generating method of a 4-valve brass instrument, allowing the production of the 12 chromatic tones within an octave. Changing the springs’ “pitch” by rotating it in the lateral plane maps the accelerometer data to select the desired pitch-octave, triggered by pushing the button assigned to the right hand thumb. A total of 6 octaves can be comfortably selected. Melodically, the Sonik Spring can thus simulate an instrument with 72 air-blown free reeds.

The loudness of the tones produced by the instrument is a function of both the absolute length of the spring as well as the amount of acceleration force exerted to make that length change from its previous position. The rate (speed and acceleration) at which the length changes is given by the joystick’s displacement and by the combined data from the three accelerometers, being assigned to changes in loudness using different mapping strategies [17]. A crescendo is achieved by continuously pulling the spring outward. A diminuendo is done with the opposite action. A sudden and strong pull or push on the spring translates into a loud sound, etc. Furthermore, notes played in staccato are triggered by pairs of short bursts of pushes/pulls of the spring while legato notes are obtained by keeping the spring still lengthwise, and changing notes with the buttons of the right hand.

Pitch bend and glissandi effects are also possible by mapping changes in “roll” and “yaw” using the right hand’s accelerometer and gyroscope, respectively.



Figure 10. Spring fully collapsed

Pitch bend and glissandi effects are also possible by mapping changes in “roll” and “yaw” using the right hand’s accelerometer and gyroscope, respectively.

Chords are generated using the five push buttons, the accelerometer and the gyroscope of the left hand controller. The software that generates the chords is largely based on the author’s previous work implemented in the wind controller META-EVI [18]. Chords can have anywhere from 0 to 4 notes. This allows the muting of the harmonic functions or use the left hand controller as a simple drone or counterpoint line if the number of chord voices is just one.

The type of harmonies that can be played depends upon the choice of a target ‘home-key’ gotten from a combination of four push buttons (using again the 4-valve brass technique) to select one of twelve different pitches and the button for the thumb to select minor or major mode. Once these choices are made, the very same four buttons select the ‘scale degrees’, which provide different chord types. Since chord types are software dependent it is possible to chose ‘non-tonal’ chords from a large array of options if desired. Chord inversion is implemented by mapping variations in the amount of “roll” of the left hand controller. Changes in chord voicing varying the register of the chord’s notes, is implemented by mapping changes in the “yaw” position. The overall loudness of the chords is mapped to the “pitch” position of the left hand controller.

As far as changes in the timbre of the sound produced by the physical model, they are obtained by mapping a series of gestural motions into synthesis and control

parameters. A vocabulary of a small group of such gestures has been implemented and it has proven to be a simple and effective way to correlate visual to auditory information [19] [20].

- a) Twisting the hand units symmetrically in opposite directions and with the same force to map changes to Filter Cutoff frequency
- b) Twisting the hand units symmetrically in opposite directions while bending the spring down to map both filter cutoff *and* resonance
- c) Bending the spring so that it defines a “U” shape mapping that shape to LFO rate, acting on the pitch being played
- d) Bending the spring so that it defines an inverted “U” shape, mapping it to LFO amplitude
- e) Shaking the interface along its lateral axis to map oscillation of the center mass to the frequency of an oscillator doing amplitude modulation

4.2 Sample Processing Mode

The Sonik-Spring can be used as a controller for real-time sound processing. In its current implementation the software uses a granular synthesis engine to playback and process sounds stored in memory [21]. The many degrees of gestural motion that the interface offers, allows the performer to convey a strong connection between the actions taken on the spring and the auditory outcome on the sound being processed in real time.

Mapping the variation of the length of the spring to different parameters, switchable using push button presses on the right hand controller, achieve the best results as far as the correspondence between the auditory and visual domains. The most striking use of the length variation is to map it to classic pitch transposition where both pitch and tempo are simultaneously altered. Holding the sound playback and performing scrubbing effects, forward or backwards, on a short section of a sound, by extending and compressing the spring, is also perceptually rewarding. Mappings of the left hand accelerometer include the independent control of a sound’s pitch and playback speed by respectively varying the spring’s lateral and longitudinal axial rotations, that is, its ‘pitch’ and its roll. The gyroscope of the left hand controller, detecting the spring’s yaw, is used to perform panning changes on the sound being processed.

The switches of the right hand are use to perform tape-like “transport functions”. Therefore sounds can be triggered forward or backwards, stopped, paused, muted and can be looped. It is also possible to choose variable loop points and isolate a chunk of an audio file anywhere within its length, with the capability to trigger the loop start point at will thus creating rhythmic effects.

The sensors of the right hand are used to perform additional functions such as control grain duration and randomize playback position. They are also used to

control parameters that perform amplitude modulation and filtering on the samples.



Figure 11. Spring bent downwards – Inverted U shape

4.3 Cognitive Mode

An interesting use of the Sonik Spring is as a tool to test different sensorial stimuli. At an immediate and simple level, it can be used to gauge an individual’s upper limbs muscle and force responsiveness by directly linking variations in a sound’s parameter such as pitch or loudness, to variations of the spring’s length. A more complex approach to study an individual’s level of cognitive perception can be done by simultaneously linking auditory, visual, spatial and force feedback. This last scenario is especially promising to medically assess people with neurological challenges [22].

5. CONCLUSIONS AND FUTURE WORK

The Sonik Spring has proved to be a very versatile instrument and an interface that it is a lot of *fun* to play with. People of different ages and with different musical backgrounds have tried it and the results show that the *Sample Processing Mode* is by far the most popular performance mode.

Using the instrument as a virtual concertina is also musically rewarding. The interface is agile, responsive and highly expressive allowing the user to develop performance skills that could reach virtuosity.

A growing interest in the use of the interface in Cognitive Mode is also evident. Collaborations with researchers in the medical field are planned.

Future work will focus on taking advantage of combining and networking the data from all sensors so as to apply “many-to-one” mapping strategies. This will reveal new meaningful information, useful for the control of synthesis parameters when the instrument is being played with a physical model, increasing the high level of feedback that it already conveys. More research is also

planned to continue exploring the two-spring mass system. Of relevant interest is the inclusion of user generated oscillatory motion to affect synthesis parameters of the physical model being used to generate sound.

6. ACKNOWLEDGMENTS

This research is made possible through the support of the Portuguese Foundation for Science and Technology (grant UTAustin/0052/2008) and the UT Austin | Portugal Program in Digital Media. I also would like to thank my father, Vitorino Henriques, for his craftsmanship and dedicated help building the hardware.

7. REFERENCES

- [1] J. Harvey. *The music of Stockhausen: An introduction*. University of California Press, 1975.
- [2] P. Cook. *Principles for designing computer music controllers*. In Proceedings of the New Interfaces for Musical Expression Workshop, 2001.
- [3] Marcelo M. Wanderley and Nicola Orio. 2002. *Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI*. Computer Music Journal, vol. 26, number 3, pp. 62-76.
- [4] S. O'Modhrain. *Playing By Feel: Incorporating Haptic Feedback into Computer-Based Musical Instruments*. Ph.D. diss., Stanford University, 2000.
- [5] FJ Clark and K W Horch. *Kinesthesia*. In Handbook of Perception and Human Performance. Vol. 1, Sensory Processes and Perception. ed: KR Boff, L Kaufman, JP Thomas. 1986: Wiley & Sons, NY.
- [6] LA Jones, *Perception of Force and Weight: Theory and Research*. Psychological Bulletin 1986, Vol. 100, No. 1, pp. 29-42.
- [7] <http://www.crackle.org/Waisvisz%27%20Small%20Web%20%28Belly%20Web%29.htm>
- [8] J. Paradiso. *The brain opera technology: New instruments and gestural sensors for musical interaction and performance*. Journal of New Music Research, 28(2): 130–149, 1999.
- [9] E. Singer. *Sonic banana: A novel bend-sensor-based MIDI controller*. In Proceedings of the International Conference on New Interfaces for Musical Expression pages 85-88, 2006.
- [10] D. Lebel and J. Malloch: *The G-Spring Controller*. In Proceedings of the International Conference on New Interfaces for Musical Expression, pp. 220-221, 2003.
- [11] J. Davidson. *Visual perception of performance manner in the movements of solo musicians*. Psychology of Music, 21:103–113, 1993.
- [12] Cadoz and M. Wanderley. *Gesture-music*. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pages 71–93. IRCAM – Centre Pompidou, Paris, 2000.
- [13] <http://www.eroktronix.com/>
- [14] J. Cottingham. *The Motion of Air-Driven Free Reeds*. In Collected Papers of the 137th Meeting of the Acoustical Society of America, 1999.
- [15] L Millot, V. Debut. *Time Domain Simulation of the Diatonic Harmonica*. In Mosart Workshop on Current Research Directions in Computer Music, Barcelona Spain, 2001.
- [16] D. Howard, S. Rimell, A. Hunt. *Force Feedback Gesture Controlled Physical Modeling Synthesis*. In Proceedings of NIME, NIME-03, McGill University - Montreal, Canada, May 22-24, 2003.
- [17] A. Hunt, M. Wanderley, M. Paradis. *The importance of parameter mapping in electronic instrument design*. In Proceedings of NIME, NIME-02, Dublin, Ireland, May 24- 26, 2002.
- [18] T. Henriques. *Meta-EVI: Innovative Performance Paths with a Wind Controller*. In Proceedings of NIME, NIME-08, Genoa, Italy, June 2008.
- [19] C. Cadoz. *Instrumental gesture and musical composition*. In Proceedings of ICMC 1988, pp.1-12.
- [20] A. Mulder. *Toward a choice of gestural constraints for instrumental performers*. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pp. 315–335. IRCAM – Centre Pompidou, Paris, 2000.
- [21] A. Gadd and S. Fels. *MetaMuse: A Novel Control Metaphor for Granular Synthesis*. In Proceedings ACM Conference on Computer Human Interaction. SigCHI, ACM, 2002.
- [22] B. Wen. *Multisensory integration of visual and auditory motion*. In Neuroscience. Issue: May, pp. 1-6, 2005.

ISOMORPHIC TESSELLATIONS FOR MUSICAL KEYBOARDS

Steven Maupin
Faculty of Engineering
University of Regina
maupin2s@uregina.ca

David Gerhard
Department of Computer Science
Department of Music
University of Regina
gerhard@cs.uregina.ca

Brett Park
Department of Computer Science
University of Regina
park111b@cs.uregina.ca

ABSTRACT

Many traditional and new musical instruments make use of an isomorphic note layout across a uniform planar tessellation. Recently, a number of hexagonal isomorphic keyboards have become available commercially. Each such keyboard or interface uses a single specific layout for notes, with specific justifications as to why this or that layout is better. This paper is an exploration of all possible note layouts on isomorphic tessellations. We begin with an investigation and proof of isomorphism in the two regular planar tessellations (Square and hexagonal), we describe the history and current practice of isomorphic note layouts from traditional stringed instruments to commercial hex keyboards and virtual keyboards available on tablet computers, and we investigate the complete space of such layouts, evaluating the existing popular layouts and proposing a set of new layouts which are optimized for specific musical tasks.

1. INTRODUCTION

Musical instruments vary in the techniques for providing access to all pitches, but a common method is to provide direct access to each note in a grid or scale. This is the technique for keyboard instruments like the piano, where all available notes are laid out on a linear scale from lowest to highest. While natural and intuitive, this layout has a number of drawbacks: Harmonic relationships between notes are not immediately obvious, and because of the presence of accidentals, scales and chords in each key are played differently meaning that as students learn the piano, they must re-learn scales and chords for each key. Stringed instruments are similar in that for each string all notes are laid out in a linear scale. This is a physical constraint of the instrument, and may in some way have influenced the linear layout of the keyboard instrument. Adjacent strings are related harmonically, which can improve the playability and understanding of harmonic relationships within the scale. Aside from the guitar, which we will consider later, stringed instruments are in fact *isomorphic*, in that harmonic relationships between notes have the same shape regardless of the key in which the note, chord, or scale

is played. A perfect fifth is always a single position over or seven positions along a string. This is not the case for the piano, since although all harmonic relationships are the same number of semitones regardless of the key, the shape of these semitones is obfuscated by the location of white and black notes on the keyboard.

Throughout this paper, we will be showing a variety of note layouts in different forms. Since black notes can be spelled more than one way depending on the key and context of the note (*e.g.* $C\sharp = D\flat$), and since keyboards do not inherently have a correct spelling for black notes, we will be variously labelling them as sharps, flats, or some combination of the two.

It should also be noted that the musical systems considered in this paper are predominantly western, in that we are not considering microtonality, and we are concentrating on equal-tempered scales.

2. ISOMORPHISM IN MUSICAL INSTRUMENTS

An isomorph is any object that exhibits a uniform shape or structure. Isomorphism applied to musical instruments means that every distinct musical performance is executed in the same way, regardless of key or location. An isomorphic musical keyboard consists of an array of note-controlling elements on which any given sequence and/or combination of musical intervals has the “same shape” on the keyboard wherever it occurs. An example of an isomorphic but non-keyboard musical instrument is a bass guitar. Each string is tuned to an equal interval apart, that of a perfect fourth. With this tuning scheme, all major chords share a common form, as all minor scales are equivalent in structure.

2.1 Uniform tiling of regular polygons

In geometry, a uniform tiling is a tessellation of the plane by regular polygon faces with the restriction that all vertices are identical. Each vertex must be surrounded by the same kinds of face in the same order, and with the same angles between corresponding faces.

There are three regular tessellations that can be formed on a plane. They are built upon three regular polygons: the square, the hexagon, and the equilateral triangle. Of the three tessellations, only the square and hexagon tilings are unidirectional. Equilateral triangle tilings are inherently bidirectional, in that, if you have a triangle that is pointing up, you require a triangle that is also pointing down

in order to form a complete tiling. The bi-directionality of triangular tiling violates true isomorphism, as identical chords will appear to have different shapes due to the directionality component of the triangles. It is this feature of triangular tiling that nullifies its use in isomorphic keyboards.

It should be noted here that we are assuming directional homogeneity in these layouts. In other words, if one direction (e.g. south) is assigned to a particular interval (e.g. a descending fifth) then regardless of the starting note, one tile south is always a descending fifth. This constraint makes the layouts easier to learn and guarantees isomorphism in chord-shapes and scale-shapes, in that a player can begin at any note, and a set of tiles in a specific shape will produce the same chord regardless of the starting note. In the future, we may expand the investigation to non-homogeneous layouts, where for example notes in a single direction may alternate or even follow a pattern. It is conceivable that pseudo-isomorphism may be possible with directional non-homogeneity, but this warrants further study.

2.2 Square Tessellations

Most stringed instruments use a square tessellation as the basis of tuning. The horizontal direction is always chromatic, where each fret represents a semi-tone. The vertical direction is often tuned to a perfect fourth or perfect fifth. Figure 1 shows the square tessellation of the violin.

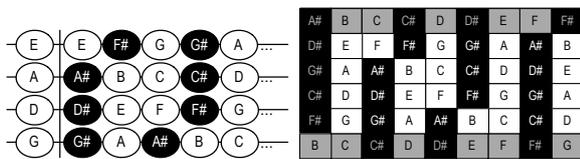


Figure 1. Violin layout and generalized tessellation

There are variations, however. The standard tuning for guitar is not completely isomorphic because the fifth string, B, is tuned one semi-tone less. This is to allow for six-string chords, commonly known as barre chords. The purpose of barre chords is another kind of isomorphism—a player can use a barre chord at any fret and produce the corresponding chord. In this case, a non-isomorphic tuning scheme is adopted to allow for more comfortable, moveable isomorphic chords. The general formula for calculating the musical pattern of squares is based upon the intervals in the vertical and horizontal directions. As stated, it is common that one of the two directions will have intervals of semi-tones. Incorporating the chromatic scale in one of the two directions ensures that all notes are available.

2.3 Hexagon Tessellations

There are comparatively fewer musical instruments that use a hexagonal tiling as opposed to square tiling. The general formula for calculating the musical pattern of hexagons is based upon the intervals in the vertical and horizontal directions. Figure 2 shows the two basic forms of the hexagon tessellation.

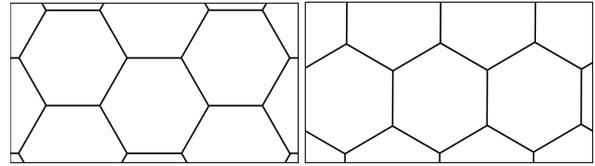


Figure 2. The two forms of hexagon tessellations: vertical (left) and horizontal (right)

In the vertical shape, there is an immediate vertical interval, but no immediate horizontal. In the horizontal shape, the opposite is true. The two differing shapes provide distinct advantages. Like the square tessellations, it is common that either the vertical or horizontal direction will have intervals of semi-tones, although this is not always the case.

3. A GENERALIZED THEORY OF MUSICAL ISOMORPHIC KEYBOARDS

With the two forms of tessellations, we can see that there are constraints on the available intervals in any given direction. This section will show that in both square and hexagonal layouts, all possible layouts can be represented by a horizontal interval and a vertical interval.

3.1 Square Layouts

As noted earlier, intervals in one direction are constrained to be equal, so that the isomorphism is guaranteed. This also means that we constrain intervals in one direction to be the opposite of intervals in the other direction. Upward intervals are the opposite of downward intervals, and left and right intervals are also constrained to be opposites. If we use $+V$ to indicate the upward or vertical interval, in semitones, and $+H$ to indicate the right or horizontal interval, we can see from Fig. 3 that these two intervals fully define the notes in a square tessellation. Starting at a note U , any closed loop of note transitions will come back to the original note.

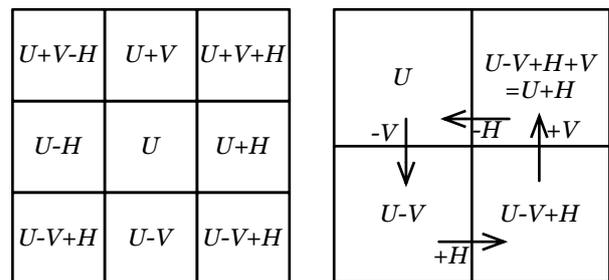


Figure 3. Generalized form of the square isomorphic tessellation. Any closed loop path will return to the original note.

3.2 Hexagonal Layouts

The hexagonal layout can similarly be represented by a vertical and a horizontal interval, although to show this we must do some analysis. Figure 4 shows the orientation of

the horizontal and vertical intervals. We also have another interval, A , which interrupts the line between H and V . All intervals are relative to U .

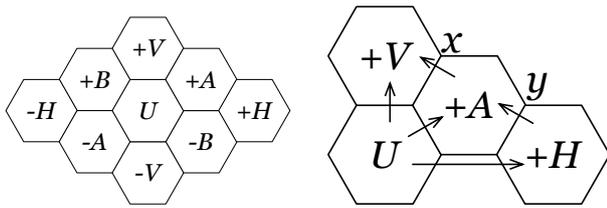


Figure 4. The triangle of interest in the vertical layout

For our layout to work, any closed circuit must bring us back to the original layout. If we consider the triangle of interest shown in Fig. 4, there are two triangles which include the still unknown interval A (assuming we have defined V and H). These triangular circuits are: $U + A + x = U + V$ and $U + H + y = U + A$. We can simplify these equations in two ways. First, by removing U from each side we are dealing with the intervals themselves, unrelated to any specific root note. This further emphasizes the isomorphic nature of these layouts. The second simplification is to note that if intervals in the same direction must be the same, then $x = y$. Therefore $A + x = V$ and $H + x = A$. If we solve first for x and substitute, we see that $H + (V - A) = A$, and further that $A = (H + V)/2$. This proves that all notes on a hexagonal grid can be defined purely by the horizontal and vertical intervals as indicated in Fig. 4. Further, it shows that if we have two notes (H, V) in a line separated by a single note A , the interval between H and V must be an even number of semitones. The same proof can be done for the B interval shown in Fig. 4, as well as with the inverse hexagonal layout, although in this case H would share a vertex with U , and V would be two notes away.

These generalizations will be used to analyze a number of historical examples of isomorphic musical interfaces, before exploring the complete space and performing a theoretical analysis of which particular layouts may be better for which musical purposes.

4. HISTORY OF ISOMORPHIC MUSICAL KEYBOARDS

Although stringed instruments can be considered to use a square tessellation layout, they are not usually considered a musical keyboard as such. There are few square-tessellation musical keyboards available. You can commonly find midi-controllers that form a square lattice but these are usually used for drum machine, music sequencing, or other rhythmic purposes. There is no standard in square grid musical keyboards and it is still generally underdeveloped. Hexagon keyboards seem to have caught on a lot faster than square keyboards, as there are several variations of them on the market today.

Hexagonal isomorphic keyboards in a musical instrument probably stem from the early 19th century accordion or concertina. The accordion utilized an offset grid of circular buttons which were organized on a hexagon lattice. In

the later 19th century, Paul von Jankó reinvented the musical layout of a piano[1], by creating a multi-row keyed instrument (Fig. 5), which consists of alternating rows of black and white keys.

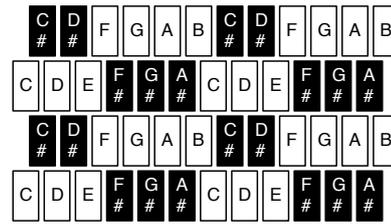


Fig. 7.

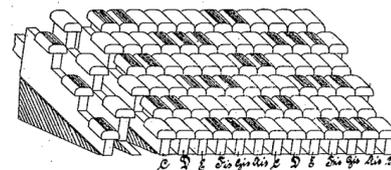


Figure 5. Jankó layout and original patent figure

It was the first isomorphic piano-like keyboard. This layout is commonly known as the Jankó layout, and can be defined as $H=2, V=0$. Today, there are several commercially available hexagon keyboards: the C-ThruAXiS-49, the Opal Chameleon, and the ThumMusic System “Thummer” [2]. The AXiS-49 and Opal Chameleon keyboards use the Harmonic Table note layout ($V=7, H=1$). Although the Thummer is not currently in production, it is filed under several patents and uses the Wicki-Hayden layout ($V=12, H=2$).



Figure 6. The thummer and the axis49, commercially available isomorphic hexagonal keyboards

The Harmonic Table note layout has been well known since the 18th century. Leonhard Euler had developed a tone matrix which he called the Tonnetz [3] and it is found to be equivalent to the harmonic table. The harmonic table integrates a perfect fifth interval in the vertical direction with a semi-tone interval in the horizontal direction. The harmonic table is below:

The Wicki-Hayden layout is the arrangement that is commonly used on accordions and concertinas, shown in Fig. 7. It was first conceived by Kaspar Wicki and a variant was patented by Hayden in 1896 [4]. The Wicki-Hayden layout integrates octaves in the vertical direction, and whole-tone intervals in the horizontal direction. It uses the “inverse” hexagon arrangement.

Interestingly, the standard computer keyboard has also been used to experiment with hexagonal layouts. Because

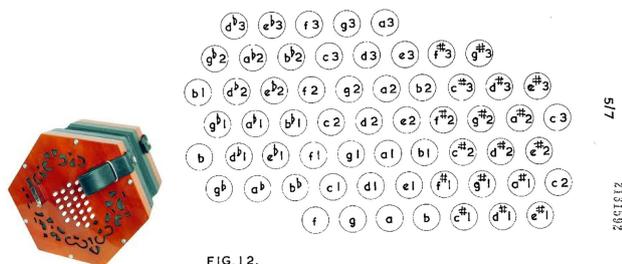


FIG. 12.

Figure 7. A concertina, and Hayden’s original patent figure for the Wicki-Hayden layout

each row of keys is offset by about half a key from the previous row, the arrangement is the same as the horizontal hex layout and it is possible to simulate, for example, the Wicki-Hayden layout by assigning each key to a separate note ($\boxed{F}=b^b$; $\boxed{G}=c$; $\boxed{H}=d$; $\boxed{T}=f$; $\boxed{Y}=g$ etc.).

Apart from hardware adaptations of hexagonal music systems, there exists software for Apple products which provide the same layout. Michael Eskin has developed the mJammer, iJammer, and HexJam applications, all of which utilize the Wicki-Hayden layout [5].

Musix, developed by Shiverware interactive (launched by two of the authors of this paper), is a fully-customizable multiple-layout isomorphic keyboard instrument. With the Musix software, over a thousand unique hexagonal isomorphic layouts can be created. Several novel isomorphic schemes have been discovered through the use of the Musix software, and testing of the usability of these layouts is planned for future work.

5. MELODIC AND HARMONIC ANALYSIS

The square and hexagon isomorphic grids will be analyzed in terms of: (1) melodic ability and (2) harmonic ability.

5.1 Melodic Ability

A melodic line is a succession of notes forming a distinctive sequence. This sequence can form either a scale or a tune (a melody). In either case, it is a musical phrase that develops horizontally, one note at a time. In music theory, there are two basic progressions that are used to characterize scales—the diatonic and the chromatic.

The diatonic scale is a seven note octave-repeating musical scale comprising five whole steps and two half steps for each octave. The “white-keys” on a piano form a diatonic scale, and can be used to construct the major scale and its seven derived modes, including the natural minor scale. The diatonic scale forms the foundation of music as we know it today. The chromatic scale is composed of twelve equally spaced pitches, each a semi-tone apart. It contains all of the notes within one octave. It has no root, or tonic, due to the symmetry of its equally spaced tones. The chromatic scale contains the twelve tones that are repeated every octave.

Most melodies are composed of mostly diatonic scales with chromatic or “accidental” inflections. By observing the ease of playing both diatonic and chromatic scales, one

can gain a sense of the level of melodic ability supported by each isomorphic layout.

5.2 Harmonic Ability

If melody is said to be the horizontal aspect of music, then harmony is the vertical aspect. Harmony is the instantaneous use of two or more simultaneous tones. All chords are in fact a musical harmony. There are infinite amounts of harmonies that can be achieved through any number of discrete tones. In music theory, there are several harmonic combinations that are found in the music of all ages. These are the octave, the dominant triad, and the dominant seventh.

The octave is an interval between two musical notes where one of which has twice the pitch of the other. They are perceived to be the same note. It is usually considered the most consonant harmony, besides perfect unison. The dominant triad is a chord composed of the root note, the diatonic third and the perfect fifth above it. The third can be either major or minor, depending on the leading tone of the scale. The root and perfect fifth do not vary. There are three versions of the triad. Using C major as an example, we have the dominant root position of C–E–G, the first inversion E–G–C, and the second inversion G–C–E. The dominant triad is the most commonly found chord in music. The dominant seventh is a chord composed of the root note, major third, perfect fifth, and minor seventh. The dominant seventh is found almost as often as the dominant triad. It is also one of the most fundamental four-note chords. By observing the ease of playing the octave, the major and minor dominant triads, and the dominant seventh chord, one can gain a sense of the harmonic ability provided by each isomorphic layout.

6. SQUARE LAYOUTS

Examples of the square layouts are shown in Fig. 8. Although there are many possible layouts, some are *degenerate*, meaning not every note is present. We restrict these layouts to only non-degenerate cases.

6.1 Semi-tone, Semi-tone (V: +1 H: +1)

Melodic: For the experimental composer who wants complete control of the chromatic scale. The diatonic scale can be played on the positive diagonal slope or along one semi-tone axis. The dual-linearity makes it easy to play scales in both the vertical and horizontal axis. Notes are repeated on the negative diagonal slope. There is no inverse for this tessellation.

Harmonic: Close harmonies, such as triads, are in range when played diagonally. Triads may be difficult to play on one axis due to the spacing of the notes. The dominant seventh is even more difficult to play due to note spacing. Discordant harmonies are abounding with close clustering. The additional octave may also be difficult to play with one hand, depending on the size of the squares.

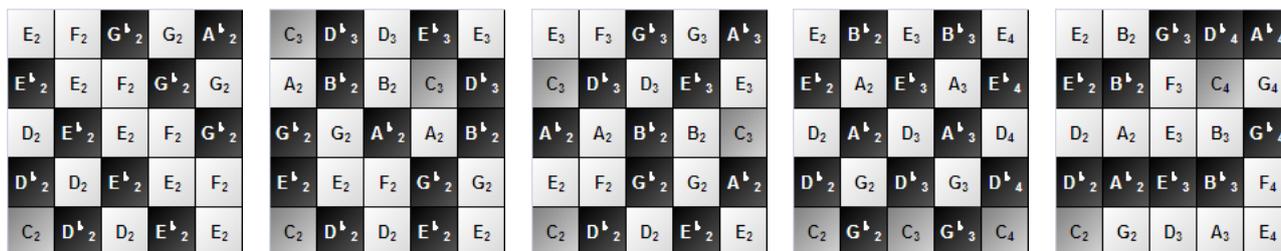


Figure 8. Examples of square layouts: +1+1; +3+1; +4+1; +1+6; +1+7;

6.2 Whole-tone, Semi-tone (V: +2 H: +1)

Melodic: The accidentals are grouped together on the whole-tone axis, which makes the diatonic scale easy to locate and perform. On the inversed matrix, the layout partially mimics a piano as the Db is located above and to the left of the D. This is useful for the composer who wishes to work with purely whole-tone and semi-tone scales. The chromatic scale is easy to play in the original layout, as a horizontal approach is taken to play the scales. Conversely, the diatonic scale is easier to play in the inverse layout as it is more compact but yet still played horizontally.

Harmonic: On both layouts, chords requiring the octave need to be performed diagonally. Triads are easily located but there seems to be no ideal fingering. The dominant seventh is not difficult to play. The inversed layout may provide slightly better fingerings for chords as the slope is less inclined than the original arrangement.

6.3 Minor Third, Semi-tone (V: +3 H: +1)

Melodic: The diatonic scale can be played a number of ways in both the original and inversed layout. There seems to be no best approach to play the scale. The chromatic scale is easily accomplished by playing three notes in the semi-tone axis, then moving up to the minor third interval. Scales are easier to play in the original layout because a horizontal approach is taken.

Harmonic: The inversed arrangement is the ideal square layout for harmonic combination. Major and minor triads are immediately at hand. The octave is readily found on the minor third axis, and the pattern repeats every four minor third intervals. The dominant seventh chord is easy to play. All harmonic combinations lend themselves well with to the inversed layout, because it assumes a horizontal approach will be taken to playing the chords. In the original layout, chords are more difficult to play because the hand is forced to play the chords in the vertical direction, sideways.

6.4 Major Third, Semi-tone (V: +4 H: +1)

Melodic: The diatonic scale can be completed in three three-note movements. The chromatic scale is played four-notes at a time, before moving up the major third interval. Both scales are easier to play in the original layout.

Harmonic: The location of the perfect fifth and diatonic thirds from the root makes this layout less ideal for harmonic combination. Triads are in reach but there appears to be no ideal fingering for them. The octave is found on

the major third axis, and the pattern repeats every three major third intervals. The dominant seventh chord requires more difficult fingering.

6.5 Perfect Fourth, Semi-tone (V: +5 H: +1)

Melodic: The original layout is ideal for playing and constructing melody. Nearly every scale contains the perfect fourth, so this layout lends itself tremendously well for the playing of scales. This arrangement happens to be the same tuning layout used by many stringed instruments. The diatonic scale is extremely easy to play. Three notes are played for every perfect fourth interval. The chromatic scale can be played by playing five notes on the semi-tone axis, then moving up to the perfect fourth interval. The inversed layout is not as good for playing scales or melodies because it requires one to play on the vertical axis.

Harmonic: On the original layout, most combinations are quite easily played. The diatonic third is played on the semi-tone axis and the perfect fifth is played on the interval above. Directly above the perfect fifth is the octave, which makes the octave easy to add. The drawback is that the dominant seventh chord is challenging to play. In general, chords are more difficult to play on the inversed layout.

6.6 Tri-tone, Semi-tone (V: +6 H: +1)

Melodic: All scales are easier to play in the original layout because they are played horizontally. The diatonic scale can be completed in just two movements in the vertical direction. Alternatively, the diatonic can be played by three notes on the semi-tone axis, then moving up the diatonic interval and back one semi-tone. The chromatic scale requires six notes to be played on the horizontal before moving up the tri-tone interval. This implies that one finger may have to play two notes. Alternatively, five notes can be played and the pattern is moved back on semi-tone on the tri-tone interval above. This layout forms an interesting checkerboard pattern which may help to provide a new approach to melodic structures.

Harmonic: The original layout is better for constructing chords, as they can be played horizontally. Major and minor triads are easy to play when the perfect fifth is played on the interval above. The dominant seventh isn't difficult to play. However, the location of the octave makes it quite difficult to add an octave to any chord structure.

6.7 Perfect Fifth, Semi-tone (V: +7 H: +1)

Melodic: The original layout is better for the performance of melodies because it is done horizontally. The diatonic scale can be completed in two or three vertical intervals. Scales require a significant jump (5 tiles back and one tile up) to get from one row to the next, but historically this has not been a problem for violin, viola, cello and mandolin players, possibly because the physical distance from one “tile” to the next is small on these instruments. This may be why the double bass uses +5+1 instead of +7+1.

Harmonic: Once again, the original layout is better for harmonic combination because chords are played horizontally. The perfect fifth is considered the most consonant musical harmony besides unison and the octave, and for that reason it is commonly found in most chords. With this layout, the perfect fifth is located immediately above the selected note. If the root note and perfect fifth are played with one finger, triads and even the dominant seventh are quite easy to play. The location of the octave makes it difficult to include, however.

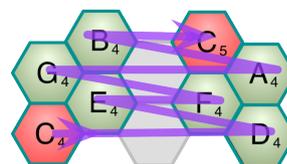


Figure 10. A major scale in the Harmonic layout

6.8 Octave, Semi-tone (V: +12 H: +1)

Melodic: This arrangement forms a completely chromatic interface. The diatonic scale and chromatic scale are to be completed on the semi-tone axis. The original layout is better for melodic ability as the scales are performed horizontally.

Harmonic: On both layouts, chords may be somewhat difficult to play due to the spacing between octaves. However, interesting chord voicings that span several octaves can be quite easily created.

7. HEXAGON LAYOUTS

Each hexagonal layout presented here uses a specific orientation of the hexagonal tessellation, as well as a specific vertical and horizontal interval. We also show the inverse of each layout, which corresponds to the same vertical and horizontal layout in the opposite hexagonal tessellation. We beg the reader’s indulgence in our naming of the three new proposed layouts, each was build separately by a contributor to the project, with specific musical intentions in mind. As with square layouts, we restrict the presentation here to “interesting”, non-degenerate cases.

Each layout is evaluated in terms of both Melodic and Harmonic features. As an example, a collection of triads in the Harmonic layout are presented in Fig. 9 and the major scale for the Harmonic layout is presented in Fig. 10.

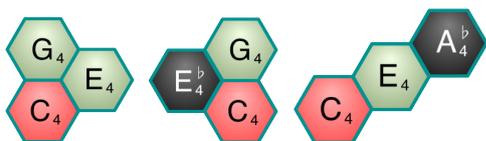


Figure 9. Triads for the Harmonic layout (see Fig. 12): Major, Minor, Augmented

7.1 Wicki–Hayden (V: +12 H: +2)

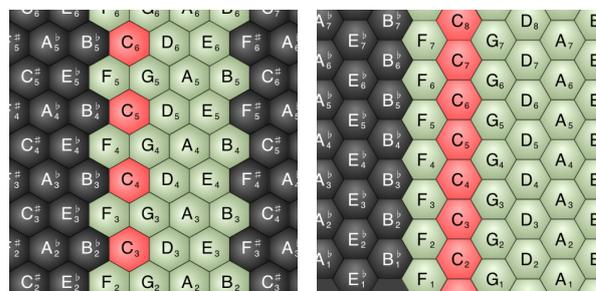


Figure 11. Wicki-Hayden (horizontal) layout: original, inverse

Melodic: The diatonic scale is intuitive to play on the original Wicki-Hayden layout. The white keys make up the center of the keyboard and the accidentals are on either side. The major scale takes three vertical movements to complete. The scales are played horizontally. Most intervals are whole steps which may make mistakes sound more consonant. The chromatic scale requires a more awkward movement, implying that non-diatonic scales and melodies containing accidentals will be a challenge to perform. On the inversed layout, both the diatonic and chromatic scales are more difficult to play, due to increased spacing between notes.

Harmonic: On the original, major and minor triads are easy to play, which is made possible by the immediate interval to the perfect fifth from the root. The fingering recommended for the triads is to play the root with the middle finger, the fifth with the ring finger, minor third with the pointer finger, and major third with the pinky. However, the location of the octave makes it difficult to add on to an existing chord. On the inversed layout, all of the chords are more difficult to play, due to increased spacing between notes. Only the original layout is the true Wicki-Hayden layout.

7.2 Harmonic Table (V: +7 H: +1)

Melodic: On the original, the diatonic scale is played in two-note movements on the horizontal axis, moving up the vertical as the scale ascends. Looking at the C in the bottom left corner, it is apparent that the major scale is played with high symmetry. The same is true for the inversed layout, although it is more condensed. Semi-tones are found in the horizontal axis, which makes the chromatic scale easy to play.

Harmonic: The original tessellation has interesting harmonic combinations. One may quickly notice that the ma-

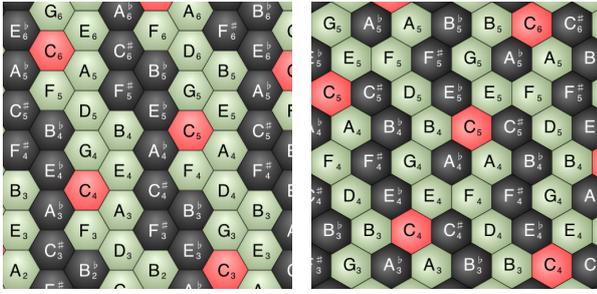


Figure 12. Harmonic layout (vertical) : original, inverse

Major and minor triads are formed by making small triangular clusters. This arrangement allows the playing of chords with only one finger. The dominant seventh chord is also easy to play. The octave spacing is close enough that one should be able to add it without much effort. This layout works extremely well for “jamming” as all the immediate surrounding notes are consonant. On the inversed tessellation, the harmonic combinations are not as compact—but this is not necessarily a bad thing. The triads can no longer be played with just one finger, but if one plays the perfect fifth that is a bit further to the right of the root, the chords are fairly comfortable to play. With this fingering, adding the octave to the chord is easily done. Performing the dominant seventh chord is more difficult.

7.3 Gerhard (V: +1 H: +7)

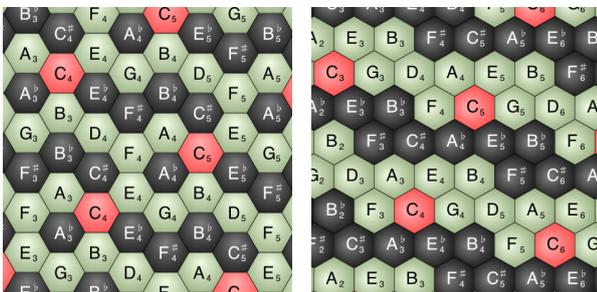


Figure 13. Gerhard layout (vertical): original, inverse

Melodic: This arrangement is similar to the Harmonic layout, its main difference being that semi-tone intervals take up the vertical axis where we had perfect fifths previously. The diatonic scale is easily completed by playing two notes in the vertical axis then moving to the right as the scale ascends. The major scale is again seen to be symmetrical. On the original tessellation, the scale is more compact. The downside is that there is no easy way to play the scales horizontally.

Harmonic: The original arrangement lends itself extremely well to major and minor triads. The major and minor thirds are adjacent and are located in between the root and perfect fifth. The dominant seventh is also in the vicinity. With this arrangement, adding the octave to the chord may be challenging. In the inverse arrangement, the triad chord shapes are triangular, as in the harmonic layout, and able to be played with one finger. The dominant seventh chord

is also easy to play, but adding the octave to the chord is difficult.

7.4 Park (V: +1 H: +5)

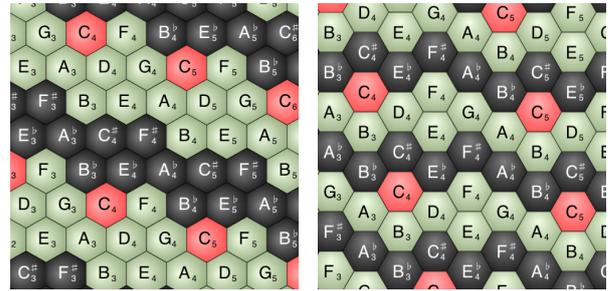


Figure 14. Park layout (horizontal) : original, inverse

Melodic: The diatonic scale is played on the negative slope. Semi-tones are in the vertical direction, which makes the chromatic scale easy to play. The inverse layout is superior because the diatonic scale is less sloped and more linear, and the chromatic notes are located directly above the root.

Harmonic: Major and minor triads are easy to play although their chord shapes are dissimilar. The dominant seventh is difficult to play. The location of the octave is convenient in that it can be added on without too much strain. In the original layout, the chords are more compact.

7.5 Maupin (V: +1 H: +3)

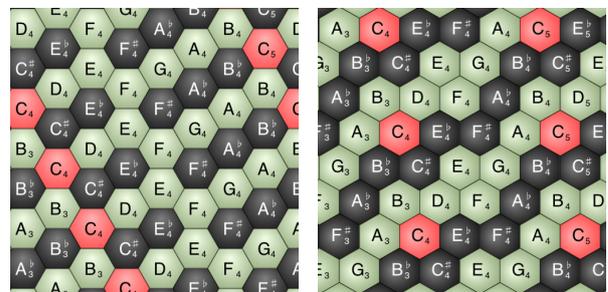


Figure 15. Maupin layout (vertical) : original, inverse

Melodic: The diatonic scale is played on the positive slope. Semi-tones are in the vertical direction, which makes the chromatic scale easy to play. In the inverse arrangement, it takes on a piano-like form. The inverse arrangement is superior because the diatonic scale is less sloped and more linear, and the semitones are located directly above the root.

Harmonic: In both arrangements, the triads require more of a stretch as compared to other layouts but may actually be more comfortable to play. The major third suggested is the one located on the diatonic scale, with the perfect fifth located beside it. The advantage is that the minor thirds are adjacent to the major thirds, which makes changing between major and minor chords easy. The added octave and dominant seventh chord are also surprisingly easy to exe-

cute in both layouts. The inverse arrangement offers chord shapes that are more comfortable to play.

8. ANALYSIS, CONCLUSIONS, AND FUTURE WORK

Of the square tessellation, we easily conclude that intervals of perfect fourths in the vertical axis and semi-tones in the horizontal axis provides the most advantageous approach to constructing and performing melody. There are several reasons for this, apart from its widespread use in stringed instruments. The diatonic scale forms a repeatable pattern when played three notes per every vertical jump. This layout also provides the ideal arrangement for the chromatic scale, as five notes per interval jump are required. This means the chromatic scale can be performed with one hand, quickly and easily, with no horizontal movement. Of the hexagonal tessellation, the harmonic table layout and the Maupin layout provide distinct advantages for melodic performance. The harmonic table layout excels in that the diatonic scale can be performed as two notes per every vertical jump. It requires little horizontal movement, and the pattern is highly symmetrical. The main advantage of the Maupin layout is that it appears to have the same layout found on a regular piano keyboard, albeit on a positive slope. The diatonic scale is easy to pick out although it requires more of a horizontal movement. The chromatic scale is easier to perform than the harmonic table layout because semi-tones are found immediately adjacent to the root note. This implies that accidental inflections would be significantly easier to perform.

Of the square tessellation, the inverted arrangement of semi-tones in the vertical direction, and minor third intervals in the horizontal direction provides the best layout for harmonic combination. This layout is the second layout presented in Fig. 8. The octave is located on the same axis as the root, which makes it easy to add to any harmonic combination. In between the octaves, the perfect fifth is found on the interval above. This is an ideal location of the perfect fifth, as the major thirds is found directly to the left, and the minor seventh to the right. This is the most comfortable way to play a dominant seventh chord on a square tessellation. In fact, all chords are found to be extremely comfortable to play with this tiling.

With the harmonic table layout, triads can be performed with one finger. It is most suited for “jammers”, a style of keyboard which ensure consonant notes are surrounding the root note. The dominant seventh is easy to play, and octave spacing is close. The downside to the harmonic layout is that harmonically complex chords are difficult to perform as all of the notes are clustered close together. The original Gerhard layout provides triads that are easy to execute, as the minor and major thirds are directly adjacent to each other. The dominant seventh chord and added octave are also easy to perform with this layout. However, as found with the harmonic table layout, the close clustering of these chords makes the hand bunch up when more harmonically complex chords are tried. With the Maupin layout, chords are much more spread out and are played similarly to that found on a piano. The minor and major

thirds are found immediately adjacent which simplifies the playing of triads. It takes on the same “comfortable” chord shapes that are found in the square tessellation of minor third intervals. The dominant seventh with added octave is found to be easily performed. The advantage of this layout is that the chordal note spacings are more spread out, and therefore feel more relaxed in technique.

We recognize that the analyses of the various layouts are at this point theoretical, so we plan to explore how individual players learn, play, and compose with these layouts. We will do controlled subject trials with both musicians and non-musicians, studying perceptual, musical, and bio-ergonomical considerations. Additionally, as discussed, we plan to study the potential pseudo-isomorphisms that may be present in any triangular or otherwise directionally non-homogenous layouts.

9. REFERENCES

- [1] P. von Jankó, “Neuerung an der unter no25282 patentirten kalviatur,” German patent office, Tech. Rep., 1885.
- [2] G. Paine, I. Stevenson, and A. Pearce, “The thummer mapping project (thump),” in *Proceedings of the 7th international conference on New interfaces for musical expression*, 2007, pp. 70–77. [Online]. Available: <http://doi.acm.org/10.1145/1279740.1279752>
- [3] L. Euler, “Tentamen novætheoriæmusicæex certissimis harmoniæprincipiis dilucide expositæ,” 1739.
- [4] B. Hayden, “Arrangements of notes on musical instruments no. gb2131592,” British Patent office, Tech. Rep., 1986.
- [5] The alternate keyboards website. [Online]. Available: <http://www.altkeyboards.com/>
- [6] D. Birnbaum, R. Fiebrink, J. Malloch, and M. M. Wanderley, “Towards a dimension space for musical devices,” in *Proceedings of the 2005 conference on New interfaces for musical expression*, 2005, pp. 192–195. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1085939.1085993>
- [7] T. Blaine, “The convergence of alternate controllers and musical interfaces in interactive entertainment,” in *Proceedings of the 2005 conference on New interfaces for musical expression*. National University of Singapore, 2005, pp. 27–33. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1085939.1085949>
- [8] A. Milne, W. Sethares, and J. Plamondon, “Isomorphic controllers and dynamic tuning: Invariant fingering over a tuning continuum,” *Comput. Music J.*, vol. 31, pp. 15–32, December 2007. [Online]. Available: <http://portal.acm.org.libproxy.uregina.ca:2048/citation.cfm?id=1326598.1326602>

IMPROVING THE EFFICIENCY OF OPEN SOUND CONTROL WITH COMPRESSED ADDRESS STRINGS

Jari Kleimola

Department of Signal Processing and Acoustics
Aalto University School of Electrical Engineering
Espoo, Finland
jari.kleimola@aalto.fi

Patrick J. McGlynn

Sound and Digital Music Technology Group
National University of Ireland, Maynooth
Co. Kildare, Ireland
patrick.j.mcglynn@nuim.ie

ABSTRACT

This paper introduces a technique that improves the efficiency of the Open Sound Control (OSC) communication protocol. The improvement is achieved by decoupling the user interface and the transmission layers of the protocol, thereby reducing the size of the transmitted data while simultaneously simplifying the receiving end parsing algorithm. The proposed method is fully compatible with the current OSC v1.1 specification. Three widely used OSC toolkits are modified so that existing applications are able to benefit from the improvement with minimal reimplementation efforts, and the practical applicability of the method is demonstrated using a multitouch-controlled audiovisual application. It was found that the required adjustments for the existing OSC toolkits and applications are minor, and that the intuitiveness of the OSC user interface layer is retained while communicating in a more efficient manner.

1. INTRODUCTION

Open Sound Control (OSC) [1], [2], [3] is a widely used content format for communicating between media-related applications. Its popularity can be attributed to the intuitive and extensible addressing scheme, which, together with the ability of describing typed parameter spaces, enables setups that may easily adapt to a wide variety of application scenarios. Furthermore, the unidirectional communication protocol simplifies the connection setup between OSC compliant end points.

In OSC, the transmitted information stream flowing between the end points, from an OSC producer/controller to the OSC receiver, is quantized into time-tagged frames. Each frame contains one or more OSC messages, which are described by a human-readable URL-style address part and a typed data vector (see Table 1). The data vector is further divided into type tags and the actual arguments carrying the instantaneous data values. The address part and the type tags are transmitted as strings, whereas the arguments are kept in binary format. The fields of the message are aligned on 4-byte boundaries.

Copyright: © 2011 Jari Kleimola et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

| Address part | Data vector | |
|-----------------|-------------|-----------|
| address | type tags | arguments |
| "/LPF/1/cutoff" | "f," | 0.50 |

Table 1. The structure of an OSC message.

Figure 1 shows an example communication sequence for a standard OSC v1.1 compliant parameter update procedure. The controller addresses a synthesizer parameter located at "/LPF/1/cutoff", sets its value to 0.50, and then immediately updates the value to 0.51. As can be seen, the controller simply needs to push the updated parameter values to the synthesizer, which parses the address string, and updates its data structures accordingly with the binary argument values. The communication language (i.e., the namespace, data types and value ranges of the arguments) is defined by the receiving end, while the controller is configured to speak in the same language. The benefits of OSC are clear: the communication language is intuitive, extensible, and the stateless communication paradigm results in simplified application models for both end point implementations.

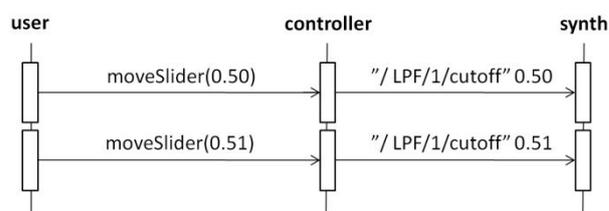


Figure 1. Standard OSC communication sequence.

The previous example also illustrates the inefficiency of the standard OSC implementations: although the human-readable addressing scheme is intuitive, long URL-style address strings waste communication bandwidth because the address part of the message is transmitted repeatedly with each parameter update. Furthermore, the receiving end needs to perform time-consuming string parsing operations for each received message. These points were criticized in [4], which also suggested that the inefficiencies might be eliminated if certain established IP-based techniques were adapted into the core of the OSC specification. However, the previous work did not elaborate how these techniques might be adapted in practice.

This paper explores the actual adaptation process of DNS-inspired [5] address mapping methods, and introduces an OSC v1.1 compliant address compression tech-

nique to improve the efficiency of OSC-based communication. The proposed method is especially useful for resource-constrained devices, where the CPU and power consumption efficiency is of great importance. It is also beneficial for systems where the physical connection between the end points is over a wireless medium, because increased network traffic has a tendency of raising the amount of lost packets.

The remainder of this paper is organized as follows. Section 2 introduces the OSC address compression technique, while Section 3 describes its implementation in three popular development environments and toolkits. Section 4 demonstrates the practical applicability of the approach using a multitouch-based interaction device in an audiovisual application setup. Section 5 provides discussion and suggestions that might be useful when speculating upon the next major OSC specification update. Finally, Section 6 concludes the paper.

2. OSC ADDRESS COMPRESSION

In order to improve the efficiency of the standard OSC communication mechanism, it seems beneficial to a) reduce the amount of transmitted data, and b) to simplify the parsing algorithm on the receiving end. This can be accomplished by compressing the potentially lengthy address string of the OSC message into an integer token, which is then transmitted in binary form inside the variable length data vector of the message, and used subsequently as the input for the receiving end parsing algorithm. The actual address part of the message is transmitted as a single character `"/`, denoting a compressed message, and at the same time ensuring backwards compatibility with the current OSC specification.

The internal server implementation of SuperCollider [6] utilizes a related method which is, however, incompatible with OSC and only supported by a few toolkit implementations. The technique proposed in this paper is fully OSC compliant, and therefore usable from any environment already supporting OSC. For example, a Pure Data [7] (Pd) patch running on the unmodified legacy OSC externals is able to take advantage of the compressed transmission stream, and base its computational logic on parsing the passed integer token. Another benefit of backwards compatibility is that the proposed method may be used concurrently with the standard OSC messaging mechanism. For instance, languages such as TUIO [8] may transmit their well-defined address strings as integer tokens, while still retaining user-extensibility: the receiving end parsing algorithm can easily differentiate between the compressed messages – which are transmitted using `"/` as the address string – from the standard uncompressed OSC messages transmitted with longer address strings.

Figure 2 shows the communication sequence of the previous example in the proposed streamlined form. The user is interacting with a universal OSC controller application, which has a slider widget in its graphical user interface. During the initial setup phase, the user links the slider with the synthesizer parameter (`"/LPF/1/cutoff"`). In

response, the controller application requests string-to-integer mapping information from the receiving end, and associates the slider with the supplied token value. The transformation from the URL-style address strings into the integer form tokens is thus carried out discreetly behind the scenes – the user of the system still views the address space of the receiving end in the intuitive string-form notation. Moreover, this procedure needs to be performed only once per session, as all slider movements are thereafter transmitted using the integer-based parameter addressing method.

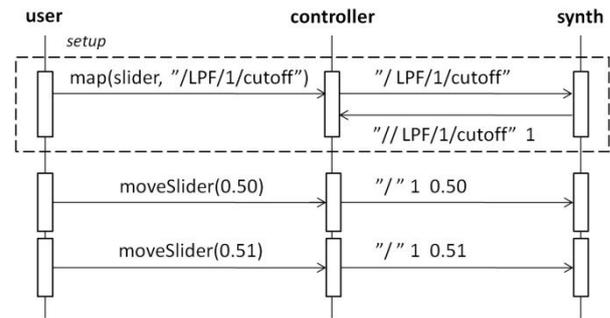


Figure 2. Proposed OSC communication sequence, showing one of the alternative mapping mechanisms within the dashed rectangle.

The reduction amount of the transmitted data depends on the complexity of the receiver defined namespace¹. The reduction ratio R between the proposed technique and the standard OSC implementation is given by expression $R = 1/\text{ceil}[0.25(L+1)]$, where L is the length of the address string of the standard OSC implementation. The reduction in the processing complexity of the parser implementation depends on the parsing algorithm, e.g., on the complexity of the hashing function. For a straightforward string comparison approach, the proposed method is substantially more efficient, as it has up to L times less comparisons than the standard OSC implementation, not counting the time wasted by non-matching string comparisons.

OSC also supports a wildcard-based parameter addressing mode, which enables control of multiple parameter values using a single message. The proposed method supports this by treating the integer token as a bitmask. To interpret the semantics of the integer parameter at the receiving end, the address string of the bitmask-based message is transmitted as `"/?"` instead of `"/` used in the normal one-to-one messages.

As shown in Figures 1 and 2, the cost of the improved efficiency is the more complicated setup phase. This is required because the controller needs to know the mapping between the string and the integer form representations. We considered two alternative mapping mechanisms: shared dictionaries [3] and request-reply mapping, as discussed in the following section.

¹ It is not uncommon to have address strings that are even longer than 20 characters.

2.1 Mapping Mechanisms

In the shared dictionary mapping mechanism, the receiver publishes its entire address space dictionary as a shared resource. This can be done by using a text-formatted file, where each line defines a single address string – integer pair, separated by a comma. Resource-constrained controller devices with fixed functionality may also opt to hard-code the transmitted integer tokens and leave the file-based mapping interpretation to the receiving end.

The setup phase of the shared dictionary mapping mechanism consists of searching the user-supplied address string from the first column of the file, and once it is found, reading the corresponding integer token value from the second column. The URL of the mapping file may be defined as a TXT record entry of a Zeroconf-based service discovery process [9]. If such technology is unavailable, the URL may be entered by the user during the setup phase, or be simply hard-coded in the controller code.

In the request-reply mapping mechanism (see Figure 2) the controller sends a request to the receiver, which replies with the integer token of the controller-supplied address string. The address string of the reply consists of the requested string prefixed with a slash. This allows non-blocking implementation at the controller end, because the controller is then able to associate the reply in an asynchronous manner.

Although this mechanism requires more complicated implementation than the shared dictionary-based alternative, it has the additional benefit of supporting the wildcard enhanced address string – bitmask mapping mode: the receiver is responsible of resolving the wildcard formatted address string into the returned bitmask value.

Following the best practices described in [3], the request-reply mechanism calls for a TCP-based connection between the controller and the receiver. Because the receiver may already use a TCP port for normal parameter updates, it needs to recognize a GET request from the normal OSC parameter updates, i.e., SET messages. For this, we propose that GET messages are sent with an empty data vector, so that the receiver end can simply test the number of arguments in the received message, and if it is zero, interpret the address string as a GET request. A similar approach has also been utilized in [10].

TCP connection enables also request-reply –based shared dictionary access: the controller may send the reserved pattern `"//*"` as the address string, which the receiver recognizes and replies by dumping the contents of the mapping file back to the controller.

2.2 Topological Considerations

The proposed method works in both distributed peer-to-peer and centralized network configurations. In the former approach the controller and the receiver are in direct connection with each other, as shown in Figure 2.

The centralized hub-based topology requires a dedicated service in the network, into which the receivers register, and into which the controllers send their parameter

updates. The hub is responsible for redirecting the received update messages to the registered receivers. Since the IP address:port of the controller-induced SET message is already pointing to the hub itself, the address string of the SET message should be prefixed with the name of the destination. This name should be the one that the receiver supplied when registering with the central OSC service.

The benefit of the centralized solution is that the address string – integer token mapping requests may be handled in a single location, simplifying the receiver end implementations. Another benefit is that the central hub may act as a router, which can map controller-centric parameter spaces (e.g., `/finger/1`) into the receiver end parameters, even providing simple transformations for the data values as is demonstrated in [11]. The hub may also be used to define OSC processing chains, where the outputs of distributed OSC modules are connected into the inputs of other OSC modules. For example, a gesture recognition module might receive raw data from a multi-touch controller, turn it into higher level gestural tokens, which are successively routed into another OSC processing module. However, the implementation of the centralized service is outside of the scope of this paper and left for future work. Instead, we will next look at three widely used OSC libraries, and describe how they can adapt the proposed address compression technique.

3. IMPLEMENTATIONS

Existing OSC applications are often built using third-party OSC toolkits. Accordingly it makes sense to look at how these commonly-used libraries need to be adjusted in order to benefit from the proposed technique. The existing applications can then adapt to the new method with minimal reimplementations efforts. This section looks at three open source implementations, and describes the required steps for their modification. The source code for these modifications is available at the accompanying website of this paper [12].

3.1 Implementation of the Proposed Technique

The proposed technique is implemented as a singleton *Streamliner* class, which holds two hashing tables for the string – integer mappings. The tables have identical content, but are indexed either with string or integer values. This enables fast conversion into the compressed representation (string-to-integer), or back to the standard OSC form (integer-to-string). The *Streamliner* class has two public methods for making these mapping conversions, and one public method for reading a file into the hashing tables.

Additionally, the message class implementation of the OSC toolkit needs to be subclassed by overriding the constructor in order to map the supplied address string into the integer token (using the *Streamliner* class). The constructor also inserts the mapped token into the argu-

ment vector of the OSC message, and sets the address string of the address-compressed message to "/".

The receiver code does not usually require additional modifications at the library level, because the toolkits are natively capable of extracting the first integer argument from the argument list of the received OSC message.

3.2 Processing and oscP5

Processing [13] is a Java-based environment for building interactive audiovisual applications. Its OSC implementation is based on the object oriented oscP5 library [14], which is straightforward to modify. The OscMessage class was subclassed with OscMessage2, and a singleton OscStreamliner class was implemented in Java. To take the advantage of the proposed method, existing applications simply need to define their OSC message objects as instances of OscMessage2 instead of OscMessage, as shown in Listing 1.

```

osc2 = new OscStreamliner("map.txt");
...
OscMessage2 msg = new OscMessage2("/lpf/cutoff");
msg.add( 0.50 );
oscP5.send(msg, destination);
    
```

Listing 1. Sending compressed messages in oscP5.

The oscP5 library routes all received OSC messages into the oscEvent(...) handler, which can then perform the parsing based on integer arithmetic, as shown in Listing 2. The received integer token can be converted back into the string form presentation, if so desired, using the OscStreamliner method osc2.address(iaddr). The OscPlugin mechanism of oscP5 was not addressed in this work.

```

void oscEvent(OscMessage msg) {
  int iaddr = msg.get(0).intValue();
  switch (iaddr) {
    case 1: ...
  }
}
    
```

Listing 2. Parsing compressed messages in oscP5.

3.3 oscpack

The oscpack library is a "set of C++ classes for packing and unpacking OSC packets" [15]. The required modifications included augmenting the library with the osc::Streamliner class, and modifying the code of a) BeginMessage class with a new constructor, taking an integer token argument, and b) streaming this token into the argument list of the OSC message at the end of the OutboundPacketStream << (BeginMessage) handler. Listing 3 shows an example of the streamlined oscpack sending procedure.

Although it is possible to hide the string-to-integer transformation of the address pattern entirely inside the BeginMessage class, as was done in the oscP5 adaptation, the form described below is more efficient in terms of CPU load.

```

osc::Streamliner osc2("map.txt");
...
osc::int32 token = osc2.iaddress("/lpf/1/cutoff");
p << osc::BeginBundleImmediate
  << osc::BeginMessage( token ) << (float)0.50
...
    
```

Listing 3. Sending compressed messages in oscpack.

The oscpack ReceivedMessage class supports the non-standard SuperCollider address patterns. In order to add additional support for the proposed method, the class was augmented with a method to extract the integer token from the received message. The parsing is simplified into the form shown in Listing 4. As in oscP5, the address pattern may be regenerated from the received integer token using osc2.address(iaddr) invocation, if needed.

```

try {
  osc::int32 token = m.iaddress();
  switch (token) {
    case 1: ...
  }
}
    
```

Listing 4. Parsing compressed messages in oscpack.

3.4 Pd and OSCx externals

The OSC implementation (OSCx) of Pd consists of sendOSC, dumpOSC and OSCroute externals, and of a static code library containing the actual OSC implementation. To retrofit the proposed method, the sendOSC external was augmented with two functions (i.e., one for generating the mapping dictionary, and another for the actual sending routine). In addition, the setmap message was added to the interface to initiate the setup phase mapping dictionary construction. This message has to be invoked from the Pd patch before the actual control session is started (see Figure 3).

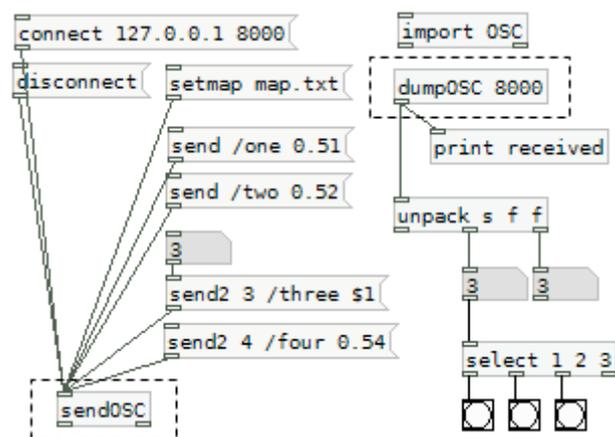


Figure 3. Streamlined Pd implementation. The OSC terminals are highlighted with dashed rectangles.

In the initial retrofit, the setmap message was the only modification visible to the user, as he still sends the OSC parameter updates using the original OSCx send message objects. However, because the external has to parse

the address string parameter of the send message for each invocation, a *send2* message was added to the interface as well. The parameters of *send2* are identical to those of *send*, except that the parameter list is prefixed with an additional integer, which is unique to all *send2* message boxes in the particular Pd patch (it does not need to match with the actually transmitted token value). The *sendOSC* external uses the prefixing integer to associate the message box instance with the actually transmitted integer token. This relieves the external from repeated string-form parsing actions, resorting to a simple integer-to-integer lookup for each sent message.

The parsing of the received OSC messages can be accomplished using the standard Pd *select* object. Thus, the more inefficient OSCroute becomes obsolete in this context.

4. EXAMPLE APPLICATION

The practical applicability of the proposed method is demonstrated by the following transparent example – wherein a touch screen is used to toggle playback of a selection of loops and apply some simple filtering. This demo uses the Pd implementation as described in Section 3.4 to both transform and distribute performance cues sent via a multi-touch application running on the Apple iPad.

In addition to the enhanced CPU efficiency, this approach reduces network traffic and hence the possibility of lost packets – which can especially cause concern in a live performance context that features wireless interfaces. In situations where the wireless transmission takes place on a mobile device, power consumption can also be reduced.

The example application can be conceptually split into three stages – the input stage, transformation stage, and the output stage, as shown in Figure 4.

Input stage: The iPad is running TUIOpad [16] – an open-source application which sends multi-touch data formatted using the TUIO protocol. This data is transmitted via a wireless connection to a laptop running a simple Processing [13] sketch, which extracts useful high-level information such as the number of fingers in contact and recognizes gestural cues. Relevant data is passed forward using typical hierarchical OSC namespaces (e.g., /touch/1).

Transformation stage: The resulting high-level messages are received by Pd, which is running a patch similar to the one illustrated in Figure 3. The augmented *sendOSC* function associates the incoming messages with a unique integer token according to the data in the mapping file. The data is now compressed – the new mapping procedure eliminates the need to repeatedly send address patterns.

Output stage: The compressed OSC messages control several looping buffers and band-pass filters in Pd. In addition to the audio control, a second Processing sketch generates a visual representation of the data. Processing is relieved of the burden of performing a string compari-

son every time an OSC message is received – a simple integer-recognition function will suffice. The visual accompaniment demonstrates the relatively concise data being parsed, in comparison with the lengthy address patterns that would otherwise be necessary.

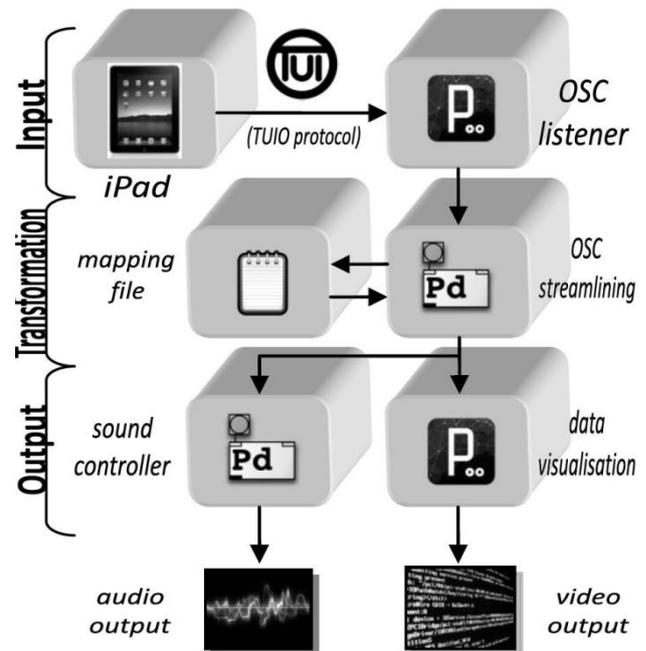


Figure 4. The data flow in the example application.

In this example, the proposed method has been demonstrated entirely within Pd in order to provide a lucid explanation. While performance issues are not necessarily of great concern in this kind of system, the benefits of implementing OSC compression at the ‘input stage’ – e.g., upon a mobile device sending compressed OSC with hardcoded integer tokens – is worthy of further investigation.

5. DISCUSSION

The proposed technique reduces the amount of transmitted data, simplifies parsing, and is straightforward to implement in existing OSC libraries and OSC compliant environments, as was demonstrated above. However, looking at the examples given in this paper, one might initially think that the technique flattens the intuitive tree-form address space of OSC into a continuous range of integers. This is not the case, because the receiving end – which defines how the integer tokens are mapped into the address strings – may construct the integer address space in a way similar to the IP addresses (e.g., 192.168.1.1). The 32-bit integer token can thus be interpreted hierarchically in a tree-form structure, and the widths of the bit fields are completely application-configurable. The hierarchical interpretation allows also wildcard-based mapping of the addresses, albeit consequently, it defines also a limit to the expansion of the address space (which is of no concern in the string-based representation). Since the

integer token is transmitted inside the data vector, and is therefore typed, it is possible to overcome this limitation by using 64-bit integers as address tokens instead of the 32-bit entity described earlier.

The downside of the proposed approach is that the message interpretation requires mapping information, and therefore the stateless presentation of OSC is not fully preserved. However, content formats with fixed address spaces may describe their a priori mapping definitions already at the specification level, which makes the messages self-contained, albeit still not truly stateless.

Section 2.1 discussed briefly the means of interpreting the received OSC message either as a SET or a GET request. Naturally, the interpretation should not be based on the amount of arguments in the message, but rather, the command verb should be included in the header part of the OSC message. Because RESTful paradigm has proved to be successful in many internet-based applications, the next major OSC specification update might consider redesigning the header part of the message accordingly. Based on our observations, a sufficient minimum set of verbs would consist of SET, GET, ADD and DELETE, possibly augmented with PUBLISH and SUBSCRIBE.

On the other hand, Zeroconf appears to be a viable solution for the service discovery and name resolution processes. In addition to working in a peer-to-peer fashion, it offers means of acquiring IP addresses without dedicated DHCP servers, which might be advantageous in distributed OSC setups. The TXT record field of the multicast DNS can also be used to describe the schema of the service, which, as discussed in Section 2.1, can be utilized to describe the string-to-integer mapping of the proposed technique.

XML-based schema language would offer advantages because it is extensible, standardized, and supported by a wide variety of libraries and related techniques (such as XPath, for example). However, since XML-based document parsing is a resource intensive process, a lighter format, such as JSON or OSC itself might prove to be more attractive for resource-constrained devices.

6. CONCLUSION

This paper proposed a technique that improves the efficiency of OSC-based communication. It decouples the user interface and the transmission layers of the protocol by mapping the string-form address representation into a corresponding integer form, which is then used in the compressed transmission stage. Three commonly used open source libraries were patched to take advantage of the proposed technique, and its practical applicability was demonstrated with an interactive audiovisual application. Finally, the paper concluded by providing suggestions for adapting certain established IP-based techniques within the OSC specification.

The OSC community is encouraged to explore the proposed technique in custom applications and OSC toolkits. We believe that the technique improves the efficiency of

OSC communication protocol in a transparent manner, but given the diversity of OSC-based application scenarios, its full potential is revealed after it is supported by a wider installation base. The source code for the discussed implementations is available at [12].

Acknowledgments

This work has been supported by the Academy of Finland (project no. 122815) and the National University of Ireland and Maynooth (John and Pat Hume Scholarship).

7. REFERENCES

- [1] M. Wright and A. Freed, "Open Sound Control: A New Protocol for Communicating with Sound Synthesizers," in *Proc. Int. Computer Music Conf. (ICMC'97)*, Thessaloniki, Greece, Sept. 25-30, 1997.
- [2] *The Open Sound Control 1.0 Specification*, 2002. http://opensoundcontrol.org/spec-1_0 (accessed 19.5.2011)
- [3] A. Freed and A. Schmeder, "Features and Future for Open Sound Control version 1.1 for NIME," in *Proc. 9th Int. Conf. New Interfaces for Musical Expression (NIME 2009)*, Pittsburgh, PA, USA, June 4-6, 2009.
- [4] A. Fraietta, "Open Sound Control: Constraints and Limitations," in *Proc. 8th Int. Conf. New Interfaces for Musical Expression (NIME 2008)*, Genova, Italy, June 5-7, 2008.
- [5] C. Liu and P. Albitz, "*DNS and BIND*," 5th ed., O'Reilly Media, 2006.
- [6] J. McCartney, "SuperCollider: a New Real Time Synthesis Language," in *Proc. Int. Computer Music Conf. (ICMC 1996)*, Hong Kong, China, Aug. 19-24, 1996.
- [7] M. Puckette, "*The Theory and Technique of Electronic Music*," World Scientific Press, River Edge, NJ, USA, 2007.
- [8] M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza, "TUIO - A Protocol for Table-Top Tangible User Interfaces," in *Proc. 6th Int. Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)*, Vannes, France, 2005.
- [9] D. Steinberg and S. Cheshire, "*Zero Configuration Networking: The Definitive Guide*," O'Reilly Media, 2005.
- [10] *oscit Homepage*, <http://lubyk.org/en/software/oscit> (accessed 19.5.2011).
- [11] J. Malloch, S. Sinclair, and M. Wanderley, "A Network-Based Framework for Collaborative Development and Performance of Digital Musical Instruments," *Lecture Notes in Computer Science*, Springer, 2008.

- [12] *Accompanying webpage of this paper*
<http://www.acoustics.hut.fi/go/smc2011-osc>
- [13] *Processing Homepage*, <http://processing.org/>
(accessed 19.5.2011).
- [14] *oscP5 Homepage*,
<http://www.sojamo.de/libraries/oscP5/> (accessed
19.5.2011).
- [15] *oscpack Homepage*,
<http://code.google.com/p/oscpack/> (accessed
19.5.2011).
- [16] *TUIOpad Homepage*,
<http://code.google.com/p/tuiopad/> (accessed
19.5.2011).

DYNAMIC INTERMEDIATE MODELS FOR AUDIOGRAPHIC SYNTHESIS

Vincent Goudard
LAM - Institut
Jean Le Rond d'Alembert
vincent@mazirkat.o
rg

Hugues Genevois
LAM - Institut
Jean Le Rond d'Alembert
genevois@lam.jussi
eu.fr

Émilien Ghomi
LRI – Université
Paris Sud 11
emilien.ghomi@lri.fr

Boris Doval
LAM – Institut
Jean Le Rond d'Alembert
boris.doval@upmc.fr

ABSTRACT

Mapping is one of the most important aspects of software instruments design. We call “mapping” the relation defined between the parameters from hardware interaction devices, and those of the process to be controlled. For software instruments, this relation between the user’s gestures and synthesis engine parameters has a decisive role in resulting ergonomics, playability and expressive possibilities of the system. The authors propose an approach based on a modular software design inspired by a multidisciplinary study of musical instruments and their playing.

In this paper, the concept of "Dynamic Intermediate Models" (DIM) is introduced as the centre of the proposed architecture. In such a scheme, DIM modules are inserted between the gestural interfaces and the audio-graphic synthesis and rendering engines. The concept of DIM is presented and explored as an extension of usual mapping functions, leading to an improvement of the interaction between the musician and his/her instrument.

Then, design and programming guidelines are presented, together with some concrete examples of DIMs that have been created and tested. Finally, the authors propose some directions to evaluate such DIMs in the architecture.

1. INTRODUCTION

1.1. Background

The invention of telephone and phonograph, almost contemporary to each other, disturbed our “traditional” relation to sound, voice and music, by allowing transmission and recording of sounds. From then, sounds could “travel” through space and time. Developments of electrical devices and later digital technology increased even more the “distance” between the body and the instrument in music production:

- electricity, by bringing new energy, previously mechanical, to machinery and instrumental devices;
- digital technology by operating a radical decoupling between musicians’ actions and effects on the instruments due to symbolic information encoding and processing.

The consequences of these decoupling on musician-instrument interactions have been studied extensively by Cadoz [3] in particular. Without repeating these discussions here, it is crucial to remember that these technologies induced a new definition of social value: from a world where work is highly regarded to a world based on information. Such an evolution has artistic, cultural and social irreversible consequences. Therefore, it is not surprising that, alongside these developments, new artistic sensibilities, new philosophical and scientific paradigms have been invented and experimented.

New questions and new musical devices emerged at the same time, mentioned by several computer music studies [1]. What is an instrument? What is inherent to its nature? How could we create repertoires if instruments keep evolving? How does an interface get its “instrumentality”? How to play yesterday’s, today’s, and tomorrow’s music with these new instruments ?

Usually, research on this topic mostly focuses either on the technical characteristics of the devices, on the software underlying them, or on the sensory-motor interaction between the musician and the “interface”. Indeed, technical aspects must be taken into account when studying the “instrumentality” of a new device but, as a counterpart, it can be useful and productive to work on a human-centred approach taking into account the cultural and social aspects of the interaction between some subjects and these new devices.

For acoustic instruments, these relations between input and output are complex and intricate, since they had been refined for centuries, in a co-construction between the morphology of the instrument, the ways of interacting with it, the available sound, and the associated repertoire. For digital sound engines, nothing is actually framed, everyone can code his own implementation of a synthesis algorithm, build his own device to control it, and have to draw links between data coming out from the interfaces and the synthesis parameters.

1.2. Instrumental interaction

In the field of Human-Computer Interaction (HCI), Instrumental Interaction is an operational interaction model introduced by Beaudouin-Lafon [2]. Within that definition, an instrument is a two-way transducer between the user and the object he wants to act on. For Cadoz, who specifically studied musical instruments, its role is to tackle the transformation from gesture to

sound in real-time. During the interaction between the musician and the instrument, “specific phenomena are produced, the behaviour and dynamic evolution of which can be mastered by the subject” [3]. Then, he identifies three types of instrumental gestures : excitation gestures, modification gestures and selection gestures.

But, as it has been pointed out by Cance et al. [4], defining precisely what is a musical instrument is a difficult task. Some of the acoustical instruments needed decades to become “instruments”. It depends on cultural and social aspects, since an instrument has not only a functional role, it has also a symbolic one. Then, instrumental quality of a device is not only depending on intrinsic qualities but constructed, through musical play, in a given cultural context.

1.3. Musical gestures, temporal and body scales

At first sight, it seems possible to distinguish the subsequent four phases in which a “musical” gesture is involved [9].

- Composition : production and writing of musical structures (not in real-time).
- Instrument making : building of the instrument and preparation in order to be played (tuning, equalisation, settings and adjustments).
- Playing : production of sounds in real-time. Often, only these actions are considered as musical gestures.
- Listening : perception and interpretation of the acoustic environment.

One can legitimately question the need to give to musical gesture such a broad definition. However, the validity of the categorisation presented above is not proved. The four phases are mostly overlaid, partially erasing the boundaries between categories of actions that seem a priori easy to state. Even when staying in a strictly classical scheme, the distinction established above is not as clear and obvious as it appears. In addition, through new forms of artistic creation (electroacoustic music, interactive music, etc.), the last decades have shown that this classification was fragile.

1.4. Origin of the concept of Dynamic Intermediate Model

For several decades, a lot of researches have focused on human-machine interfaces [2] and the optimisation of layers and mapping methods, although the exact contours and boundaries of the concept of mapping remains unclear. Thus, comparative studies on different types of mapping have been published [11], defining categories (one-to-one, one-to-many, etc.).

In general, we can consider that a classical, static, mapping is reducible to a matrix operation: multiplication of an input vector (the values directly or indirectly provided by the gestural controllers after linear or non-linear scaling) by a matrix describing the mapping function. This operation provides the vector used to generate the control parameters of the synthesis.

Then, the type and characteristics of the mapping is linked to the properties of the matrix. Is it symmetric, diagonal, triangular ?

Other approaches are possible, among which the methods based on gesture recognition techniques (neural networks, hidden Markov models ...) which do not need to explicit the link between gesture and synthesis.

In our case, we followed a different and complementary direction, assuming that a living and coherent synthesis requires too many parameters and evolves too rapidly to be entirely controlled by the musician.

2. DYNAMIC INTERMEDIATE MODELS

2.1. DIM characteristics

In recent years, research projects going in the same direction have been identified, sometimes in conjunction with convincing artistic achievements¹. This encouraged us to consider the generalisation of these ideas and to propose a completely modular software architecture.

The goals of using DIMs are of different natures and lead to the description of functional and structural aspects. Without being exhaustive, however, we discuss some of the essential characteristics they should show :

- to enrich the sensed gesture with modulations operating at frequencies beyond human gesture frequencies (e.g. fast bounces)
- to control multiple parameters from a single input gesture, since lively audio synthesis often requires more parameters than what one can manage in real-time
- to be easily integrated in a modular software architecture, allowing on-the-fly practice and instrument setting.
- to provide a range of modules corresponding to various needs related to the different gestures involved in music at different time scales (composition, setting, playing ...)
- to offer several levels of monitoring and drive simultaneously various synthesis and rendering engines with the same algorithms, in order to improve the coherency of their output
- to be bi-directional: the DIM should be able to communicate in either direction with the interface, other DIMs or the synthesis engine (cf. Fig. 1) in order to regulate the (non linear) interactions between the different stages.

In a metaphoric way, the role of a DIM in a synthesis process can be compared to the action of a bow, or a piano mechanism, etc. and the modular software architecture enables to strike a violin string with a hammer, or to bow a piano string.

¹ Notably, works by Jean-Michel Couturier [7], Cyril Henry [15] or Mathieu Chamagne.

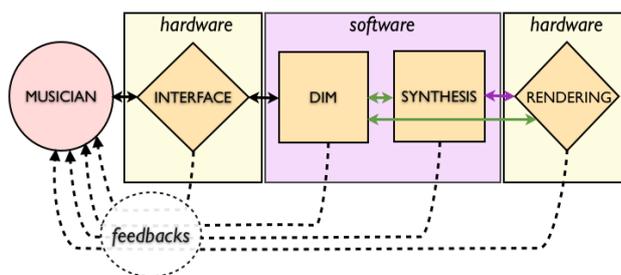


Figure 1. Overview of DIM mapping

2.2. DIM typology

As previously defined, the concept of Dynamic Intermediate Model is, by construction, very large. It must possibly be implemented in various musical tasks envisaged, which result in sound and musical processes at different scales, depending on musical task and gestures. For this purpose, and also in order to provide a wide range of interactions and processings, DIMs can be based on models developed in various scientific fields, enabling very different behaviours. In the OrJo Project², we initially focused on three types of DIMs :

- "physical" models
- "topological" models
- "genetic" models

2.3. DIMs as movement generators

The processes at work in a DIM operate on several aspects of the movement transduction expected in a musical instrument.

• quality of motion :

To give an acoustic example, a plectrum for a string instrument provides a particular pinch quality and its attack sounds different from that of the fingers. Also, the strings position on the instrument body allows to transform the seemingly linear movement of the hand in a variation of this particular movement, namely a succession of pinches which rhythm is related to the spacing of the strings. The interaction devices are often devoid of surface roughness (for instance a pen-tablet has not the roughness of the horsehair of a bow). A DIM will need to integrate this "surface aspect" of the virtual object.

• non-linear movements :

Richness of sound is partly related to the overall nonlinear phenomena in action in a musical instrument and contribute to make the sound rich and subtle. These non-linearities partly due to materials in acoustic instruments (and electronics) often lack in the standardised digital values of software instruments parameters. In the digital realm, non-linearities are actually often found in bugs and codec misuse, all kind of "glitches" that are precisely sought by a vein of

² OrJo is a research project funded by the FEDER and Ile-de-France Regional Council. It gathers the following companies and research labs : Puce Muse, LAM, LIMSI and 3DLized. It aims at developing audiovisual instruments for collective artistic performances.

electronic music now bearing this name, to produce rich sonorities and unexpected soundscapes [5]. The DIM should thus re-introduce saturation, exponential curves, distortion, and other jitter in the transfer functions of digital instruments.

• Augmented movement :

By acting on complex models, a one-dimensional variation can be converted into a multi-dimensional polyphonic movement. For example on a string instrument, the many notes of a chords can be played with one touch. This enhancement of movement may act in vertical (poly-phonic), horizontal (poly-rhythmic) dimensions, or on the many dimensions of timbre. We may for example control a "target parameter" that a set of elements would reach through a displacement-logic of their own.

2.4. Mental representation of DIMs

As a computer process, the algorithm may remain abstract for the musician who needs a mental model to predict the outcome of his/her action (ideally "sing" what he/she plays) and avoid the cognitive overload that involve the direct handling of too many parameters. However, this mental model is not so much an intellectual understanding of each of the processes at work, but rather a "body understanding" of the manipulated object, made up of the interaction device and the algorithms that are as one [18]. Rather than "input parameter", we thus prefer the term "handle", or "action item", to describe how we "catch" the virtual object being manipulated.

3. DIM IMPLEMENTATION

We present the implementation of two DIMs in the Meta-Mallette, trying to give a concrete -albeit limited-idea of the use of DIMs as a tool for experimental digital instruments making.

3.1. Meta-Mallette

Meta-Mallette³ is a software environment for playing computer-assisted music, images (and more) in real time [14]. It allows to load virtual instruments playable with joystick-like interfaces, pen-tablet or more expert interfaces such as the Meta-Instrument⁴.

Until now, the instruments developed for the Meta-Mallette directly included all parts of the whole process between the player and audio/graphics rendering. The latest Meta-Mallette version now enables to separate these various elements and reconnect them differently, opening the way for further experimentation, notably on mapping. The work carried out at LAM during the OrJo project precisely aims at developing elements that will ease empirical testing required for any music instrument design.

³ Meta-Mallette is available for download on Puce Muse website .

⁴ The Meta-Instrument developed at Puce Muse is pluggable in the Meta-Mallette, and offers 54 independent sensors sampled every 2ms.

3.2. DIM « Roulette » and « Verlet »

We will try to present an example of the use of DIM in the Meta-Mallette⁵. Let's suppose a setup consisting of a hardware interface, two cascaded DIMs and a Karplus-strong synthesis module.

3.2.1. DIM example 1 : Roulette



Figure 2. Roulette trajectories engendered by polygon motion.

The first DIM named "Roulette" is a geometric model inspired by Pascal's roulette⁶, a regular polygon that can move the following ways :

- tip on one of its corner
- sliding along one of its sides
- pendulum around an axis

These motion primitives have parametrized temporal evolution curves that will change the quality of induced movement. For example, the sliding motion will look like a fall, an burst, or a simple displacement according to the linear or quadratic speed.

Although resembling a physical model, this model differentiates itself by movements which time periods are tightly defined, allowing a rhythmic play to observe a beat. Moreover, the absence of pseudo-physical constraints allows to escape the apparent "rule" anytime.

The handles of the Roulette instrument are:

- the target's position that the polygon will follow
- the selection of movement type to be performed
- the diameter of the polygon and the number of sides
- its position and orientation
- the duration of each movement type

The reaction points (output parameters) of this DIM are:

- the diameter of the polygon and its number of faces
- its position and orientation
- the ongoing progression of movement
- the end of the movement

⁵ Interested readers will, however, better look into actual examples available on the LAM webpage. <http://www.lam.jussieu.fr/orjo/>

⁶ Roulette is a generalisation of the cycloid, named after the Treaty of Roulette written by Blaise Pascal in 1659.

3.2.2. DIM example 2 : Verlet

The 2nd DIM named "Verlet"⁷ is based on the pseudo physical Verlet algorithm. It can simulate a structure of points connected by elastic links. Here, this skeletal model is contained in a box with which it can collide.

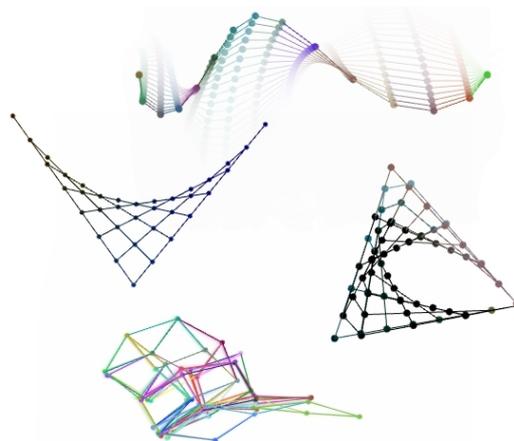


Figure 3. Various Verlet model states

The action points of the Verlet instrument are :

- the position is the orientation of the box
- the length and width of the box
- the length and stiffness of elastic links
- the size and direction of force applied to the model points

The reaction points of the "Verlet" DIM are :

- a matrix containing every point's position
- a matrix containing impact velocities

3.3. DIM interconnection

In the chain of interaction mentioned above, the player acts on the Roulette model which transforms the nature of his/her movement : by moving the Roulette's target (e.g. with a pen tablet), the model reacts with predictable behaviour, yet a behaviour of its own.

Its movement could be directly used to drive the sound synthesis algorithm. Yet, in this example, it is re-used here to drive a 2nd DIM : the position and orientation of the Roulette's polygon controls the position and orientation of the Verlet's box.

Thus, percussive movements caused by a rocking motion of the Roulette polygon generate a multitude of micro-movements generated by the Verlet algorithm "shaken" by the Roulette motion (fig. 4:1). The instrument-player can thus control with a simple gesture an intuitive process that complexify the (sensed) gesture and generates a complex set of movements.

⁷ This numerical integration scheme was developed in 1967 by physicist Loup Verlet. Andrew Benson made an implementation for Max/Jitter, available on <http://cycling74.com>.

Let's notice that the chaining of these two modules could be done in reverse order, i.e. by controlling a Verlet model which edges will act as a multitude of targets for the Roulette model (fig. 4:2). We could as well connect another Roulette model to the shaken Verlet (fig. 4:3), or make any other connection between DIMs, combining several movement scales.

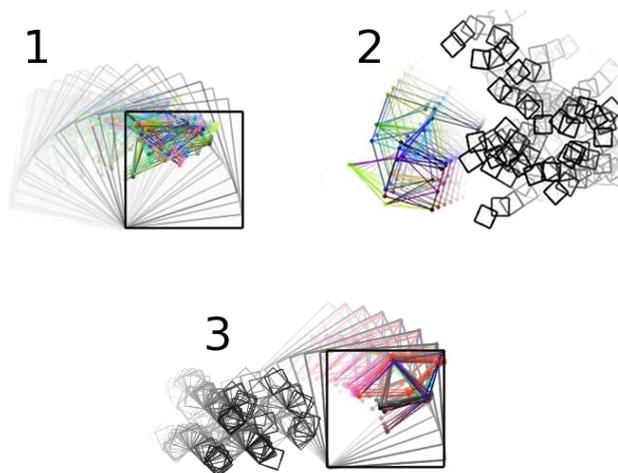


Figure 4. DIMs interconnections :

- 1 : Verlet edges controlling Roulette's targets,
- 2 : Roulette's polygon controlling Verlet's box,
- 3 : Roulette's polygon controlling Verlet's box, which edges control in turn other Roulette's targets.

3.4. Audio synthesis

To sonify the many sources of movement generated by the DIMs, we adapted known synthesis algorithms (Karplus-Strong, FM, granular) to enable the use of matrices to control many synthesis parameters in parallel on multiple voices.

For this purpose, we have jointly developed tools to map values between the input matrix and the synthesis parameters that allow various polyphony strategies (e.g. distribution, interpolation, per-voice detune) in case the number of movement sources and audio synthesis voices do not match. Conversely, various parameterized grids and functions allow ergonomic scaling between movement data and synthesis parameters, easing the task of using traditional music theory rules (tonality, rhythmic division).

The synthesis algorithm used in this example is based on the (plucked string) Karplus-Strong algorithm [12, 13]. We included a "make up gain" in the feedback loop to compensate for the damping function the fundamental frequency, thus allowing to set duration and harmonic filtering independently, and a non-linear pinch parameter which enrich the sound with transients and harmonics. The Meta-Mallette environment allow to plug the data output by the interaction device, the Verlet model, the Roulette model, or any combination of them. This choice of inputs combination to final parameters of

sound synthesis is an essential aspect of the adjustment required for the empirical instrument making process.

3.5. Graphical rendering

For now, we have focused our research on graphical representations that help understanding the model and mainly performing a monitoring task. Namely, the representation should help understanding the behaviour, predicting trajectories, by possibly showing the inner assembly. Indeed, DIMs are not necessarily intended to be viewed, especially if they are included in a larger whole. Moreover, there is more than one possible representation for a DIM. Modules providing the graphic synthesis have been separated from the algorithmic model in order to change the visual representation, or simply not use them.

As generators of movement, the output of a DIM may be used for purposes other than audio and graphic synthesis control. In particular, they can drive external elements such as acoustic devices they will excite, motors, light system, or to produce a force feedback. The possibilities are numerous and likely to exceed the framework of the developments undertaken in the current phase of our research.

4. CONCLUSION

The instrumentality of these new devices, as well as of "classical" instruments, does not result from their intrinsic properties only. It is constructed through music playing, interactions between musicians and the design and development of the instruments. The evolution and enrichment of the concept of instrumentality raised by these new practices imposes a pluri-disciplinary approach for the "science of instruments".

However, the OrJo Project, within which this research takes place, involves developing new instruments and also to gauge interest and richness, from an aesthetic and artistic point of view (creation and pedagogy) and as interaction models that could be generalised to other areas of Human-Computer Interaction. By the end of the project, one or more assessment methods should be developed.

4.1. Evaluation of the DIMs

If any scientific research needs to validate the assumptions made, if any engineering work involves assessing the outcome of developments, the task is not easy when one is interested in how a complex system (modular, half-material, semi-software) can become a musical instrument, that is to say considered as such [6].

Unlike many other devices and tools, musical instruments generally require a relatively long learning time. One could consider that this is a defect related to poor ergonomics... On the contrary, one could think that a long learning of the variety and subtlety of tones that can be produced makes the expressiveness of an instrument.

However, being able to choose more or less complex DIMs gives us hope that it is possible to fit the various artistic situations and educational purposes (discovery, learning, amateur orchestra, professional orchestra, etc.). In particular, we have seen considerable differences of assessment between soloist and orchestral practices in the Meta-Orchestra⁸. Interviews were led in a previous project, aiming at a better understanding of the concept of instrumentality with the help of psycho-linguistic methods [4]. This study should be deepened, and compared to works addressing similar problems [17].

Beyond the sound grammar proposed by Schaeffer [16], the Temporal Semiotic Units [8] proposed by the MIM⁹ seems to us an interesting tool for assessing the richness and ease to achieve such musical figures with a given instrument.

4.2. Limitations and perspectives

When asking a luthier on the importance of the wood, the glue, the varnish to make a good violin, he/she may answer the crucial factor is the tight and precise assembly of all the elements. The modular approach may encounter limitations when considering the importance of inter- and retro-action between the elements that make a good instrument.

Nevertheless, we believe the developments described in this article will be useful for experimentation. Their availability in an modular environment like the Meta-Mallette¹⁰ will hopefully encourage many digital instrument makers, researchers, composers and musicians to use them and further investigate this still unexplored field.

4.3. Acknowledgement

We would like to thank all partners involved in OrJo project : Puce Muse, LIMSI, and 3Dlized.

5. REFERENCES

- [1] Battier, M. « L'approche gestuelle dans l'histoire de la lutherie électronique. Étude de cas : le theremin », in *Les nouveaux gestes de la musique*, H. Genevois et R. de Vivo (eds). Éditions Parenthèses, 139-156, 1999.
- [2] Beaudouin-Lafon, M. « Moins d'interface pour plus d'interaction », *Interfaces Homme-Machine et Création Musicale*, H. Vinet et F. Delalande (eds), Hermès, 123-141, (1999).
- [3] Cadoz, C. « Musique, geste, technologie », *Les nouveaux gestes de la musique*, H. Genevois et R. de Vivo (eds), Éditions Parenthèses, 47-92, (1999).

⁸ The Meta-Orchestra is a live audiovisual electronic music orchestra initiated by Serge de Laubier in 2007. It relies on hardware interfaces and a software environment, the Meta-Mallette.

⁹ Laboratoire Musique et Informatique de Marseille, <http://www.labo-mim.org>

¹⁰ Developments led during the OrJo project will be downloadable from the "Meta-Library", a software library currently developed by Puce Muse and available winter 2012. <http://pucemuse.com>

- [4] Cance, C., Genevois, H., Dubois, D. « What is instrumentality in new digital musical devices? A contribution from cognitive linguistics and psychology », *Proceedings of CIM09*, to be published « La musique et ses instruments » (2011).
- [5] Cascone, K. « The Aesthetics of Failure: « Post-Digital » Tendencies in Contemporary Computer Music », *Computer Music Journal*, 24:4, pp. 12-18, The MIT Press, 2000
- [6] Castagne, N., Cadoz, C. « 10 criteria for evaluating physical modelling schemes ». in *DAFX'03: Proc. of the 2003 conf. on Digital Audio Effects*, 2003.
- [7] Couturier, J-M., *Utilisation avancée d'interfaces graphiques dans le contrôle gestuel de processus sonores*, thèse, Université de la Méditerranée, Marseille, 2004.
- [8] Frémiot, M., et al., *Les Unités Sémiotiques Temporelles – Éléments nouveaux d'analyse musicale*. Editions MIM, Document Musurgia, 1996.
- [9] Genevois, H., « Geste et pensée musicale : de l'outil à l'instrument », *Les nouveaux gestes de la musique*, H. Genevois et R. de Vivo (eds), Marseille : Éditions Parenthèses, 1999, p. 35-45
- [10] Ghomi, E. « Utilisation de modèles intermédiaires pour le mapping de paramètres de synthèse », master report ATIAM, (2006)
- [11] Hunt, A., Wanderley, M. M., Paradis, M. « The importance of parameter mapping in electronic instrument design ». in *NIME '02: Proc. of the 2002 conf. on New interfaces for musical expression*, pages 1-6, Singapore, Singapore, 2002. National University of Singapore
- [12] Jaffe, D. A., Smith, J. O., « Extensions of the Karplus-Strong plucked string algorithm, » *Computer Music J.*, vol. 7, no. 2, pp. 56-69, 1983.
- [13] Karplus, K., Strong, A., « Digital synthesis of plucked string and drum timbres, » *Computer Music J.*, vol. 7, no. 2, pp. 43-45, 1983.
- [14] de Laubier, S., Goudard, V. « Puce Muse - La Méta-Mallette », *Proceedings of Journées d'Informatique Musicale (JIM 2007)*, Lyon, 2007
- [15] Momeni, A., Henry, C. « Dynamic Independent Mapping Layers for Concurrent Control of Audio and Video Synthesis », *Computer Music Journal*, Spring 2006, Vol. 30, N°1, 49-66, (2006)
- [16] Schaeffer, P., *Traité des objets musicaux, essai interdisciplines, nouvelle édition*, Seuil, 1977
- [17] Stowell, D., Plumbley, M.D. & Bryan-Kinns, N. « Discourse analysis evaluation method for expressive musical interfaces », *Proceedings of New Interfaces for Musical Expression (NIME'08)*, Genova, 2008.
- [18] Varela, F. J., Thompson, E. T., and Rosch, E. *The Embodied Mind: Cognitive Science and Human Experience*. The MIT Press, 1991.

FROM SNOW [TO SPACE TO MOVEMENT] TO SOUND

Alexandros Kontogeorgakopoulos

Cardiff School of Art and Design,
University of Wales Institute, Cardiff

akontogeorgakopoulos@uwic.ac.uk

Olivia Kotsifa

Cardiff School of Art and Design, University of Wales Institute, Cardiff

okotsifa@uwic.ac.uk

Matthias Erichsen

Cardiff School of Art and Design, University of Wales Institute, Cardiff

merichsen@uwic.ac.uk

ABSTRACT

The current paper concerns a ‘work in progress’ research and design project regarding a forthcoming mixed media interactive performance, which integrates ‘space design’, sound, visuals and snowboarding. The aim is to create a playful and even provocative experience to the users/performers and to the spectators of the final event by mixing and blending music, sound design, architecture, visual projections and freestyle snowboarding. It is a collaborative effort between a French freestyle snowpark development, a snowboarding events company named H05, and three researchers and practitioners in computer music, architectural design and electronic engineering. Computer motion tracking techniques, a variety of spatial and body sensors and sonic transformations of pre-composed material have been and are currently explored for the realization of the musical part of the piece. The fundamental and key concept is to map sound features and interactively composed sound objects to snowboarding full body gestures. Architectural design plays a critical role in the project, since the composed space shapes the snowboarding movements, which then form the corresponding musical and visual elements that will be introduced to our work in the future. The current paper describes our initial designs and working prototypes used during a test period in the H05 snowparks in the Alps.

1. INTRODUCTION

Interactivity in art, design and performance is experiencing a growth in exploitation of late and is considered as a significant element in contemporary creative practice [1, 2, 3, 4]. An interesting example of a design field where novel paradigms of interactivity are constantly introduced is the design of new digital musical instruments [5]. Another example, which is equally relevant to our project, is the technologically mediated interactive dance/music systems and performances. [6, 7, 8]

In this paper, we are using technology to make a responsive environment for snowboarders. The users experience an immediate engagement with the space

through the sound that their own movements generate. Therefore, through exploration of their environment, they realize that their bodily effort and snowboarding tricks are integral to the composition of sounds and music. It is worth mentioning that so far, interactive performances in theatre and dance are widely known, but as far as the authors are aware of, there is no interactive performance related to snowboarding.

Our aim is to “control” music and visuals through movements and through freestyle snowboarding in specifically designed snowpark. Of course, mere control is not synonymous to interaction. According to Robert Wechsler “Interaction relates to spontaneity, openness and communication” [6]. Therefore, in our designs, we consider as interaction something more than using a designed physical and spatial interface from a number of snowboarders performing a standard show, event or a competition. We are designing an environment for creativity, exploration and play; not a task oriented interactive system.

This paper presents all the elements of the working process that will contribute towards the future realization of an interactive multimedia snowboarding event. It summarizes all the project phases; viz., conception, research, design, construction and testing. The first part explains the concept and covers the architectural aspects. The second part focuses on the snowboarding “choreography” and the tools and techniques employed for motion tracking. The last part introduces the interaction design and aims to provide some insight in the composed musical material.

2. DESIGN AND PROTOTYPING

2.1 From Snow to Space

“Architecture is frozen music” and “music is architecture in movement”, two statements by Novalis together with Xenakis’ “architecture becomes an art of time and music an art of space” in Polytopes are the main influence for this project [9].

Similar landscape or outdoor architectural projects using snow and ice as the main construction material, apart from vernacular architecture projects, can be seen in [10]. Many interactive art/architecture projects focus on interior spaces- [4][11][12]. However a project of this sort, which is developed outdoors, in a natural landscape, has to deal with the inherent difficulties of the weather conditions such as wind, insufficient snow etc.

The main architectural concept of the project was a synthesis of structures, modules and volumes, which would allow movement as well as projections on them. A controversial dialogue is created between moving snowboarders and solid, simple static modules. Figure 1 shows the basic snowpark modules.

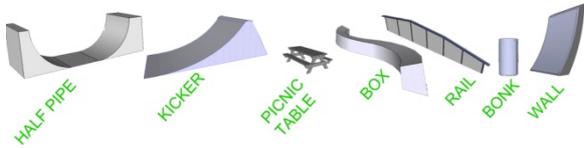


Figure 1. Basic Snowpark Modules

The site for the event is in the 7 Laux Ski Resort, in the French Alps. However, due to the weather conditions the test took place in Meribel ski resort, not far from the final event site, in the snowpark freestyle zone, which is a space for snowboarders to perform tricks.

Collaborating with HO5 [13], an events and snowpark development company responsible for both sites - 7 Laux and Meribel - we could propose different designs that were built for them as well as tested by their snowboarders.

As far as the architecture is concerned, there are two design phases: the first one is a simple design to be built for the interaction tests. During the first tests, it was important to explore the gestures and sounds as well as their interactivity using the technologies. For this reason a familiar environment was important, in order to qualify the basic movements.

For the tests, a very simple module arrangement was used; four modules are integrated to the design: a kicker, a box, a 'bonk', and a wall; modules which snowboarders already use for tricks and jumps. Using modules and surfaces familiar to the snowboarders helped us decide on which ones we would choose to redesign for the future event.

The choice of the modules depended not only on the variety of movements that each snowboarder facilitated but also on the modules' size and weight. The latter proved to be very useful since during the tests it was easy to move and re-arrange them. The idea was to have a common starting point and later three different options – either a box or a wall or a little kick and a bonk (Figure 2).

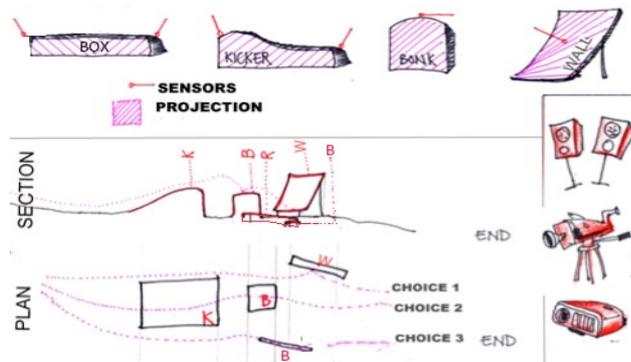


Figure 2. Sketches of the Modules Used for the Tests.

The starting point was 25m before the modules and the total area where the tricks took place, up to the ending point, was 40m².

A different architectural design is being introduced for the next phase of this project to encourage more tricks and therefore more interesting sound compositions. Figure 3 shows a sketch idea discussed with HO5. Although the initial idea was to use snow as our main construction material, after discussing it with HO5 we realized that it would not be financially feasible. A substantial volume of snow is expensive to produce and to maintain. The idea then is to utilize a composition of several little boxes, much smaller in length than the ones that they already use, scattered on a sloped area totalling 100m². Small kickers would facilitate the landing on the boxes. LED lights would be emitted from inside each box and would turn on or off interactively each time a snowboarder lands on them.

Four cubes made out of snow would be placed close alongside and the modules and interactive videos would be projected on the side more visible to the audience.

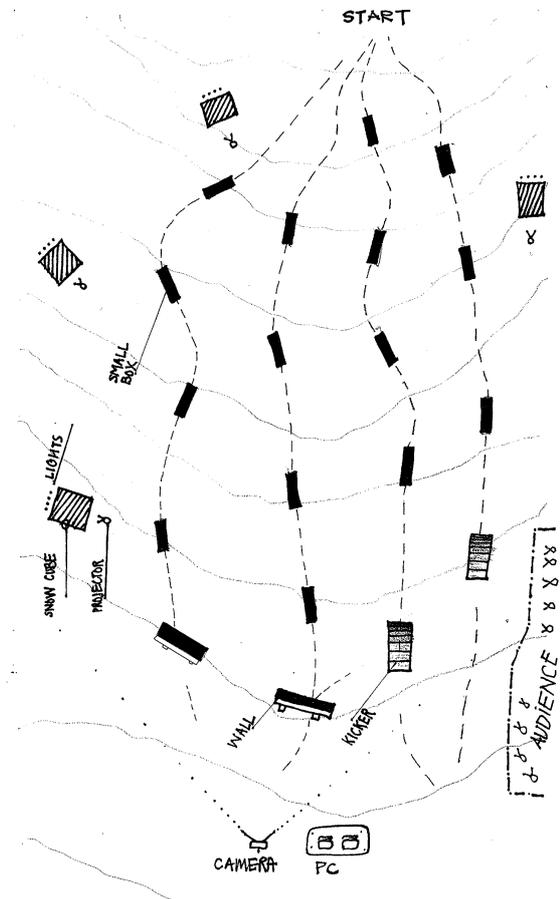


Figure 3. Sketches of the Proposed Snowpark for the Final Outdoor Event.

2.2 From Space to Movement

Snowboarding is a quite new winter sport and for some of its pioneers is considered an art form [14]. Ever since its inception as a sport, it has never stopped evolving and developing new styles. 'Freestyle' snowboarding on snowparks created a number of novel snowboarding

tricks and aerial maneuvers. In the current project, a few of them have been selected, modified and blended with the media aspects of the interactive performance.

When dealing with an environment used by the users/performers, it was important to research the different actions that one can take within the space. We wanted to give more choices to each snowboarder and not just restrict them with one and only path of progression. For this reason, the design consists of a modular arrangement, which would give them the choice and therefore the production of different and interesting sounds through their body gestures.

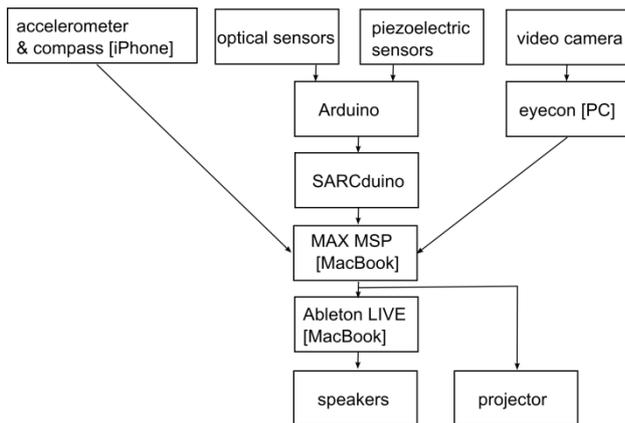


Figure 4. Schematic Representation of the Performance Setup

In order to capture the snowboarders' motion and realize an expressive dialogue between the elements of our piece, i.e. snow, space, movement and sound, it was necessary to use a set of tools and systems coming from a wide range of technologies [15, 3]. Basic features of the snowboarding performance such as motion, presence, position, orientation, velocity, acceleration and rotation had to be detected and tracked. Therefore electronic circuits, a range of sensors and transducers including a camera, microcontrollers, a computer and software have been employed largely for the connection between the physical and the digital world (figure 4).

Camera based motion tracking is very common in interactive art and design [16]. A computer system, software running computer vision algorithms and video cameras are essentially mandatory for every project that make use of video tracking technology. For the proposed site-specific interactive performance, high-end expensive motion trackers were considered inappropriate due to the outdoor nature of the project and for financial reasons. More appropriate modular software solutions including the Eyeswebplatform, MAX MSP Jitter with cv.jit library, 'Processing' programming language with the openCV library and the openFrameworks framework with the ofxOpenCV have been discarded too due to the programming complexity involved in order to make those environments work easily with the architectural elements of the project.

It became evident during the tests, that the most interesting interactions were based on the presence of the

snowboarders at specific locations of park. Therefore, Eyecon software, a commercial computer vision system specifically developed for interactive dance performances by Palindrome Inter-Media Performance, has proven itself useful in our setup [17]. Particularly, Eyecon offers a highly intuitive feature that lets you graphically define lines, zones and fields wherein the conceived interaction is to take place. For example a snowboarder can touch one of these virtual lines, which are drawn according to the architecture of the performance space and trigger or modulate graphics, videos and elements of the musical composition. A simple webcam positioned along the snowpark and a laptop running Windows XP has been used during the tests.

A variety of sensor technologies have been considered for prototyping too. Piezoelectric sensors have been used to transduce shock and vibration into an electric voltage. These sensors have been mounted on *bonk* and *wall* surfaces to detect when a snowboarder hits them. A custom analogue signal conditioning circuit has been designed and built to produce an accurate and reliable signal output.

Photoelectric switches have been considered too, in order to detect the snowboarders' presence on different points throughout the available paths in the snowpark. A small number of optical retroreflective sensors have been placed and tested in our designed installation space, which give accurate information about both the snowboarders' temporal- and spatial-presence at those particular positions.

Figure 5 shows the piezoelectric, sensor-based, impact sensing and pulse-stretching signal conditioning sub-circuit and figure 6 shows the retroreflective optical sensing signal-conditioning sub-circuit. Both of them have been designed to provide clear detectable pulses to the Arduino board, which is used as a sensor interface in our project, without damaging its input circuitry. The limited space of the article does not the authors permit to explain the details of their operation.

All the sensors worked well during the tests but probably will not remain in the final version of the installation, since the number of modules and the dimension of the space make this solution financially impossible for the time being. Moreover their cabling -wireless communication was too expensive and their size proved to be problematic when used in outdoor spaces.

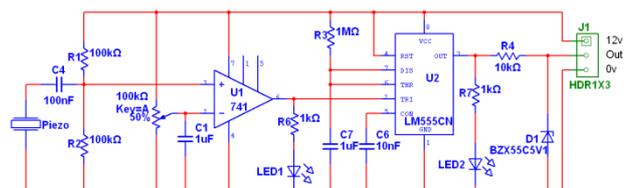


Figure 5. Piezoelectric Transducer Analogue Signal Conditioning Circuit.

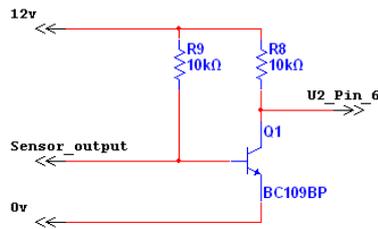


Figure 6. Retroreflective Optical Sensor Input Interface Circuit.

Accelerometers and electronic compasses have also been employed for motion tracking purposes. For the tests, we have used an iPhone, which offered a wide range of sensors including the previous ones. The c74 application has been used in order to transmit the data from the smartphone to the computer[18]. Spins, turns and ‘ollies’ were detected with these sensors.

All the digital signal conditioning and the gesture analysis from the sensors signals, from the iPhone and the Arduino board, have been processed in the MAX MSP software [19]. The SARCDuino sensor acquisition protocol has been used to communicate between the Arduino and MAX MSP [20]. Basic digital signal processing operations like scaling, smoothing, averaging, debouncing, edge detection mapping a given input range to a desired output range and low-level motion feature extraction has been programmed graphically using this intuitive block based programming language. A number of interesting and useful MAX MSP objects have been obtained by the *The Digital Orchestra Toolbox* developed by researchers at the IDML laboratory in McGill University [21].

2.3 From Movement to Sound

In every interactive multimedia project, which relies on physical computing, it is beneficial to be able to describe what actually happens in plain language. Moreover it is essential to break the project down into the stages of input, output and processing [15]. Table 1 summarizes and Figure 2 ‘sketches up’ some of the basic interaction designs we have used and tested so far. The following paragraphs explain and try to offer an insight on the depicted interaction decisions.

As we have already mentioned, making music and media art through movement is not novel. Interpreting movement data collected through a variety of electronic sensors and mapping them into composition procedures that generate, sequence and transform music and every kind of media material is the core of interactive art and interactive design [22][23].

| Input | Output | Processing |
|-------|-----------|----------------------|
| start | play clip | when he start moving |
| stop | stop clip | when he stops moving |

| | | |
|-------------------|-------------------------------|--|
| take off | reverb on | when he is in the air |
| landing | reverb off | when he hits the ground |
| location | change clip / play a new clip | when he crosses some predefined points |
| sustain | retrigger clip | when he stays on a certain point |
| hit bonk/jib/wall | play sound effects | when the board touches these surfaces |
| accent | granulation | when he is doing fast and big nervous gestures |
| speed | tempo | - |
| grind / slide | distortion / bit reduction | the contact point works like a slider |
| spin | tremolo | like a rotary potentiometer |
| flip | flanger | like a rotary potentiometer |
| carving | panorama | like an envelope |
| shifty | volume | like an envelope |
| grab | turn off the volume | when he touches the board |

Table 1. Table of Some Basic Interactions.

How can snowboarders organize and structure musical and visual material through their physical gestures? The answer in our approach was to find an effective interaction, musical and visual material that maximizes the chances of aesthetic interesting results from the point of view of both the performers and the audience. It has been strongly considered that an understandable mapping from snowboarding to music and visual projections should reach everybody in the performance in a straightforward manner. Hence elementary research concerning the snowboarding culture and events has been carried out in the design phase. Particularly, the compositional strategy of the piece has been strongly based on the snowboarding, street and urban cultures, which are characterized by musical genres such as hip-hop, rap, dub, pop and electronica. A very useful resource, which immediately re-

flects the aesthetic qualities of this particular group of people, are the snowboarding films/documentaries and the web site of our partner HO5parks [11, 24].

From a musical point of view, the initial material of the composition was a database of sound samples obtained from commercial recordings, field recordings, sample libraries and other musical material sequenced and performed by the first author of the paper. Acapella sections and refrain sections from songs, drum and percussion loops, base line patterns, guitar motifs and pure recordings from the Alpine environment, all form this mid-size database of pre-composed, unstructured musical elements.

The musical idiom used is rhythmical, repetitive and tonal. The structure is open, non linear and unfolds during the performance. Clearly the piece is a remix of pre-existing musical material relevant to the snowboarding culture. Elements of the music, like arrangement, form and timbre qualities of each sequenced sample are indeterminate.

The scenario followed for the predetermined and indeterminate actions, based on Winkler's four basic categories, is that of "*performer improvisation and predetermined computer sequences*" [23]. Aspects of indeterminacy are used only to choose specific samples from the collection and for timbre modification through the use of digital audio effects. Mapping of performance gestures to the parameters of classical signal processing algorithms (granulation effects, filtering, pitch shifting, reverberation, distortion, etc) has been regularly employed in every part of the piece. Simple timing processes i.e. quantization, delaying and speed change have been used as well. Sections of the piece documented during the tests will hopefully be uploaded soon on the HO5 website [11] and will be presented in the conference.

Core to the design of the musical interactions was the understanding that simple, immediate and straightforward motion-sound relations were essential for the snowboarders. It has been verified during the tests that, generally, the sonic modifications were necessary to explicitly follow the flow of motion. Hence simple cause-effect relationships were designed; i.e. start music when movement starts, create harsh sounds when big nervous gestures are produced. In general, it was difficult to engage the snowboarders with interactions that generated subtle timbre variations and make them concentrate on fine audio outcomes. However interactions that augmented directly their snowboarding performance i.e. tricks that require equilibrium linked to digital audio effects like 'spectral freeze', inspired them and motivated them for further exploration.

The short duration of their movements, 1- 2 sec approximately, indicated that continuous control of sonic elements was not very easily perceived. Even from the audience point of view, it became clear that only long movements like smooth slow turns could be mapped to dynamic audio modifications. Jumps and every possible module riding had to be mapped to obviously distinguishable sound effects like triggered music clips.

From a technical point of view, a laptop computer and two software environments have been used for the realization of the musical piece. As we have explained in the previous section of the paper, sensor data acquisition from wearable and spatial sensors, have been realized with the help of an Arduino board and an iPhone. Then, all the analysis and mapping of sensors' output data, including the ones from the video tracking system, have taken place in Max MSP graphical programming language. Simple explicit strategies were sufficient to control the compositional procedures [25]. As mentioned above, the computer vision algorithms are running on a separate computer linked with the previous one through Ethernet by the OSC protocol. All the control events were transmitted via MIDI to an Ableton Live-equipped digital audio workstation, which performed all the sound related real-time processing [26].

3. CONCLUSIONS AND FUTURE WORK

The project, even at this early testing phase provided interesting results and offered us a very informative experience, thought beneficial for the design of the future event. Every difficulty we experienced, production-wise at the location and particularly during the collaboration period with the HO5 has already started shaping and forming our ideas and designs regarding the final version of the multimedia piece.

A night performance with artificial lights will be much more controllable and robust compared to the unexpected lighting conditions that were experienced during the daytime tests. Use of infrared illumination is planned, in order to integrate the visual projections with the music, the movement and the architecture, without disturbing the video tracking procedure. Retroreflective tape placed on the snowboarders' legs or even powered infrared sources will improve the visibility of the tracked objects and the overall performance of the computer vision algorithms. So far, colored LED lights have been tested, as seen in Figure 7, in order to track multiple points in the dark or under low-light conditions. This solution will not, in all probability, be implemented in the final event, since it is not well adapted to the visual aesthetics of the performance.

It became clear during the tests that when snowboarders learn from the experience of play, they react and later redefine the sense of place. We tried to raise the awareness of the dialogue between them and the environment. According to Fox and Kemp, the environment can be either an entity or a discrete organization of devices and systems, and the behavior can be a direct response or emergent. Users become participants either willingly or unwillingly, and their behaviors are translated not only to themselves and others within a particular space, but also to those on the outside looking in. [4]



Figure 7. Tests with LED Lights

The sense of sound is much underrepresented in discussions of architectural experience and it is very often only dealt with from a design standpoint, relative to the negative aspects [4]. Interactive performances in designed spaces, as well as interactive architecture, have begun to question this. We believe that in the future there will be a stronger relationship between architecture, music and movement through projects of this type.

A series of events is envisaged to take place next winter in the HO5 snowparks and soon after in an indoor venue in Paris.

Acknowledgments

We would like to thank the Strategic Insight Program founded by HEFCW in UK and the HO5 events and snowpark design organization for their help and collaboration.

4. REFERENCES

[1] S. Dixon, *Digital Performance*, The MIT Press, 2007.

[2] J. Chadabe, *Electric Sound: The Past and Promise of Electronic Music*, Prentice Hall, 1997.

[3] J. Noble, *Programming Interactivity*, O'Reilly, 2009.

[4] M. Fox, M. Kemp, *Interactive Architecture*, Princeton Architectural Press, 2009

[5] E. Miranda, M. Wanderley, *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*, A-R editions, 2006.

[6] R. Wechsler, "Artistic Considerations in the Use of Motion Tracking with Live Performers: a Practical Guide", in *Performance and Technology: Practices of Virtual Embodiment and Interactivity*, Ed S. Broadhurst and J. Machon, Palgrave Macmillan, 2006.

[7] A. Camurri, S. Hashimoto, M. Ricchetti, A. Ricci, K. Suzuki, R. Trocca, G. Volpe, "EyesWeb: Toward Gesture and Affect Recognition in Interactive Dance

and Music Systems", in *Computer Music Journal* 24(1), 2000, pp.57-69.

[8] J. Schacer, "Motion to Gesture to Sound: Mapping for Interactive Dance", in *Proc. Int. Conf on New Instruments for Musical Expression (NIME2010)*, Sydney, 2010, pp.250-254.

[9] I. Xenakis, *Musique de l'Architecture*, Parentheses, 2004

[10] L. Fung, J. Debany, *The Snow Show*, Thames and Hudson, 2005.

[11] Troika, C. Freyer, S.Noel, E. Rucki, *Digital by Design: Crafting Technology for Products and Environments*

[12] L. Bullivant, *Responsive Environments*, Architecture, Art and Design, V&A, 2006.

[13] <http://www.ho5park.com>

[14] J. Smith, *The Art of Snowboarding*, Ragged Mountain Press, 2007.

[15] D. O'Sullivan, T. Igoe, *Physical Computing – Sensing and Controlling the Physical World with Computers*, Course Technology, Cengage Learning, 2004.

[16] G. Levin, "Computer Vision for Artists and Designers: Pedagogic Tools and Techniques for Novice Programmers". *Journal of Artificial Intelligence and Society*, Vol. 20.4. Springer Verlag, 2006.

[17] R. Wechsler, F. Weiss, P. Dowling, "Eyecon – a Motion Sensing Tool for Creating Interactive Dance, Music and Video Projections", in *Proc. of the Society of Study of Artificial Intelligence and the Simulation of Behavior (SSAISB)*, Leeds, 2004.

[18] <http://www.nr74.org/c74.html>

[19] <http://cycling74.com/>

[20] <http://www.somasa.qub.ac.uk/~MuSE/>

[21] http://www.idmil.org/software/digital_orchestra_toolbox

[22] R. Rowe, *Interactive Music Systems*, The MIT Press, 1993.

[23] T. Winkler, *Composing Interactive Music – Techniques and Ideas Using Max*, The MIT Press, 1998.

[24] <http://www.pirate-movie-production.com/>

[25] D. Arfib, J.-M. Couturier, L. Kessous, V. Verfaillie, "Strategies of mapping between gesture parameters and synthesis model parameters using perceptual spaces", in *Organised Sound*, 7(2), 2002, pp.135–152.

[26] <http://www.ableton.com/>

A BAYESIAN APPROACH TO DRUM TRACKING

Andrew N. Robertson

Centre for Digital Music

School of Electronic Engineering and Computer Science,

Queen Mary University of London

andrew.robertson@eeecs.qmul.ac.uk

ABSTRACT

This paper describes a real-time Bayesian formulation of the problem of drum tracking. We describe how drum events can be interpreted to update distributions for both tempo and phase, and how these distributions can be combined together in a real-time drum tracking system. Our algorithm is intended for the purposes of synchronisation of pre-recorded audio or video with live drums. We evaluate the algorithm of a new set of drum files from real recordings and compare it to other state-of-the-art algorithms. Our proposed method performs very well, often improving on the results of other real-time beat trackers. The algorithm is implemented in C++ and runs in real-time.

1. INTRODUCTION

This paper concerns the problem of real-time drum tracking, which estimates the time-varying tempo and beat locations from drum signals. This is very important for the construction of interactive performance systems in rock and pop music where drum events define the rhythm of the piece. By using the estimated metrical location of beats, a drum tracker can be used to synchronise pre-recorded audio and visual components with musicians or to analyse aspects of the music relative to the beat, as might be required for generative music.

There have been many approaches to the problem of beat tracking on audio files from a database. A comprehensive review of these is given by Gouyon and Dixon [1]. For use in performance systems, we require algorithms that operate in real-time. Autocorrelation-based approaches are used by *DrumTrack* [2] and *Btrack~* [3], a causal version of Dan Ellis' dynamic programming algorithm [4]. *B-Keeper* [5] is a specialised drum tracker that uses a rule-based approach and adapts system parameters using explicitly coded expert-knowledge. *IBT* [6] is a real-time implementation of the BeatRoot algorithm by Dixon [7] that uses multiple agents, each representing a tempo and phase hypothesis. Seppänen et al. [8] jointly estimate beat and tatum in a system that was of sufficiently low computational cost to be implemented for the S60 smartphone.

Copyright: ©2011 Andrew N. Robertson et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Passages featuring complex rhythms such as syncopation, expressive timing or drum fills present a challenge to these algorithms, since these are situations where musical events no longer occur in synchrony with the underlying beat of the music. In traditional approaches to beat tracking, the assumptions often remain hidden, such as the notion that events or high values in the onset detection function correspond to events that are 'on the beat'. Whilst this might often be true, this is not the case with syncopated rhythms, and this results in beat trackers that do relatively well across the board but exhibit difficulties following complex rhythms [9].

Here, we shall adopt a Bayesian formulation approach to the problem. One advantage of this is that the underlying assumptions have to be made explicit. Cemgil et al. [10] have formulated the problem of tempo tracking for music transcription within a Bayesian framework. Bayesian approaches have been used successfully to tackle some related problems. In automatic accompaniment, Grubb and Dannenberg [11] proposed a stochastic method for following a vocalist in which their current location was represented by a probability distribution.

2. ALGORITHM DESCRIPTION

Traditional beat trackers tend to require three stages:

- feature extraction: input to beat tracking systems tends to either be *event-based*, in which the input is a list such as pitch or event onset times, or else they accept '*continuous*' input from an onset detection function which is a frame-by-frame measure of the strength of the extent to which the current audio frame is a note onset. Bello et al. [12] provide a detailed discussion of the various functions that have been employed.
- tempo induction: this process provides an approximation of the tempo and phase, thus initialising the beat tracker.
- beat tracking or following the beat: this is an iterative process by which the algorithm processes features from the audio to update the current beat location and infer the location of the next beat.

Here we will make some simplifying assumptions that allow us to formulate the problem of beat tracking in a Bayesian framework. Firstly, we will be using an event-based approach so that input to the system is in the form of a simple

description of a drum event such as ‘kick’ or ‘snare’. Since drum events in rock music are characterised by the stick or beater hitting the skin, this way of representing the signal makes sense intuitively. It might fare less well in, say, jazz music, where drum events can be more diverse in terms of their sonic qualities, but in rock and pop the defined beat means that an event-based representation helps by removing noise from the input and decreasing the amount of computation that is required. We also accept as input the music sequencer’s beat events whose tempo is controlled by the output of the algorithm.

We propose two distributions for the random variables representing the beat period and the phase which represent our belief as to the values of these two quantities. These feature in our model as parameters τ , the beat period, and θ the phase of the relation to the sequencer’s click, and we wish to update these distributions on the basis of observed drum events. To do so, we shall employ Bayesian reasoning. Supposing we have a model characterised by parameters \mathbf{w} , then our uncertainty about the parameters can be expressed as a *prior* probability distribution $p(\mathbf{w})$, that reflects our hypothesis before observing the data D . On observing data D , we can update our uncertainty through the use of Bayes’ theorem:

$$P(\mathbf{w}|D) = \frac{P(D|\mathbf{w})P(\mathbf{w})}{P(D)} \quad (1)$$

The term $P(D|\mathbf{w})$ is the conditional probability of observing data D , given the current model parameters \mathbf{w} , and is referred to as the *likelihood function*. The term on the left side, $P(\mathbf{w}|D)$ is the *posterior* and is our distribution over the hypotheses given the observation of the data which is what we want to calculate. The term $P(D)$ is the probability of observing the data under all possible hypotheses. This can be considered to be a normalising constant to ensure that the posterior probability will integrate to one, and in practice this often does not need be evaluated.

2.1 Updating the tempo distribution

Music psychologists have identified that humans have preferred tempo rates and broadly there tends to be a preference for an optimal beat interval between 600 and 700 ms [13] with preferred periods ranging between 429 to 725 ms [14]. Here we enforce a strict limitation on the allowed tempi, referred to here as τ_{min} and τ_{max} . We chose a minimum beat period of 360 ms and maximum 800 ms, corresponding to 168 beats per minute (BPM) and 76 BPM respectively. We will describe how to update the beat period hypothesis assuming there is a prior beat period distribution and a current estimate $\hat{\tau}$ between these limits.

Suppose our incoming data, D , is in the form of a drum event at time t_n , specified in milliseconds from the computers’ system time. The Bayes formula states that

$$p(\tau|D, \theta) \propto p(D|\tau, \theta)p(\tau) \quad (2)$$

Then for another recent drum event occurring at previous time, t_k , the interval between the two events is $t_n - t_k$ ms. Given our current beat period estimate, $\hat{\tau}$, this interval

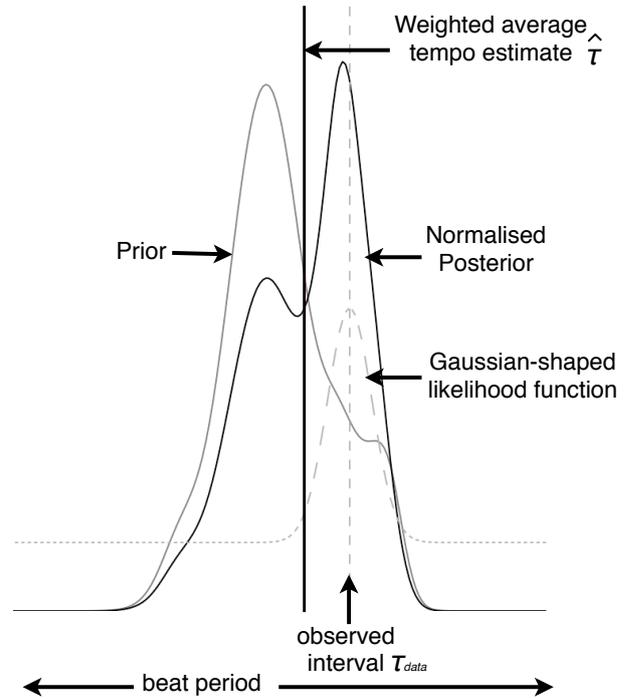


Figure 1. The likelihood function (light grey) and the prior (dotted) give rise to a new posterior tempo distribution (dark grey). The Gaussian shape in the likelihood function is centred on the new tempo observation, τ_{data} , indicated by a light grey vertical line. The new beat period estimate, $\hat{\tau}$, is given by the weighted average over the posterior distribution (vertical dark line).

corresponds to an integer number of beat intervals, given by

$$v(n, k) = \text{round}\left(\frac{t_n - t_k}{\hat{\tau}}\right), \quad (3)$$

and thus, where this is positive, our estimate for the beat period from observing drum events at times t_n and t_k is

$$\tau_{data} = \frac{t_n - t_k}{v(n, k)} \quad (4)$$

Our *likelihood function* can be computed directly from this estimate given some model assumptions which we must describe explicitly. We assume that the actual tempo gives rise to noisy observations, so that our likelihood function, the probability distribution of the actual beat period given this sole observation, is a mixture of a Gaussian distribution of standard deviation σ_T centred around the observed estimate τ_{data} and white noise across all possible values of τ . This gives

$$p(\tau|\tau_{data}) = \nu_T + (1 - \nu_T) \frac{1}{\sigma_T \sqrt{2\pi}} \exp\left(-\frac{(\tau_{data} - \tau)^2}{2\sigma_T^2}\right), \quad (5)$$

where ν_T is the quantity of noise across all possible values. We expect that there will be a relatively high amount of drum events, none of which we wish to place too much emphasis upon. In order to do so, we chose the value 0.7 for ν_T , suggesting that 70% of the time our observed

data point is unrelated to the actual beat period. We also wish to pick an appropriate balance for σ_T (set by hand to $\frac{\tau_{max} - \tau_{min}}{32}$). If it is too high, the Gaussian will be wide and the tempo will not be picked out sufficiently, whereas if it is too low, the Gaussian will be shaped like a delta peak and there will be less contribution to values for τ close to but different from τ_{data} . Since we want successive observations that are close to a central value to reinforce each other, we require a balance between being too specific and not giving enough weight to the data.

Our posterior distribution is calculated through the application of the Bayes formula, as stated in Equation 2. We take the product of our prior distribution and our likelihood function, and then normalise. Finally, our beat period estimate $\hat{\tau}$ can be calculated as the integral of the posterior over all possible values of τ , so that

$$\hat{\tau} = \int_{\tau_{min}}^{\tau_{max}} \tau p(\tau) d\tau \quad (6)$$

where $p(\tau)$ is the posterior distribution for the beat period. An alternative is to take the maximum a posteriori probability (MAP) estimate. Finally, we also model subtle changes in the underlying tempo that may occur during the performance by adding Gaussian noise around the beat period estimate $\hat{\tau}$. Thus at successive time steps between drum events, we also update our posterior distribution:

$$p(\tau) \leftarrow p(\tau) + n(0, \sigma_{noise}^2) \quad (7)$$

Sequential use of the Bayes' theorem mean that the posterior will be used as our prior when the next data observation is made and the process repeats iteratively.

2.1.1 Tempo Initialisation

In order to initialise the system in the absence of a count-in or a reliable prior approximation for the tempo, we use the same technique as above but choose a uniform prior that reflects our lack of knowledge about the tempo. The result on iteratively updating the posterior is that within a couple of bars the distribution tends to peak around the correct tempo, or occasionally a multiple of the tempo such as double or half. In performance, we would most likely use an approximation for the tempo and phase to initialise the algorithm with more suitable prior distributions representing our knowledge about these variables.

2.2 Phase Estimation

Since we intend to use the drum tracker to synchronise accompaniment, it makes sense that there is a unique tempo, of beat period $\hat{\tau}$, that is the outcome of the tempo tracking process described in Section 2.1 above. In estimating the phase of the sequencer, we make the assumption that this tempo estimate is correct and events are only analysed for phase *relative to that tempo*. This way, we can treat phase independently and have only one distribution which can be updated using a similar methodology as employed for tempo. Whilst it would be formally more correct to treat the two variables as forming a joint distribution, in practice the tempo distribution tends towards a sharp peak

around the correct underlying tempo and so the assumption is reasonable. If phase were treated differently, then this would raise the problem of keeping track of a phase distribution for each tempo, which would incur significant computational costs. Many beat trackers work by finding the optimum tempo and then outputting the phase estimate at that tempo, although some approaches to beat tracking jointly analyse tempo and phase, for example as found in Eck [15] and Dixon [16]. Thus, we may say that

$$p(\theta|D, \hat{\tau}) \propto p(D|\theta, \hat{\tau})p(\theta, \hat{\tau}) \quad (8)$$

On observing a new drum event at time t_n , we first need to update our posterior distribution to account for the time interval between these events. As this interval increases, we expect uncertainty as to the exact tempo and noise within the drummer's internal timing to result in increasing uncertainty as to the beat location. This can be modelled using the tempo distribution. The interval since the last update is $t_n - t_{n-1}$ ms, where t_{n-1} is the time of the previous drum event measured by the computer's system time. At our tempo estimate $\hat{\tau}$, it corresponds to a specific number of beat intervals given by:

$$v_n = \frac{t_n - t_{n-1}}{\hat{\tau}} \quad (9)$$

Then at a given phase θ , the contribution from phases around θ can be calculated using the tempo distribution to account for how the various possibilities of phases and tempos might combine together. To understand this, consider how an earlier phase at a slower tempo would predict the same beat location as a later phase at a faster tempo. So, for a 1ms difference in beat period, the interval of v_n beat periods creates a time difference of v_n ms in phase offset, which translates as a difference of $\frac{v_n}{\hat{\tau}}$ in terms of relative phase. Since the arrays are recorded discretely, we give the formulation as a sum rather than the integral. Then

$$p(\theta) = \sum_k p(\hat{\tau} + kd_T)p(\theta - \frac{v_n kd_T}{\hat{\tau}}) \quad (10)$$

where d_T is the time interval between adjacent bins in the tempo array (we used 240 bins for the beat period range 360 to 800 ms, so $d_T = 1.83$). Thus, if for notational purposes here, we define ψ_n as $\frac{v_n}{\hat{\tau}}$, then the phase at θ has a contribution of $p(\theta)p(\hat{\tau})$ from the tempo estimate $\hat{\tau}$, a contribution $p(\theta - \psi_n d_T)p(\hat{\tau} + d_T)$ from the tempo of period $\hat{\tau} + d_T$, a contribution $p(\theta + \psi_n d_T)p(\hat{\tau} - d_T)$ from the tempo of period $\hat{\tau} - d_T$, and there is a contribution $p(\theta - 2\psi_n d_T)p(\hat{\tau} + 2d_T)$ from the beat period $\hat{\tau} + 2d_T$, and so on. We then renormalise the array to get our updated prior distribution at time t_n . This now reflects how uncertainty in the tempo distribution translates into uncertainty in phase given the interval between two most recently observed events. We now wish to calculate our likelihood function, $p(\theta|D)$. The drum event happens at time t_n , which corresponds to a phase θ_n . If the closest beat time (recent or predicted) is at $\xi(n)$ ms relative to the computers system time (where the n is included to acknowledge the dependance upon t_n), then the relative phase is given by

$$\theta_n = \frac{t_n - \xi(n)}{\hat{\tau}} \quad (11)$$

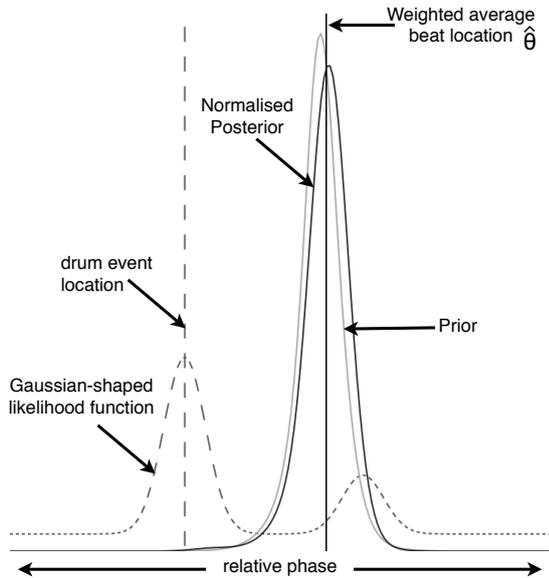


Figure 2. The posterior (dark grey) is calculated as the product of the prior distribution (light grey) and the likelihood function (dotted). In this case, the event is most probably an early sixteenth since it is approximately a quarter of the beat period to the left of our phase estimate, labelled $\hat{\theta}$.

Where the event occurs on the beat, we will model this as a combination of a Gaussian centred on the events location relative to the beat estimate and noise across all possible phases. So, given the observed drum event at relative phase θ_n , the likelihood function is

$$p(\theta|\theta_n) = \nu_P + (1 - \nu_P)g(\theta, \theta_n, \sigma_P) \quad (12)$$

where the Gaussian contribution is

$$g(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (13)$$

However, this also depends upon the drum event type, the drum pattern and the location of the event. Only if we interpreted every event as being on the beat, would we wish to use the formulation of Equation 12. To understand how this is so, consider an event happening on an eighth note. In this case, we should not necessarily model the likelihood function as a Gaussian around the beat location, since if we believe it to be an eighth note, then it would suggest that the beat is half a period before and after the drum event. Thus, our likelihood function ought to be dependent upon metrical position in the bar.

Here, we will use a methodology that assumes we have a reasonable estimate for the phase and see how this allows us to interpret the rhythmic features. To do so, we divide the region between beats into twelve regions of equal duration with the first region, labelled 0, which corresponds to time $\xi(n)$. We measure the phase relative to the computer's click so that $\xi(n)$ always has a phase of zero. Whilst the predicted beat time $\xi(n)$ is not necessarily at the same relative phase as our optimal estimate $\hat{\theta}$ (since we may be in the process of adjusting), in practice the two will be very

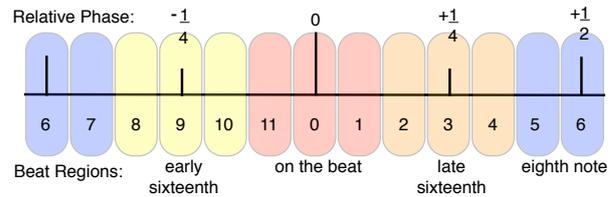


Figure 3. Phase is defined relative to the computer's click track (at time $\xi(n)$) which has phase 0. Events happening in the different regions are interpreted differently, resulting in different Gaussian mixtures in the likelihood function.

close since the system constantly makes adjustments as its output so that, according to the current tempo and phase estimate, $\hat{\theta}$ is predicted to be zero in the near future.

We can say that we would expect events happening in region 0 close to the expected beat location at time $\xi(n)$ and the adjacent regions, labelled 1 and 11, to correspond to events that are 'on the beat'. Thus these events have a likelihood function as given in Equation 12. Events in regions 5, 6 and 7 might constitute events at beat times, but when if our beat estimate is reasonable, it is more likely that they correspond to eighth note events that happen an eighth note after the actual beat. Thus we represent the likelihood for these events as a mixture of two Gaussians, one around where the event happened and one an eighth note away. In this case,

$$p(\theta|\theta_n) = \nu_P + (1 - \nu_P)[\alpha g(\theta, \theta_n, \sigma_P) + (1 - \alpha)g(\theta, \theta_n + \frac{1}{2}, \sigma_P)] \quad (14)$$

where the Gaussian function, g , is as specified in Equation 13. Through testing we chose a value of alpha of 0.3. In the two region groups 2, 3 and 4, and 8, 9, 10, which we interpret as potentially being sixteenth note events that occur after and before the beat respectively, we use the likelihood function given by

$$p(\theta|\theta_n) = \nu_P + (1 - \nu_P)[\alpha g(\theta, \theta_n, \sigma_P) + (1 - \alpha)g(\theta, \theta_n \pm \frac{1}{4}, \sigma_P)] \quad (15)$$

where the sign in the second Gaussian is minus for events occurring after the beat and plus for before and alpha is again 0.3. In Figure 2, we show an event that uses this mixture of noise with the remainder one Gaussian weighted by 70% around the event and one Gaussian a quarter of a beat period later weighted by 30%, due to the chance that it is an early sixteenth note. We found in practice that whilst the balance is helpful to stabilise a beat interpretation, weighting the sixteenth too highly would inhibit the tracker to respond to variation in tempo and result in too high a stability at alternative phase estimates. The need for a balance between inertia and reactivity has been commented on by Gouyon and Dixon [1] as a common feature of rhythm description systems. We adjusted the value for likelihood noise ν_P by hand and chose 0.56.

2.2.1 Update Rule

Having updated both distributions, we estimate that the underlying beat period is $\hat{\tau}$ and the relative phase is $\hat{\theta}$. This

| Drummer (piece) | Human Tapper | Proposed Bayesian | B-Keeper | Btrack~ | CFM |
|---|--------------|-------------------|-------------|---------|-------------|
| Adam Whitfield (Dropped D) | 26.4 | 19.1 | 23.6 | 23.3 | - |
| Whetham Allpress (BlitzKrieg) | 18.3 | 11.8 | 19.5 | 23.1 | 21.1 |
| Whetham Allpress (Cannibal Island) | 15.3 | 15.4 | 52.9 | 20.4 | - |
| Jem Doulton (Funky Riff#07) | 30.1 | 14.7 | 14.2 | 20.2 | 19.6 |
| Mark Heaney (Syncopated Beats piece I) | 20.7 | 47.8 | 37.1 | 42.1 | 36.5 |
| Mark Heaney (Syncopated Beats piece II) | 17.7 | 58.0 | 35.6 | 73.9 | - |
| Al Pickard (Funk) | 20.2 | 18.3 | 17.6 | 22.9 | - |
| Hugo Wilkinson (Follow The Leaders) | 17.5 | 18.0 | 20.7 | 23.2 | 23.4 |
| Rod Webb (Reggae beat) | 15.3 | 22.3 | 15.1 | 26.3 | 45.6 |
| Adam Betts (Hum1) | 20.1 | 25.4 | - | 50.3 | 20.9 |
| Adam Betts (Swung loop) | 25.9 | 27.1 | - | 29.1 | 20.2 |
| David Nock (Speed up drums) | 16.4 | 12.5 | 34.7 | 20.6 | 21.9 |
| Marcus Efstratiou (Billy Jean) | 14.5 | 15.9 | 21.7 | 21.6 | 16.7 |
| Metronome (120 BPM) | 17.3 | 5.2 | 2.1 | - | 12.3 |

Table 1. Average absolute error (ms) between the output of beat tracking algorithms and drum events from the percussive onset detector *bonk~* that are ‘on the beat’. The time in bold is the closest synchronisation to the ground truth (as defined by the drum beats) for each piece. Those marked ‘-’ failed to track the beats and lost synchronisation.

phase error corresponds to $\hat{\theta}\hat{\tau}$ ms. The single beat period sent as output to our sequencer is $\hat{\tau} + 0.6\zeta$, where ζ is the estimated remaining phase error (reset to the new estimate $\hat{\theta}\hat{\tau}$ after every event). On each beat, a new period is sent out, so we update:

$$\zeta \leftarrow 0.4\zeta \quad (16)$$

Subsequent information in the form of new drum onset events leads the process repeats iteratively.

3. EVALUATION

We have evaluated the algorithm on audio files of drum recordings, made either in band rehearsals or during the development and testing of the drum tracking software system *B-Keeper* [5]. These files have the characteristic of being at relatively steady tempi, so that there are no sudden shifts, although a couple do change tempo slowly over the course of half a minute or so. We have collected the beat locations predicted by four algorithms: the proposed Bayesian tracker, Stark et al.’s *Btrack* [3], our previous algorithm *B-Keeper* [5], and a comb filter matrix beat tracker (CFM) described in Robertson et al. [17], based on methods similar to Eck’s Autocorrelation Phase Matrix [15]. We also collected the annotations made by the author using an Oxygen MIDI keyboard and tapping with a cowbell sound.

In some cases, we allowed manual adjustments to ensure that each algorithm was initialised to the correct beat period and phase. *B-Keeper* requires an approximation of the tempo for initialisation. *Btrack~* tends to find the right tempo without any intervention, suggesting it is a reliable option when manual intervention is not allowed. The comb filter matrix beat tracker requires two consecutive taps at the beat period to indicate the tempo and phase. In the case of our proposed algorithm, we initialised the algorithm with a uniform prior and waited for it to find the correct tempo. In the few cases where the phase estimate was on the off-beat, we reset the phase distribution to be

uniform so that the algorithm would be able to find the correct phase estimate.

In order to evaluate the algorithms, we require ground truth annotations for where the beat locations are. We maintain that drum events define the beat where they occur on the ‘one’, ‘two’, ‘three’ and ‘four’ of the bar. Whilst we might expect some expressive playing, all trackers are at the same disadvantage in predicting such shifts in phase. We made use of the expert ability of human tappers to understand meter and respond to changes in tempo and phase to allow us to automatically label those drum events which would be the ground truth if one annotated by eye. The audio from the kick and snare drum microphones was passed through the *bonk~* object [18], a fast onset detector for percussive sounds. Where these times were within 80 ms to the human annotated beat locations, the onset times were understood as defining the beat locations and thus used as ground truth in the evaluation. This method differs from conventional beat tracking analysis where human annotations are taken as ground truth on the basis that the drums define the beat and so clear drum onsets can be used instead.

We computed the average absolute error between kick and snare events and the beat time where these fall close to the beat. The results are presented in Table 1. This gives an indication of how well each drum tracker has predicted the drummer’s actual beat. We have found that our proposed method performs very well with respect to other state-of-the-art algorithms. In many cases, it gives comparable results to *B-Keeper* and on six tracks out of fourteen (often the more conventional rhythmically speaking) it is closer than the human tapper. Whilst *Btrack~* behaves reliably, the mean offset is higher than our proposed algorithm for all but one track. This might be due to a latency issue since where the tracking was successful, we observed our method to have a mean error of roughly zero (with no bias before or after the beat), whilst *Btrack~* tended to have positive error between 10 and 20 ms. We also found evi-

dence of the negative asynchrony as described in Aschersleben [19], whereby human subjects tend to tap *before* the beat and this asynchrony varied between 0 and 20 ms.

4. CONCLUSIONS AND FUTURE WORK

Firstly we remark on two ways in which our system might be improved.

- **Rhythm Interpretation:** Since we have a relatively stable way in which to track small changes in the beat period and phase, we might explicitly model and learn the rhythmic pattern. This might extend our ability to handle tempo shifts or recover from errors, since if we are able to interpret events correctly, then new events could convey more reliable information than the current formulation permits.
- **Learning Probabilities:** Whilst we have used parameter settings made by hand, such as the standard deviations of the Gaussians that contribute to the likelihood functions for phase and period, we might be able to estimate these from the data. In addition, we might estimate the ratio between noise and reliable estimates. Approximations to these from analysis of drum data might improve the performance of the system or be the default starting parameters as observations of a particular drummer's style might allow the system to learn more optimal values.

We have presented a real-time Bayesian algorithm for the tracking of tempo and phase in drum signals. This is evaluated on real signals from drum tests and we find that the algorithm compares favourably with the state-of-the-art. With a view to supporting the community's aims for Reproducible Research, the algorithm, the code required to test it and the drum files are available online¹.

5. REFERENCES

- [1] F. Gouyon and S. Dixon, "A review of automatic rhythm description systems," *Computer Music Journal*, vol. 29, no. 1, pp. 34–54, 2005.
- [2] N. Collins, "DrumTrack: Beat induction from an acoustic drum kit with synchronised scheduling," in *Proceedings of International Computer Music Conference*, 2005.
- [3] A. M. Stark, M. E. P. Davies, and M. D. Plumbley, "Real-time beat-synchronous analysis of musical audio," in *Proc. DAFX-09*, 2009, pp. 299–304.
- [4] D. P. W. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.
- [5] A. Robertson and M. Plumbley, "B-Keeper: A beat-tracker for live performance," in *Proc. International Conference on New Interfaces for Musical Expression (NIME)*, New York, USA, 2007, pp. 234 – 237.
- [6] J. Oliveira, F. Gouyon, L. G. Martins, and L. P. Reis, "IBT: A real-time tempo and beat tracking system," in *International Conference on Music Information Retrieval*, Utrecht, 2010, pp. 291–296.
- [7] S. Dixon, "Evaluation of the audio beat tracking system beatroot," *Journal of New Music Research*, vol. 36, no. 1, pp. 39–50, 2007.
- [8] J. Seppänen, A. J. Eronen, and J. Hiipakka, "Joint beat and tatum tracking from music signals," in *International Conference on Music Information Retrieval ISMIR 2006*, 2006, pp. 23–28.
- [9] R. B. Dannenberg, "Toward automated holistic beat tracking, music analysis and understanding," in *International Conference on Music Information Retrieval*, 2005, pp. 366–373.
- [10] A. T. Cemgil, H. J. Kappen, P. Desain, and H. Honing., "On tempo tracking: Tempogram Representation and Kalman filtering," *Journal of New Music Research*, vol. 28, no. 4, pp. 259–273, 2001.
- [11] L. Grubb and R. B. Dannenberg, "A stochastic method of tracking a performer," in *Proc. International Computer Music Conference*, 1997, pp. 301–308.
- [12] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio In Processing*, vol. 13, no. 5, Part 2, pp. 1035–1047, 2005.
- [13] J. London, "Cognitive constraints on metric systems: Some observations and hypotheses," *Music Perception*, vol. 19, pp. 529–550, 2002.
- [14] A. Semjen, D. Vorberg, and H.-H. Schulze, "Getting synchronized with the metronome: Comparisons between phase and period correction," *Psychological Research*, vol. 61, no. 44-55, 1998.
- [15] D. Eck, "Beat tracking using an autocorrelation phase matrix," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2007, pp. 1313–1316.
- [16] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, pp. 39–58, 2001.
- [17] A. Robertson, A. M. Stark, and M. D. Plumbley, "Real-time visual beat tracking using a comb filter matrix," in *Proc. International Computer Music Conference*, 2011.
- [18] M. Puckette, T. Apel, and D. Zicarelli, "Real-time audio analysis tools for Pd and MSP," in *Proceedings of International Computer Music Conference, San Francisco.*, 1998, pp. 109–112.
- [19] G. Aschersleben, "Temporal control of movements in sensorimotor synchronization," *Brain and Cognition*, vol. 48, no. 1, pp. 66–79, 2002.

¹ <http://www.eecs.qmul.ac.uk/andrewr~>

TOWARDS A GENERATIVE ELECTRONICA: HUMAN-INFORMED MACHINE TRANSCRIPTION AND ANALYSIS IN MAXMSP

Arne Eigenfeldt

School for the Contemporary Arts
Simon Fraser University
Vancouver, Canada
arne_e@sfu.ca

Philippe Pasquier

School of Interactive Arts and Technology
Simon Fraser University
Surrey, Canada
pasquier@sfu.ca

ABSTRACT

We present the initial research into a generative electronica system based upon analysis of a corpus, describing the combination of expert human analysis and machine analysis that provides parameter data for generative algorithms. Algorithms in MaxMSP and Jitter for the transcription of beat patterns and section labels are presented, and compared with human analysis. Initial beat generation using a genetic algorithm utilizing a neural net trained on the machine analysis data is discussed, and compared with the use of a probabilistic model.

1. INTRODUCTION

The goal of this research is to create a generative electronica using rules derived from a corpus of representative works from within the genre of electronica, also known as electronic dance music (EDM). As the first author and research assistants are composers, we have approached the problem as a compositional one: what do we need to know about the style to accurately generate music within it?

EDM is a diverse collection of genres whose primary function is as dance music. As such, the music tends to display several key characteristics: a constant beat, repeating rhythmic motives, four beat measures grouped in eight measure phrases. Despite these restrictions, a great deal of variety can be found in other elements within the music, and can define the different genres – the specific beat pattern, the overarching formal structure, the presence and specific locations of the breakdown (the release of tension usually associated with the drop out of the beat) – and it is these variations that create the musical interest in each track.

The primary goal of this work is creative. We are looking for methods – many of which are borrowed from MIR – that can be used both for offline analysis, as well as real-time generation in performance: we are not interested in genre recognition or classification. Our initial research is concerned with the analysis of a corpus from both a bottom-up (e.g. beat patterns) as well as top-down (e.g. formal structures) perspective, as both are defining characteristics of the style. Although some generation has

already been undertaken, creative use of these analyses will be the future focus.

2. RELATED WORK

Little research has been done exclusively upon EDM, with the exception of Diakopoulos et al. [1], who used MIR techniques to classify one hundred 30-second excerpts into six EDM genres for live performance using a multi-touch surface. Gouyon and Dixon [2] approached non-electronic dance music classification using a tempo-based approach.

Automatic transcription of polyphonic music is, as Hainsworth and MacLeod suggest, one of the “grand challenges” facing computational musicology [3]. Klapuri gives an excellent overview of the problem [4].

Research specifically into drum transcription has recently been undertaken [5, 6, 7], including a very thorough overview by FitzGerald [8]. The parsing of compositions into sections from audio data has been researched as well [9, 10, 11, 12, 13].

Our research is unique in that it is carried out by composers using a combination of two of the standard live performance software tools, MaxMSP and Ableton Live, and is specific to electronic dance music.

3. DATA COLLECTION

One hundred tracks were chosen from four styles of EDM: *Breaks*, *Drum and Bass*, *Dubstep*, and *House*. The selection of these styles were based upon a number of factors: they are produced for a dance-floor audience and display clear beat patterns; the styles are clearly defined, and significantly different from one another; there is common instrumentation within each of the separate styles; they are less complex than some other styles.

Individual tracks were chosen to represent diverse characteristics and time periods, ranging from 1994-2010, with only four artists being represented twice. The tracks contain many common formal and structural production traits that are typical of each style and period.

Breaks tempi range from 120-138 beats per minute (BPM), and is derived from sped-up samples of drum breaks in Soul and Funk music which are also commonly associated with hip-hop rhythms. Off-beats occur in the hi-hat, similar to *House*, with many parts being layered to add variety. The beat is moderately syncopated, empha-

sizing two and four. Notable artists in this genre are Crystal Method, Hybrid, and Stanton Warriors.

Drum and Bass (D&B) has a tempo range of 150-180 BPM, with a highly syncopated beat containing one or more sped-up sampled breakbeats. As the name suggests, the bass line is very important, most often a very low frequency (sub-bass) sampled or synthesized timbre. Notable artists in this genre are Dom & Roland, Seba, and Klute.

Dubstep has a tempo range of 137-142 BPM, with a half-time feel that emphasizes the third beat (rather than two and four). It tends to be rather sparse, with a predominant synthesized bass line that exhibits a great deal of rhythmic low frequency modulation, known as a “wobble bass”. Notable artists in this genre are Nero, Skream, and Benga.

House has a tempo range of 120-130 BPM, with a non-syncopated beat derived from Disco that emphasizes all four beats on the kick, two and four on the snare, and off-beats in the hi-hat. *House* music typically involves more complex arrangements, in order to offset the straight-forward repetitive beat, and often has Latin and Soul/R&B music influences, including sampled vocals. Notable artists in this genre are Cassius, Deep Dish, and Groove Armada.

Each recording was imported into Ableton Live¹, and, using the software’s time-warp features, and adjusted so that each beat was properly and consistently aligned within the 1/16 subdivision grid. As such, each track’s tempo was known, and analysis could focus upon the subdivisions of the measures.

4. BEAT ANALYSIS

Initial human analysis concentrated upon beat patterns, and a database was created that listed the following information for each work:

- tempo;
- number of measures;
- number of measures with beats;
- number of unique beat patterns;
- length of pattern (1 or 2 measures);
- average kicks per pattern;
- average snare hits per pattern;
- number of instrumental parts per beat pattern;
- number of fills.

From these, we derived the following features:

1. kick density (number of measures with beats / (pattern length / kicks per pattern));
2. snare density (number of measures with beats / (pattern length / snares per pattern));
3. density percentile (number of measures / number of measures with beats);
4. change percentile (number of measures / number of unique beat patterns).

In order to determine whether these were useful features in representing the genres, a C4 Decision-Tree (J48) classifier was run, using the features 1-4, above (note that tempo was not included, as it is the most obvious classi-

fier). The Decision-Tree showed that snare density and kick density differentiated *Dubstep* and *House* from the other genres, and, together with the change percentile, separated *D&B* from *Breaks*. The confusion matrix is presented in Table 1. Note that differentiating *Breaks* from *D&B* was difficult, which is not surprising, given that the latter is often considered a sped-up version of the former.

| | Breaks | Dubstep | D&B | House |
|---------|--------|---------|------|-------|
| Breaks | 0.75 | 0.05 | 0.12 | 0.08 |
| Dubstep | 0.04 | 0.96 | 0.00 | 0.00 |
| D&B | 0.33 | 0.00 | 0.59 | 0.08 |
| House | 0.00 | 0.00 | 0.00 | 1.00 |

Table 1. Confusion matrix, in percent, for kick and snare density, and change and density percentile.

While this information could not be used for generative purposes, it has been used to rate generated patterns. Actual beat patterns were hand transcribed, a task that is not complex for human experts, but quite complex for machines.

4.1 Machine Analysis: Beat Pattern Detection

In order to transcribe beat patterns, a Max for Live² patch was created for Ableton Live that transmitted bar, beat, and subdivision information to Max³, where the actual analysis occurred. Audio was analyzed in real-time using a 512 band FFT, with three specific frequency bands selected as best representing the spectrum of the kick, snare, and hi-hat onsets: 0-172 Hz (kick); 1 kHz-5kHz (snare); 6 kHz-16kHz (hi-hat). Frame data from these regions were averaged over 1/16th subdivisions of the measure.

Derivatives for the amplitude data of each subdivision were calculated in order to separate onset transients from more continuous timbres; negative values were discarded, and values below the standard deviation were considered noise, and discarded: the remaining amplitudes were considered onsets. The 16 value vectors were then combined into a 16x1 RGB matrix within Jitter, with hi-hat being stored in R, snare in G, and kick in B (see Figure 1).

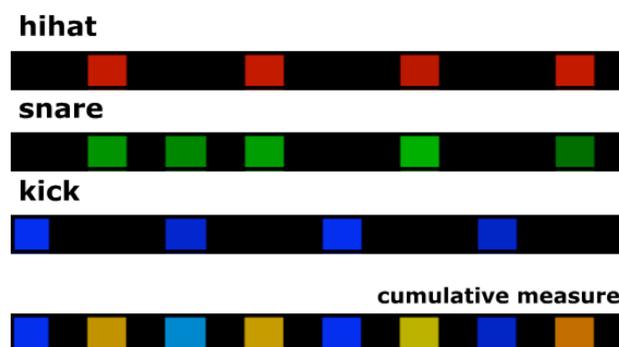


Figure 1. Example beat transcription via FFT, into 16x1 Jitter matrices. Brightness corresponds to amplitude.

¹ <http://www.ableton.com/>

² <http://www.ableton.com/maxforlive>

³ <http://cycling74.com/>

4.1.1 Transcribing Monophonic Beat Patterns

15 drum loops were chosen to test the system against isolated, monophonic beat patterns. These patterns ranged in tempo from 120-130 BPM, and consisted of a variety of instruments, with one or more kick, snares, tuned toms, hi-hats, shakers, tambourines and/or cymbals. Table 2 describes the success rate.

| Onsets | Transcriptions | Correct | Missed | False positives |
|--------|----------------|---------|--------|-----------------|
| 389 | 373 | 0.84 | 0.12 | .10 |

Table 2. Transcription success rates given 15 drum loops. Missed onsets tended to be of low amplitude, while false positives included those onsets transcribed early (“pushed beats”) or late (“laid-back beats”).

4.1.2 Transcribing Polyphonic Beat Patterns

Transcribing beat patterns within polyphonic music was less successful, mainly due to the variety of timbres that shared the same spectral regions. Furthermore, specific instruments, such as the bass in the low frequency, or synthesizer textures in the mid and high frequencies often used percussive envelopes that were difficult to discriminate from beat patterns (whose timbres themselves were not limited to noise).

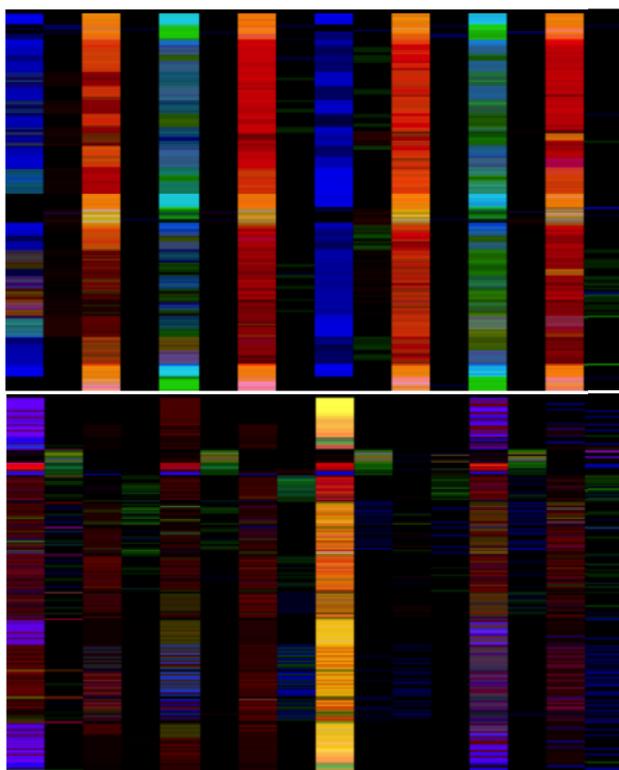


Figure 2. Two “beat fingerprints” for entire compositions: a single measure is presented as a horizontal line, with successive measures displayed top to bottom. Top, the House track “Funky Enuff”: blue indicates mainly kick, red hi-hat, demonstrating the “four to the floor” with hi-hat off-beats typical of House music. Bottom, the Dubstep track “Age of Dub”: yellow indicates snare and hi-hat, demonstrating the half-time feel of Dubstep.

Successive measures were accumulated into longer matrices, with the second dimension corresponding to the number of measures within the composition. This resulted in generating a track’s “beat pattern fingerprint”, visually displaying similarities and differences between individual compositions and genres (see Figure 2).

While the track fingerprints provided interesting visual information, a fair degree of noise remained, due to the difficulty in separating actual beats from other timbres that shared the same spectrum. For example, human analysis determined that the track in Fig. 3, top, contained only a single beat pattern, present throughout the entire duration; machine analysis calculated 31 unique kick patterns, 40 snare patterns, and 20 hi-hat patterns. As a result additional filtering was done, removing all onsets whose amplitudes were below the mean. This tended to remove false positive onsets from breakdowns.

5. BEAT GENERATION

Although generation is not the focus of our research at this time, some initial experiments have been undertaken.

5.1 Genetic Algorithm using a Neural Network

We trained a neural network (a multilayer perceptron with four nodes in the hidden layer) using patterns from the machine analysis described in Section 4.1. A fifth output was specified in which random patterns were fed in order for the neural network to be able to identify non-genre based patterns. The three individual patterns – kick, snare, hi-hat – were concatenated into a single 48 value floating point vector which was fed to the network.



Figure 3. Example beats created by the genetic algorithm using a neural network as fitness function; top, a Dubstep pattern; bottom, a House pattern.

A genetic algorithm was created in MaxMSP in order to generate a population of beat patterns, using the trained neural network as the fitness function. Individuals, initially randomly generated, were fed to the neural network, which rated each individual as to its closeness to the patterns of a user-selected genre (similarity being determined by an algorithm that compares weighted onsets and density); individuals ranked highest within the genre were considered strong, and allowed to reproduce through crossover. Three selection methods were used, including top 50%, roulette-wheel, and tournament selection, resulting in differences in diversity in the final population. Mutation included swapping beats, and removing onsets, as randomly generated patterns tended to be much more dense than required. Using an initial population of

100 individuals, a mutation rate of 5%, and evolving 20 generations, two examples are shown in Figure 3.

5.2 Genetic Algorithm using a Probabilistic Model

A second approach was explored within the genetic algorithm – the fitness function being the Euclidean distance from prototype patterns from each genre. These prototype patterns were calculated by accumulating onsets for all measures in every analyzed track, eliminating those scores below 0.2, and generating a probabilistic model (see Figure 4).



Figure 4. Proto-patterns for *Dubstep*, top, and *House*, bottom, based upon onset probabilities derived from machine analysis, with probabilities for each onset.

The machine analysis for these proto-patterns can be compared to those generated from the human analysis using the same criteria (see Figure 5). Note within *House*, only a single pattern occurs; the more active snare in the machine analysis suggests difficulty in the algorithm in separating percussive midrange timbres – such as guitar – from the snare.



Figure 5. Proto-patterns for *Dubstep*, top, and *House*, bottom, based upon onset probabilities derived from human analysis, with probabilities for each onset.

Additional mutation functions were employed that used musical variations, in which musically similar rhythms could be substituted – see [14] for a description of this process. Example patterns evolved using this model are given in Figure 6, using an initial population of 100 individuals, a mutation rate of 5%, and evolving 20 generations.



Figure 6. Three *House* patterns evolved using a genetic algorithm using machine-derived prototype patterns as fitness functions.

The use of a genetic algorithm in this second model to generate beat patterns might seem superfluous, given that a target is already extant. However, the result of the GA is a population of patterns that can be auditioned or accessed in real-time, a population that resembles the prototype target in musically interesting ways. No variation methods need to be programmed: instead, each pattern has evolved in a complex, organic way from the genre’s typical patterns. Lastly, unlike generating patterns purely by the probability of specific onsets found in the proto-pattern, new onsets can appear within the population (for example, sixteenths in the *House* patterns shown in Figure 6).

6. STRUCTURAL ANALYSIS

Within Ableton Live, phrases were separated by hand into different sections by several expert listeners (however, only one listener per track):

- Lead-in – the initial section with often only a single layer present: synth; incomplete beat pattern; guitar, etc.;
- Intro – a bridge between the Lead-in and the Verse: more instruments are present than the Lead-in, but not as full as the Verse;
- Verse – the main section of the track, in which all instruments are present, which can occur several times;
- Breakdown – a contrasting section to the verse in which the beat may drop out, or a filter may remove all mid- and high-frequencies. Will tend to build tension, and lead back to the verse;
- Outro – the fade-out of the track.

The structures found within the tracks analysed were unique, with no duplication; as such, form was in no way formulaic in these examples.

Interestingly, there was no clear determining factor as to why section breaks were considered to occur at specific locations. The discriminating criteria tended to be the addition of certain instruments, the order of which was not consistent. Something as subtle as the entry of specific synthesizer timbres were heard by the experts as sectional boundaries; while determining such edges may not be a difficult task for expert human listeners, it is extremely difficult for machine analysis. Furthermore,

many of the analyses decisions were debatable, resulting from the purely subjective criteria.

6.1 Machine Analysis: Section Detection

These fuzzy decisions were emulated in the machine analysis by searching for significant changes between phrases: therefore, additional spectral analysis was done, including:

- spectral energy using a 25 band Bark auditory modeler [15], which provides the spectral energy in these perceptually significant bands;
- spectral flux, in which high values indicate significant energy difference between frames, e.g. the presence of beats;
- spectral centroid, in which high values indicate higher overall central frequency, e.g. a full timbre, rather than primarily kick and bass;
- spectral roll-off, in which high values indicate the presence of high frequencies, e.g. hi-hats.

These specific features were found to be most useful in providing contrasting information, while other analyses, such as MFCC, 24 band Mel, and spectral flatness, were not as useful. Spectral analysis was done using Malt & Jourdan's zsa externals for MaxMSP⁴

As with beat pattern analysis, these features were analyzed over 1/16 subdivisions of the measure, and stored in two separate RGB Jitter matrices, the first storing the Bark data (3.3-16 kHz in R, 450-2800 in G, 60-350 Hz in B), the second the spectral data (Flux in R, Centroid in G, Roll-off in B). See Figure 7 for examples of these spectral fingerprints.

For each of the nine vectors (three each, for Bark, Spectral, and Pattern), derivatives of amplitude differences between subdivisions of successive measures were calculated; these values were then also summed and compared to successive measures in order to discover if section changes occurred at locations other than eight bar multiples⁵. Having grouped the measures into phrases, phrase amplitudes were summed, and derivatives between phrases calculated; as with pattern recognition, negative values and values below the mean were dropped. This same mean value served as a threshold in scoring potential section breaks, as each phrase in each of the nine vectors were assigned positive scores if the difference between successive values was greater than this threshold (a new section) or below this value for subsequent differences (reinforcing the previous section change). Summing the scores and eliminating those below the mean identified virtually all section changes.

Sections were then assigned labels. Overlaying the human analysis section changes with the mean values for the nine features, it was found that breakdowns had the lowest energy in the low and high Bark regions, while verses had the highest energy in all three Bark regions (when compared to the entire track's data). See Figure 8 for an example.

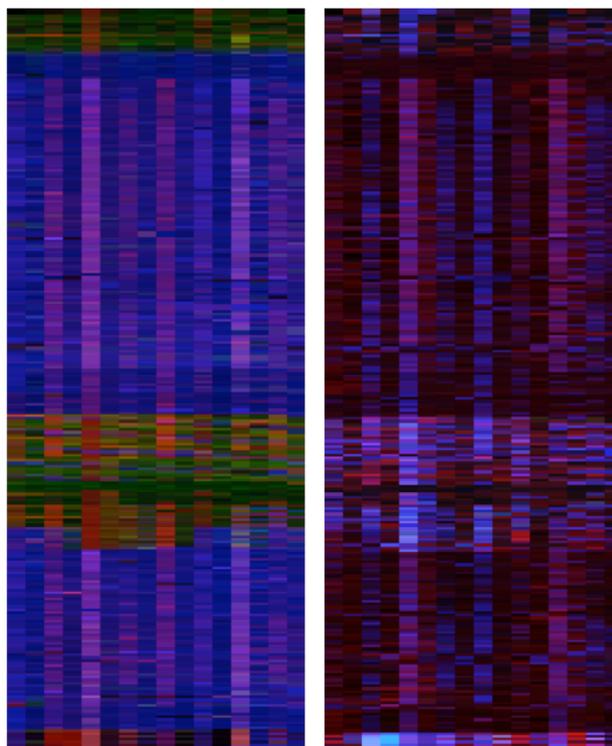


Figure 7. Spectral fingerprints for the *Breaks* track “Blowout”, with Bark analysis, left, and Flux/Centroid/Roll-off, right. The section changes are clearly displayed: in this track, both low and high frequencies are removed during the breakdown, leaving primarily the midrange, shown green in the Bark analysis.

Thus, those sections whose mean values for low and high Bark regions were below the mean of all sections, were tentatively scored as breakdowns, and those sections whose mean values for all three Bark regions were above the mean of all sections, were tentatively scored as verses.

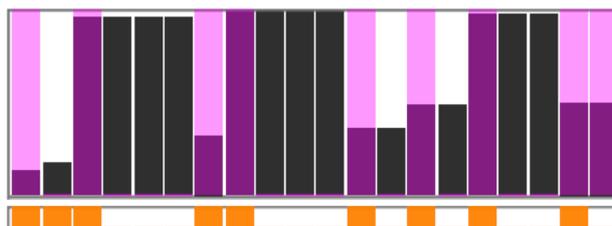


Figure 8. Mean amplitudes per section for twenty phrases for the *Breaks* track “Burma”. Gray represents the normalized amplitudes over the sections, pink represents the human-analyzed section divisions, orange the machine-analyzed section divisions, including a false positive in the lead-in.

A Markov transition table was generated from the human analysis of all sections, and the machine labels were then tested against the transition table, and the scores adjusted. Thus, a low energy section near the beginning of a track (following the lead-in) may have been initially labeled a breakdown, but the transition table suggested a higher probability for a continued lead-in. After all possi-

⁴ <http://www.e--j.com>

⁵ The most formal variation occurred in *House* music, ironically considered the most static genre.

ble transitions (forwards and backwards) were taken into account, the label with the highest probability was selected.

Each phrase within 32 tracks was machine labeled for its section: Table 3 presents the results. 5 tracks that displayed unusual forms (e.g. low energy verses) in the first three genres brought the scores down significantly.

| Genre | Phrases | Correct | Percentile |
|----------------|---------|---------|------------|
| <i>Breaks</i> | 174 | 122 | 0.70 |
| <i>D&B</i> | 264 | 189 | 0.72 |
| <i>Dubstep</i> | 184 | 124 | 0.67 |
| <i>House</i> | 152 | 122 | 0.80 |

Table 3. Success rate for machine labeling of sections.

7. CONCLUSIONS AND FUTURE WORK

Accurately creating music within an EDM genre requires a thorough knowledge of the model; while this knowledge may be implicit within composers, this research is the first step in making every decision based upon explicit analysis.

7.1 Improvements

Several improvements in the system are currently being made, including:

- Better beat detection involving comparing FFT matrix data between different regions of the tracks to determine similarities and differences within a phrase (i.e. comparing measure n and $n + 4$) and between phrases (n and $n + 8$).
- Incorporating fill detection to determine sectional change. Fills occur in the last 1, 2, 4, or even 8 measures of a section, and display significantly different features than the phrase, and lead to a significantly different section following.

7.2 Future Directions

Signal processing is an integral element of EDM, and we are currently involved in human analysis of typical DSP processes within the corpus, in determining representative processes, and their use in influencing structure. Similarly, pitch elements – bass lines, harmonies – are also being hand-transcribed.

Acknowledgments

Thanks to Christopher Anderson, Alan Ranta, and Tristan Bayfield for their analyses, and David Mesiha for his work with Weka. This research was funded in part by a New Media grant from the Canada Council for the Arts.

8. REFERENCES

- [1] D. Diakopoulos, O. Vallis, J. Hochenbaum, J. Murphy, and A. Kapur, “21st Century Electronica: MIR Techniques for Classification and Performance,” Int. Soc. for Music Info. Retrieval Conf. (ISMIR), Kobe, 2009, pp. 465-469.
- [2] F. Gouyon and S. Dixon, “Dance Music Classification: a tempo-based approach,” ISMIR, Barcelona, 2004, pp.501-504.
- [3] S. Hainsworth, M. Macleod, M. D. “The Automated Music Transcription Problem,” Cambridge University Engineering Department, 2004, pp.1-23.
- [4] A. Klapuri, “Introduction to Music Transcription,” in A. Klapuri & M. Davy (Eds.), Signal Processing Methods for Music Transcription, 2006, pp. 3-20.
- [5] J. Paulus, “Signal Processing Methods for Drum Transcription and Music Structure Analysis,” PhD thesis, Tampere University of Technology, Tampere, Finland, 2009.
- [6] O. Gillet, G. Richard, “Automatic transcription of drum loops,” Evaluation, 4, 2004, pp. 2-5.
- [7] O. Gillet, G. Richard, “Transcription and Separation of Drum Signals From Polyphonic Music,” IEEE Transactions On Audio Speech And Language Processing, 16(3), 2008, pp.529-540.
- [8] D. Fitzgerald, “Automatic drum transcription and source separation,” PhD thesis, Dublin Institute of Technology, 2004.
- [9] M. Goto, “A Chorus Section Detection Method for Musical Audio Signals and Its Application to a Music Listening Station,” IEEE Trans. Audio, Speech, and Lang. Proc. 14(5), 2006, pp. 1783-1794.
- [10] N. Maddage, “Automatic Structure Detection for Popular Music,” IEEE Multimedia, 13(1), 2006, pp.65-77.
- [11] R. Dannenberg, “Listening to Naima: An Automated Structural Analysis of Music from Recorded Audio,” Proc. Int. Computer Music Conf. 2002, pp.28-34.
- [12] R. Dannenberg, M. Goto. “Music structure analysis from acoustic signals,” in D. Havelock, S. Kuwano, and M. Vorländer, eds, Handbook of Signal Processing in Acoustics, v.1, 2008, pp. 305-331.
- [13] J. Paulus, “Improving Markov Model-Based Music Piece Structure Labelling with Acoustic Information,” ISMIR, 2010, pp.303-308.
- [14] A. Eigenfeldt, “The Evolution of Evolutionary Software: Intelligent Rhythm Generation in Kinetic Engine,” in Applications of Evolutionary Computing, Berlin, 2009, pp.498-507.
- [15] E. Zwicker, E. Terhardt, “Analytical expressions for critical-band rate and critical bandwidth as a function of frequency,” J. Acoustical Society of America 68(5) 1980: pp.1523-1525.

THE CLOSURE-BASED CUEING MODEL: COGNITIVELY-INSPIRED LEARNING AND GENERATION OF MUSICAL SEQUENCES

James Maxwell

Simon Fraser University SCA/SIAT
jbmaxwel@sfu.ca

Philippe Pasquier

Simon Fraser University SIAT
pasquier@sfu.ca

Arne Eigenfeldt

Simon Fraser University SCA
arne.e@sfu.ca

ABSTRACT

In this paper we outline the Closure-based Cueing Model (CbCM), an algorithm for learning hierarchical musical structure from symbolic inputs. Inspired by perceptual and cognitive notions of *grouping*, *cueing*, and *chunking*, the model represents the *schematic* and *invariant* properties of musical patterns, in addition to learning explicit musical representations. Because the learned structure encodes the formal relationships between hierarchically related musical segments, as well as the within-segment transitions, it can be used for the generation of new musical material following principles of *recombinance*. The model is applied to learning melodic sequences, and is shown to generalize perceptual contour and invariance. We outline a few methods for generation from the CbCM, and demonstrate a particular method for generating ranked lists of plausible continuations from a given musical context.

1. INTRODUCTION

1.1 Music Perception: Specificity and Invariance

The music perception and cognition literature has long acknowledged the existence of categories of perceptual change used by listeners to build mental representations of musical forms. The term “contour” has been used to describe *directionality* in perceptual change [1, 2, 3] and studies have shown contour to be a primary attribute used in the short-term recognition of basic musical patterns, and the comprehension of musical structure [4, 5, 6]. At a more detailed level of description, there are also categories of *invariance*, which express the perceptual similarity of patterns which are quantitatively *dissimilar*. For example, the music descriptor “pitch interval” is invariant across changes of absolute pitch level, just as the category of “rhythm” is invariant across changes in tempo. Finally, at the most detailed level of description, there are quantitatively “identical” percepts—specific pitches, for example. These three levels of perception share a complex interaction during music listening, which appears to develop with age and musical experience (for an overview, see Dowling [7]).

Copyright: ©2011 James Maxwell et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1.2 Probability or Planning?

Ever since the publication of Shannon’s “Information Theory”, musicians and researchers have focused on the information-theoretic properties of musical structure (in particular, the “Markov property” [8]) when considering the type of formal continuity demonstrated by a given musical “style.” Although the importance of such information-theoretic properties cannot be denied, we suggest that they are *effects* of musical thinking, not necessarily generative factors. For the CbCM, we propose an alternative approach founded on ideas from music perception, cognition, and memory. Although the CbCM demonstrates information-theoretic properties similar to other related models, it will be seen that the learned structure allows for an approach to generation unlike that applied in conventional Markov models.

Like our earlier MusicDB [9], the CbCM was designed as a ‘musical memory’, to be used in a larger interactive composition system. Aspects of the CbCM can be compared to Lartillot’s models for motivic extraction [10, 11], but whereas those models focus on pattern detection, the CbCM emphasizes hierarchical structure and sequence generation. The CbCM also shares some similarity with Deutsch & Ferøe’s hierarchical pitch representation [12], though the formalism differs considerably. Following a detailed discussion of the model, we will give examples of its implementation in a composition environment called *ManuScore*, where it is used to generate sorted lists of phrase continuations from a given musical context.

The CbCM makes no claims of biological plausibility. Rather, it takes theoretical ideas from the cognitive science of music as jumping-off points in the formulation of a model for hierarchical sequence learning and generation.

1.3 Ideas and Terminology

The CbCM has been designed with reference to a number of principles of musical memory:

1. *Association*: The process by which events that occur in close temporal succession form connections in memory.
2. *Cueing*: The process through which “one memory cues another memory with which it has formed an association” [13].
3. *Closure*: “When some aspect of the acoustical environment changes sufficiently, a boundary is created”

[13]. The perception of this boundary is referred to as *closure*.

4. *Grouping*: “The tendency for individual items in perception to seem related and to bond together into units” [13]. In the context of music, events tend to be *grouped* around points of perceptual *closure*.
5. *Chunk*: “Chunks are small groups of elements (5-9) that, by being frequently associated with one another, form higher-level units, which themselves become elements in memory” [13].

1.4 Two Dimensions of Hierarchy

Hierarchical models of musical form, like Lerdahl & Jackendoff’s “Generative Theory of Tonal Music” (GTTM) [14] organize musical materials into segments of increasing duration, building “motifs” at the lowest levels, “phrases” and “sections” at higher levels, and finally complete compositions at the highest levels.

However, there is another type of hierarchy in music which is perhaps not immediately apparent, and which has not generally been applied in computational models of music learning and generation; a dimension relating to the *specificity* of perceptions. When describing the pitch sequence {C4 G4}, many computational models will represent this sequence by enumerating its various *attributes*; the pitches C4 and G4, the melodic interval of 7 semitones, and perhaps the contour (“+”). This conception tends to give equal weight to each attribute in the music representation. We propose that the *invariant* categories of interval and contour are not merely *attributes* of an event, but rather represent hierarchically prerequisite states, such that each pitch in a sequential context *requires* an interval, and each interval *requires* a contour. Such a conception of hierarchy is similar to Conklin’s notion of “subsumption” [15], though our implementation expresses this idea explicitly in the CbCM topology (see Figure 1). The primacy of contour in short-term melodic recognition and the adaptability of melodic recognition to changes in absolute pitch level [1, 6] support this general conception of hierarchy.

With this in mind, the CbCM employs a music representation with three hierarchical levels of specificity: 1) **Schema**, the fundamental level pertaining to the detection of perceptual change (i.e., *pitch contour*), 2) **Invariance**, which captures relative quantities like those described by *pitch intervals*, and 3) **Identity**, which deals with the absolute quantitative values of the percepts themselves. Note that the application of such concepts need not be limited to pitch material; “contour”, for example, could be applied to rhythmic *augmentation* and *diminution*, or to changes in harmonic *tension* or *density*.

2. OVERVIEW OF THE MODEL

2.1 General Design

The CbCM can be conceptualized as a graph, the structure of which is learned from a time series of symbolic musical inputs. The graph is hierarchical and concurrent, so that states of the graph are represented by one or more active

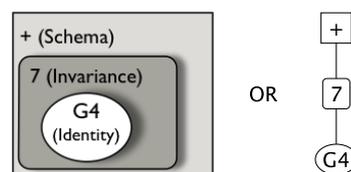


Figure 1. “Nesting” nodes of increasing specificity.

nodes. Concurrency in the CbCM reflects the manner in which musical structure unfolds simultaneously along multiple hierarchical dimensions. For example, a common statement like “*at the end of the third verse*” implies a location along three concurrent temporal dimensions: “at the **end**”—a position at the level of a musical *phrase*, “of the **third**”—a location in the complete form of the song, and “*verse*”—a position at the level of a musical *section*.

The states of the CbCM are ordered along two hierarchical dimensions; a *vertical* dimension, and a *nesting* dimension. The *vertical* dimension is explicitly sequential, and is arranged into one or more levels, each of which corresponds to a level of formal organization similar to those proposed by models like the GTTM. The *nesting* dimension organizes states and their substates according to the three levels of hierarchical specificity mentioned above: **Schema**, **Invariance**, and **Identity**. Considering these relationships under the formalism of Hierarchical State-Machines, we can say that **Identity** states are *implicitly Invariance* states, which are *implicitly Schema* states, as suggested by Figure 1.

For the sake of simplicity, we will use the term “levels” to refer to locations along the *vertical* dimension, and the term “states” (substates/superstates) to refer to locations along the *nesting* dimension.

2.2 Preprocessing and “Closure”

The CbCM builds its specific structure based on the notion of *closure*; i.e., the delineation of formal boundaries by the perception of musical change. However, it does not define or calculate the parameter over which change is detected. Rather, this value—referred to as the *closureSignal*—must be provided with the stream of input events. The CbCM detects significant changes in the *closureSignal*, and uses these changes to define segments in the learned model. This design is convenient, as it decouples the segmentation *criteria* from the hierarchical learning process; i.e., the specification of the *closureSignal* can be tailored to the type of input (melodic, rhythmic, harmonic, etc.).

A preprocessing step must be used to calculate the *closureSignal* for each input, in consideration of the type of input and the specification of an appropriate *closure* measure for that type. By isolating different input types in this manner, we acknowledge Snyder’s notion of “soft closure” (though we will refer to this generally as *closure*). In our implementation, we are learning/generating melodic pitch sequences, and use calculations for melodic pitch expectancy based on Margulis [16], and rhythmic expectancy based, in part, on Desain [17]. Details of the preprocessing used in the current study are given in Section 4.2.

Computationally, the CbCM is realized as a network, the nodes of which are arranged into one or more levels, each of which contains a sequence of states (and their substates). Each node on a level has a single parent, and each substate has a single superstate, so that the sequential structure within a given level forms a tree, as does the state structure.

2.3 Learning and Inference in the CbCM

The CbCM is an online learner, and thus always runs its inference step *before* proceeding with learning. Each level of the CbCM has a single *rootNode* and a single *contextNode*. The *rootNode* does not store any information, but rather serves as an initial state from which edges to other nodes can be searched and/or created. The *contextNode* acts as a pointer to the current state on a given level. It is updated with each input, and thus progresses through its level as musical transitions are perceived. Each time a given node becomes the *contextNode*, a *counts* variable is updated, for use during generation. Because each level has its own *contextNode*, the model progresses through all levels *simultaneously*, reflecting the notion of concurrency described in Section 2.1.

Since the *contextNode* acts as a pointer to the current state of the CbCM, inference involves a search through the *contextNode*'s attached edges for a transition matching the input. Failure to find a matching transition at the current state forces the search to be repeated at the superstate. Thus, failure to match the **Identity** state (*pitch*) forces a search of the **Invariance** state (*interval*), and failure to match the **Invariance** state forces a search of the **Schema** state (*contour*), as shown in Figure 2 (in the diagram, *contextNodes* are represented in white, light grey nodes represent superstates, and dark grey nodes represent hypothetical learned states). This pattern of defaulting toward the superstate when a transition cannot be found characterizes the *generalization* process used by the CbCM.

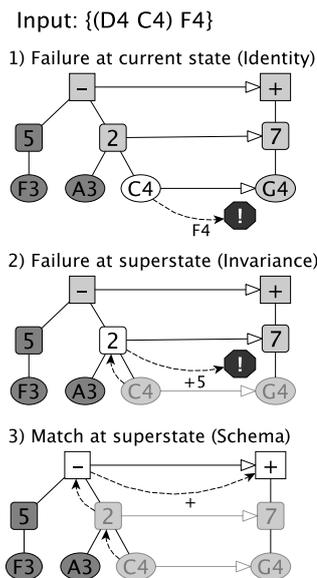


Figure 2. Generalizing to the deepest applicable state.

Learning in the CbCM involves the construction of a hierarchical network representing the states and transitions

embodied by a set of source works. At initialization, the CbCM has a single level with its *contextNode* set to the *rootNode*. Since only the *rootNode* node exists, the current state is assumed to be **Schema**. Learning cannot proceed without an initial context, so the first input is ignored. When processing subsequent inputs, the CbCM first performs inference, as described above. Since no transitions can be found, the model proceeds with learning. Learning a new transition involves adding a new node, expressing the greatest *specificity* possible, to the network. With only the *rootNode* in place, the CbCM extracts the **Schema** information from the input, creates a new **Schema** state node, and connects its edge to the *rootNode*.

The pattern for learning new nodes/states is shown in Figure 3. If the current state has no transitions to the input, only the **Schema** state can be learned, as shown in the 1st iteration of Figure 3. If an appropriate learned state can be reached via the current state's superstate, then the input can be added as a substate of the reached state, as in the 2nd iteration of Figure 3. This process continues until the **Identity** state representation has been learned (3rd iteration). In a trained CbCM, the edges connecting nodes represent *associations*, with the strongest associations being made between **Identity** states (since **Identity** states can only be formed through repeated exposure to a particular transition).

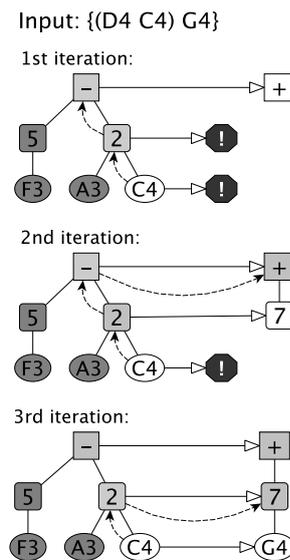


Figure 3. Learning states in order of reachability.

2.3.1 Closure and CbCM Structure

As mentioned in Section 2.2, the specific structure of the CbCM is determined through changes in *closure*, the value of which is calculated during preprocessing. The node itself stores the *closureSignal* value, which is updated each time the node becomes the level's *contextNode*. The input *closureSignal* is monitored at each time step, and moments of *decreasing* closure are used as indicators of formal change; i.e., when the *closureSignal* of the current input falls below that of the *contextNode*, a boundary is formed. During inference, this boundary forces the CbCM to limit its search for transitions to those edges with connections

into *higher level* nodes. During learning, the boundary marks the end of the current segment, forcing the learned node to be added to a *higher level in the model*, as shown in Figure 4.

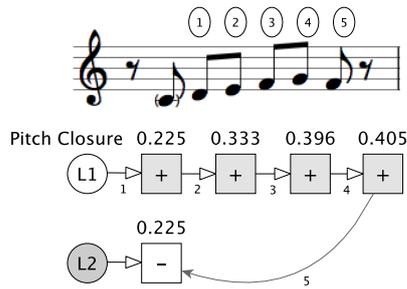


Figure 4. A decrease in *closureSignal* forces learning to a higher level.

Nodes created at higher levels thus represent *chunk boundaries* [13], marking the beginnings of perceptual segments. Each higher level node *cues* an L1 node, so that the continuation of a higher-level node is *always a sequence of L1 nodes*. This can be seen in Figure 5, which shows the *cueing* connection (edge 6) made between the L2 contextNode (-) and the new L1 node (-). In the learning algorithm, this pattern is achieved by setting the level's contextNode back to the rootNode every time a segment boundary is detected. When the following E4 is received (Figure 5), the search at L1 is carried out on the rootNode (which is now the contextNode). Since no descending Schema transition has been learned, a new node is added to the rootNode. At the end of a single pass through the input sequence in Figure 5, the model has learned two contour segments: {+ + + +} and {- -}.

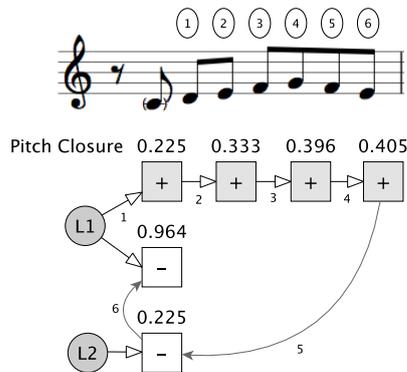


Figure 5. Learning reverts to L1 after a higher-level transition.

When learning creates nodes at higher levels (i.e., beyond L2) the process follows the same general pattern. If L1 detects a segment boundary, the CbCM attempts to learn the input at L2, but if L2 also detects a segment boundary, learning is passed to L3, and so on.

In order to clarify the formal relationship between higher-level nodes, we make an additional across-level connection when adding nodes above L2. This connection passes from the contextNode of the current segment's level to the newly learned higher-level node. We refer to this connection as

a “formal cue”, since the transition it describes is never directly output during generation; it serves only to form a *cueing* relationship between *chunk boundaries*, thus establishing hierarchical structure.

A diagram of a pitch CbCM trained on a *single pass* of the opening theme from Bach's BWV 846 (Fugue) is given in Figure 6. In the diagram, the contour symbols (+, -) indicate **Schema** states, the round-bracketed numbers indicate **Invariance** substates, and the square-bracketed numbers indicate **Identity** substates. The curved arrows from L1 indicate across-level *cues*, and the curved arrows connecting higher levels back into L1 show the cueing function of *chunk boundaries*. The *formal cues* in the model are labelled with italics (F1, F2, and F3).

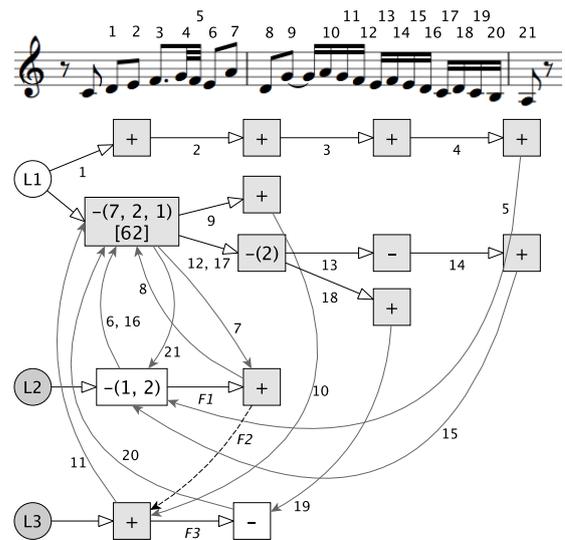


Figure 6. A CbCM for pitch, trained on a single pass of the theme from Bach's BWV 846 (Fugue).

2.3.2 A Note About Trained CbCM Structure

A few observations can be made about the CbCM, considering it as a graph—or rather, a composite of hierarchically-related graphs—in which the nodes at each state (**Schema**, **Invariance**, **Identity**) form *directed graphs* with the following properties:

1. For each level in the model a tree subgraph exists, describing the work/corpus at a particular level of temporal/formal organization.
2. The L1 tree represents the set of all perceptually grounded segments in the corpus.
3. Each level above L1 also represents a set perceptually grounded segments, but at higher level of formal structure (the segments contained by higher levels are analogous to “time span reductions” in the GTTM).
4. The L1 tree is the intersection of all higher-level subgraphs in the model (i.e., all higher levels have paths through L1).

5. *Formal cues* ensure that all higher-level nodes also form a separate subgraph describing the relationship between all perceptual (**L1**) segments in the corpus.
6. For every higher-level node L_{n_x} there exists at least one path through **L1**, of *length* > 1 , which terminates in an adjacent higher-level node L_{n_y} .
7. In a trained CbCM there exists at least one path ${}_xP_y$, in each state graph (**Schema, Invariance, Identity**), describing the complete sequence of transitions in a particular source work.

3. GENERATION FROM THE CBCM

The CbCM is a hierarchical encoding of musical information designed with an emphasis on sequence generation. One of our design priorities was to link the learning and generation processes in a cognitively grounded manner. To do this, we modelled the algorithms on ideas from Logan’s “Instance Theory” of learning. Logan’s theory proposes that novices initially approach a problem with a “general algorithm that is sufficient to perform the task”, but that with repeated exposure they eventually learn to “respond with a solution from memory” [18]. The theory explains the iterative nature of learning and the rapid increases in efficiency observed with repeated exposure to the conditions of (and solutions to) a given problem.

Given the learning algorithm described in Section 2.3, we can see how the CbCM might demonstrate *instance-based* learning in the context of generation. Consider the problem of trying to repeat the example transition {C4 G4}. After a single exposure, the CbCM would extract only contour information: “**Schema** = +”. When trying to repeat the transition, a “general algorithm” could thus proceed via heuristic search; i.e., using the **Schema** information to build a search space of pitches *above* C4. After a second exposure to the transition, the *interval* substate “**Invariance** = 7” would be learned, and generation could proceed according to a “rule”—i.e., ‘add 7 to the previous note.’ Finally, after a third exposure to the sequence, the CbCM would learn the substate “**Identity** = G4”, and could recall the transition directly from memory.

Of course, when executing the “general algorithm”, we must have some criteria for evaluating potential solutions in the search space. Here we turn to Ritchie’s notion of “quality.” Ritchie defines *quality* as a degree of membership in the set of objects that define a given “class”—a genre, for example [19]. Since the CbCM uses the *closureSignal* as a constraint on well-formedness, it follows that the calculated *closureSignal* of a given production will reflect the *quality* of that production. Our basic quality calculation is:

$$Q_s = f(\text{contextNode}, s) \quad (1)$$

$s \in S$

where Q_s is the quality rating of production s , $f()$ is a function used to calculate the *closureSignal* of a given transition (used during preprocessing), and S is the set of possible productions. The sequence $(\text{contextNode}, s)$ represents the transition from the *contextNode*’s value (e.g., a

MIDI note) to the value of a possible production (output) which, in this case, would also be a MIDI note.

In a trained CbCM, the *previously learned* transitions should provide ‘exemplars’ for good productions, so that the highest quality *novel* productions should result in *closureSignals* proximal to those produced by the learned transitions:

$$Q_n = 1 - \min_{\tau \in T} (|Q_n - Q_\tau|) \quad (2)$$

$n \in N$

$$N = S \setminus T \quad (3)$$

where Q_n is the quality rating of production n , N is the set of *novel* productions at the *contextNode*, and T is the set of productions made possible by the learned transitions at the *contextNode*. Depending on the input type, S may be infinitely large, as is the case with ‘unquantized’ rhythmic values. Thus, for practical purposes, we limit S to some finite set of discrete symbols—i.e., quantized rhythmic values or MIDI note numbers.

3.1 Generation by Planning

From the preceding discussion, it is clear that two basic approaches to generation are possible: 1) selecting transitions based on *quality*, and 2) selecting transitions probabilistically using the *counts* values of all reachable nodes. The first approach will be strongly influenced by the *closureSignal* calculation function, while the second approach will result in behaviour analogous to a variable-order Markov model. Of course, a combination of *quality* and probability could also be used.

However, the CbCM also provides a mechanism for generating segments through a process of *planning*. This is the approach used for generating continuations in our *ManuScore* composition environment, discussed in Section 4. In this approach, we consider transitions not at the note-to-note level, but rather at the phrase level. If we look at Figure 6 from the perspective of **L2**, we can think of **L2-Node1(-2)** as a *goal*, which can be successfully achieved by a specific *plan*—in this case, the **L1** sequence {+ + + +}. This segment can be generated by backtracking through edge 5 to **L1-Node4**, and along the chain of *parent* nodes, until we reach the *rootNode*, as shown in Figure 7.

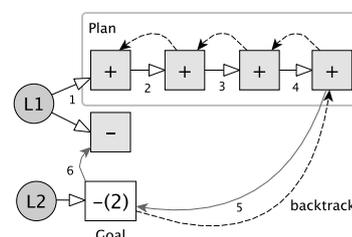


Figure 7. Building a plan through backtracking.

Generating plans from levels above **L2** follows an “unwinding” pattern, in which plans alternate between *output plans* (sequences of **L1** nodes) and *formal plans* (sequences of higher-level nodes). The *formal plans* are determined by backtracking along the *formal cues* discussed

in section 2.3.1. Since *formal plans* are sequences of higher-level nodes, they do not result in output, but rather provide *chunk boundaries* from which we can generate *output plans* (which can be directly output). As plans are unwound from the top, each generated *output plan* can be placed on a stack for subsequent evaluation.

3.2 Novelty and Quality in the CbCM

Ritchie defines *novelty* as the degree of *dissimilarity* of a production to existing examples of that genre [19]. Assuming that the CbCM has been sufficiently trained on appropriate examples (that is, its structure represents a “genre”), we can evaluate *novelty* in terms of the intersection between a given subgraph in the CbCM and the inferred subgraph of the production. For a given state/node, Equation 3 defines a *local set* of novel productions (transitions not yet learned in the current context) with which a more general measure of novelty can be calculated.

Of course, producing a single novel transition doesn’t guarantee “novelty” any more than exploiting a known transition prohibits it. Novelty is dynamic and cumulative. For this reason, it is useful to calculate the *potential* for novel generation at the current time step ($NLim^t$) cumulatively through time:

$$NLim^t = \frac{\sum_{i=0}^n \left(\frac{|N|}{|S|} \right)^{t-n}}{n+1} \quad (4)$$

Equation 4 can be used to determine $NLim^t$ over the entire CbCM, by continually incrementing n from the beginning of training/inference, or it can be used locally, by setting n to the sequential position of the *contextNode* on its branch. The *novelty rating* Λ_p^t of a given production p will thus be:

$$\Lambda_p^t = \begin{cases} NLim^t & \text{if } p \in N \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The use of $NLim^t$ can allow the model to determine the probability of generating a novel production in the current context, while Λ_p^t helps estimate the novelty of a given production from a trained CbCM.

4. IMPLEMENTATION

As mentioned above, we designed the CbCM to serve as a musical memory for a larger music composition system. In our *ManuScore* software, the user can request continuations of a given context, for which the CbCM provides a sorted list of possible options. The user can toggle through the ranked pitch and rhythm segments independently, auditioning each pitch/rhythm combination via MIDI.

4.1 Pitch and Rhythm Models

In order to model the phenomenon of *primitive grouping*—the independent grouping of pitch and rhythm information at low levels of auditory processing [13]—while at the same time modelling the interaction of pitch and rhythm in music cognition, we designed our melodic learning/generation system to utilize two CbCMs; one for pitch

and another for rhythm. Each model performs “primitive” segmentation at **L1**, by monitoring changes of *closure* in a single domain, but both models combine pitch and rhythmic closure information when segmenting at higher levels. In this way, the trained system retains a vocabulary of independent pitch and rhythm ‘motives’ at **L1** (which can be combined in novel ways during generation), but is also able to represent high-level musical form. During generation, we construct *plans* for each model independently, then pair up the pitch and rhythm plans for rendering to the score.

4.2 Preprocessing of the *closureSignal*

In our implementation, a preprocessing step determines the *closureSignal* for pitch using Margulis’ “Melodic Expectancy Model” [16]. We *do not* apply the model in its complete form, but use only the “basic expectancy” calculation, treating points of *decreasing* expectancy as instances of “*soft closure*” [13]. Because we are not modelling harmonic context in the current study, we omit the “stability” coefficient from Margulis’ formula (as recommended in [16]):

$$z = (p \times m) + d \quad (6)$$

where z is the expectancy, p is the “proximity rating”, m is the “mobility” (as in Margulis’, 2/3 for repetitions, 1 otherwise), and d is the “direction rating.” For details see Margulis [16]. We scale z to real a number in the range [0,1] and use it as our *closureSignal* for pitch.

To calculate rhythmic closure, we use a combination of rhythmic proximity and rhythmic expectancy. For the expectancy calculation we use Desain’s “basic expectancy” from his “(De)composable Theory of Rhythm” [17]. For the proximity calculation we use a simple exponential function:

$$p = 1.0 - \frac{x^2}{n^2} \quad (7)$$

where p is the calculated proximity, x is the IOI time, and n determines the temporal window over which connectivity will be maintained—we set this value to 6000 milliseconds to approximate the span of short-term memory [13]. Because any duration of silence separating events decreases their perceptual connectivity [13], we apply a further scaling for events separated by a silence greater than 800 milliseconds.

$$p = \mu - \frac{\mu \times (x - 800)^2}{n^2} \quad (8)$$

where μ determines the initial scaling amount, x is the separation time, and n determines the temporal window over which the scaling will occur (5200 ms in our model). We set μ to 0.7, so that any separation greater than 800 ms incurs an immediate 70% reduction in connectivity. If $p < 0$ we set it to zero.

The final rhythmic *closureSignal* z is the product of the proximity and expectancy:

$$z = p \times s \quad (9)$$

5. MODEL TRAINING AND TESTING

We trained *ManuScore* on a monophonic arrangement of the Fugue from Bach's BWV 846, from the Well-Tempered Klavier. Our choice to limit the source material to a single work allowed us to easily identify novel productions. In making the arrangement we tried to maintain as much of the work's formal integrity as possible, while reducing the four-part setting to a single monophonic voice¹.

To test both inference and generation, we generated output phrases as *continuations* of a given context. A melodic fragment was entered by hand, and the CbCM performed inference on the fragment and updated its state accordingly. Generation was then initiated from the **L2 contextNode** of both models, and segments were formed via backtracking (Section 3.1). The backtracking process halted when the inferred state was reached; i.e., we backtracked to the point where the input fragment ended. All segments generated in this manner were sorted by summing the *counts* values of their constituent nodes. When determining the *count* attributed to a given state/node, we summed the *counts* values of all implicit superstates (i.e., for an **Identity** plan node, we included the *counts* of its **Invariance** and **Schema** superstates).

6. RESULTS

As an initial test, we entered the first two pitches of the training work, which produced the continuation in Figure 8-A. Here, the top staff represents the input context and the bottom staff represents the CbCM's continuation. The greyed-out note (F4) in the continuation indicates that the rhythmic value was not part of a generated plan, but rather was determined probabilistically in order to provide IOI values for all notes in the generated pitch sequence². The label on the generated segment "P 1/7 - R 1/2" indicates that the CbCM is displaying the first of seven pitch options combined with the first of 2 rhythm options. Although the CbCM in *ManuScore* does infer/generate rhythmic IOI values, it does not handle note durations, so we extended the 'sustain bars' in the generated output by hand, for the sake of clarity. Toggling to continuation "P 2/7 - R 2/2" we get the pitch/rhythm combination shown in Figure 8-B, which is a *novel* sequence not contained in the source work.

In Figure 8-C, we transposed the context fragment up a tritone to {F#4 G#4}, producing a *context* not found in the source work. The output is essentially the same, generating 7 pitch options and 2 rhythm options, but has been transposed up a tritone. This spontaneous transposition indicates that the CbCM has 'defaulted' to the **Invariance** state during inference and modified its continuations accordingly. A similar test is shown in 8-D, except that in this case we entered a 4-note fragment which followed the *contour* of the original, but not the interval sequence. The

¹The arrangement can be viewed at: http://rubato-music.com/home/Media/Bach_846_Fugue_monophonic.pdf

²It is worth note that this IOI is incorrect; in the source work, the F4 falls on the downbeat immediately following the G4.

Figure 8. Continuations generated by the CbCM in *ManuScore*.

production correctly completes the contour by continuing with a {+2 -2} pattern. Inspection of the pitch CbCM during inference revealed that the transitions {C4 F#4} and {F#4 B4} from the context were inferred at the **Schema** state, as expected, while the transition {B4 C5} was inferred at the **Invariance** state. Since the source work has a "+1" transition {E4 F4} in the same context, this behaviour was also expected, indicating that the CbCM was able to transition to a more *specific* substate.

For our next test, we entered a longer context running from the beginning of the source work to half-way through measure 5. The first option produced from this longer con-

text is shown in Figure 8-E. The sequence produced, {G4 F#4 G4 A4}, is once again a novel segment, which does not appear in the original. It provides a valid continuation of the context, and is of particular interest since it maintains the modulation to the dominant (alteration of F4 to F#4) introduced in the previous measure. Option “P 3/3 - R 3/8”, produced from the same context, produces a quotation of the original.

Finally, we entered a completely novel, but still idiomatic context fragment, resulting in the production shown in Figure 8-F. The continuation is stylistically appropriate and is not a direct quotation from the source work.

7. CONCLUSION

The continuations generated by the CbCM in this preliminary test suggest that the model may be capable of generating context-sensitive “quotations” from the training set, in addition to reasonably well-formed novel productions.

The training process showed the expected pattern of learning, achieving complete training (i.e., learning all **Identity** states, as discussed in Section 2.3) after 3 passes over the source work. After training, both the pitch and rhythm CbCMs created 4 hierarchical levels. The pitch model created 181 **Identity** state nodes, 68 **Invariance** nodes, and 35 **Schema** nodes, over 274 transitions. Because the pitches comprising the source work are represented only by the **Identity** nodes (**Invariance** and **Schema** states are *implicit*), we use the count of **Identity** nodes for the calculation of compression, resulting in a 66% compression ratio.

8. FUTURE WORK

Our next step with the CbCM will be to evaluate performance with a larger body of source works. We are also interested in examining more closely the segmentation produced by our melodic expectancy-based approach. Although melodic expectancy seems a reasonable candidate for a *closureSignal*, the determination of an ideal *closure* calculation remains an open question. There is also a great deal of room for investigation into the various methods for generation offered by the structure of the CbCM as a memory model. We are currently developing a modular cognitive architecture for music, using the CbCM as a form of long-term memory.

Acknowledgments

This research was made possible, in part, by a grant from the *Social Sciences and Humanities Research Council of Canada*.

9. REFERENCES

- [1] W. Dowling, “Context effects on melody recognition: Scale-step versus interval representations,” *Music Perception*, vol. 3, no. 3, pp. 281–296, 1986.
- [2] D. Deutsch, *The psychology of music*. Academic Pr, 1999.
- [3] D. Levitin, “Memory for musical attributes,” *Foundations of cognitive psychology: Core readings*, pp. 295–310, 2002.
- [4] W. Dowling and J. Bartlett, “The importance of interval information in longterm memory for melodies,” *Psychomusicology: Music, Mind and Brain*, vol. 1, no. 1, 2008.
- [5] A. Lamont and N. Dibben, “Motivic structure and the perception of similarity,” *Music Perception*, vol. 18, no. 3, pp. 245–274, 2001.
- [6] J. Edworthy, “Interval and contour in melody processing,” *Music Perception*, vol. 2, no. 3, pp. 375–388, 1985.
- [7] W. Dowling, *The development of music perception and cognition*. Foundations of Cognitive Psychology. Cambridge: MIT Press, 1999.
- [8] C. Ames, “The markov process as a compositional model: a survey and tutorial,” *Leonardo*, vol. 22, no. 2, pp. 175–187, 1989.
- [9] J. Maxwell and A. Eigenfeldt, “The musicdb: A music database query system for recombination-based composition in max/msp,” in *Proceedings of the 2008 International Computer Music Conference*, 2008.
- [10] O. Lartillot, “A musical pattern discovery system founded on a modeling of listening strategies,” *Computer Music Journal*, vol. 28, no. 3, pp. 53–67, 2004.
- [11] O. Lartillot and P. Toiviainen, “Motivic matching strategies for automated pattern extraction,” *Musicae Scientiae*, vol. 11, no. 1 suppl, p. 281, 2007.
- [12] D. Deutsch and J. Feroe, “The internal representation of pitch sequences in tonal music,” *Psychological Review*, vol. 88, no. 6, p. 503, 1981.
- [13] B. Snyder, *Music and memory: an introduction*. The MIT Press, 2000.
- [14] F. Lerdahl, R. Jackendoff, and R. Jackendoff, *A generative theory of tonal music*. The MIT Press, 1996.
- [15] D. Conklin and M. Bergeron, “Feature set patterns in music,” *Computer Music Journal*, vol. 32, no. 1, pp. 60–70, 2008.
- [16] E. Margulis, “A model of melodic expectation,” *Music Perception*, vol. 22, no. 4, pp. 663–713, 2005.
- [17] P. Desain, “A (de) composable theory of rhythm perception,” *Music Perception*, vol. 9, no. 4, pp. 439–454, 1992.
- [18] G. Logan, “Toward an instance theory of automatization,” *Psychological review*, vol. 95, no. 4, pp. 492–527, 1988.
- [19] G. Ritchie, “Assessing creativity,” *Institute for Communicating and Collaborative Systems*, 2001.

EVALUATION OF SENSOR TECHNOLOGIES FOR THE RULERS, A KALIMBA-LIKE DIGITAL MUSICAL INSTRUMENT

Carolina Brum Medeiros and Marcelo M. Wanderley

Input Devices and Music Interaction Laboratory,

Centre for Interdisciplinary Research in

Music Media and Technology,

McGill University, Montreal, QC, Canada

carolina.medeiros@mail.mcgill.ca, marcelo.wanderley@mcgill.ca

ABSTRACT

Selecting a sensor technology for a Digital Musical Instrument (DMI) is not obvious specially because it involves a performance context. For this reason, when designing a new DMI, one should be aware of the advantages and drawback of each sensor technology and methodology. In this article, we present a discussion about the Rulers, a DMI based on seven cantilever beams fixed at one end which can be bent, vibrated, or plucked. The instrument has already two sensing versions: one based on IR sensor, another on Hall sensor. We introduce strain gages as a third option for the Rulers, sensor that are widely used in industry for measuring loads and vibration. Our goal was to compare the three sensor technologies according to their measurement function, linearity, resolution, sensitivity and hysteresis and also according to real-time application indicators as: mechanical robustness, stage light sensitivity and temperature sensitivity. Results indicate that while strain gages offer more robust and medium sensitivity solution, the requirements for their use can be an obstacle for novice designers.

1. INTRODUCTION

In the context of DMIs, stability and robustness are often discussed as ways of evaluating a device's behaviour and as a means of expressing the desired parameters and features required for performance. In most cases, these requirements differ from those expected in the laboratory environment. Often, DMIs require some adaptation after performer's practice sessions, through technical and player's evaluation. Also, stability and robustness are required qualities for learning and practice process.

The Rulers, an instrument developed in 2004 by David Birnbaum [1] [2], has undergone two versions and numerous performances, but none of these versions have produced a stable instrument. The first version was based on using Infrared (IR) sensors to measure the distance between the sensor and the cantilever beam. The second ver-

sion used Hall sensors to measure the same variable. In this paper, we present a third version of The Rulers that uses strain gage (SG) sensors to measure the strain on the beam. The goal of this work is to provide more sensitivity and stability for The Rulers, as well as present a possible application of strain gages on DMIs.

Although widely used in industrial applications, strain gage sensors have not been widely employed in DMIs, possibly due to their relative complexity when compared to more popular sensors for measuring force and pressure, like Force Sensitive Resistors (FSR). This paper introduces the use of the strain gages in DMIs, and compares their performance relative to other, more popular, sensor technologies applied to The Rulers.

2. THE RULERS

The instrument was designed to induce the gesture of plucking or bending the free end of seven beams. The lengths of each of the seven aluminum cantilever are various. Therefore, each beam oscillates at a different frequency when plucked. This provides visual and passive haptic feedback to the player, otherwise the oscillations are not used for acoustic purposes. The beams' motion - created by a variety of gestures - is measured by sensors, which output signal is mapped to control a computer-based sound synthesizer. Figure 1 shows The Rulers being played.



Figure 1. The Rulers

The expected and unexpected beam movements executed can be approximate by the Euler-Bernoulli beam equations [3]. These equations, considering some boundary conditions, provide information about the displacement and the strain at each point of the beam. Nevertheless, the full

mathematical description of the riddle would only be possible by numerical simulation of the physical model or by making use of real-time control/monitoring of some variables in conjunction with the agreement between a good mechanical design and its implementation. The sensors placement is showed in the Figure 2.

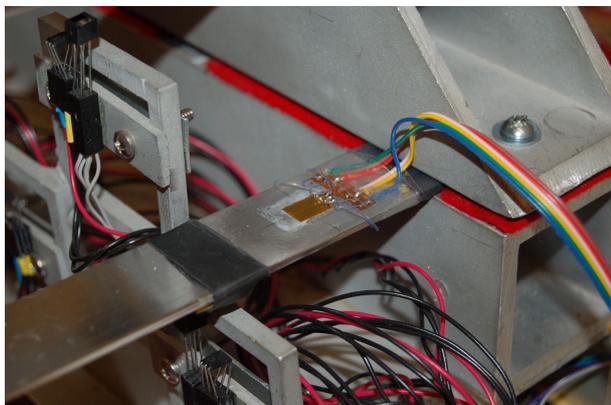


Figure 2. Sensors placement

Usually, approximating the load as a constant concentrated force and considering the fixed point as a clamp, it's possible to predict the strain at any point of the beam. Consequently, taking in consideration the area of concentrated strain and the SG's maximum relative strain rating, the optimal point for the sensor application can be determined.

2.1 Infrared Sensor Version

This version's conception is based on measuring the distance between the infrared system (IR transmitter + IR receptor) and the beam. Therefore, the displacement value at the free end is obtained from the distance between sensor and beam next to the fixed end. Even though, the relation between the displacement and the measured distance is considered linear, this can not be assured due to the mechanical construction of the instrument.

The infrared system contains a transmitter (photodiode) and a receptor (phototransistor). The former transmits infrared waves towards the direction of the beam. The beam, partially reflects the incoming wave towards the receptor's direction. The receptor will then sense a delayed and lossy wave compared with the transmitted wave.

This sensor presents a low complexity and low set up time, but, its response is not linear. This downside must be partially compensated by calibration and software corrections, otherwise the response might contain substantial errors [4]. Linearization methods can be applied both in the hardware [5] and in the software [6, 7] implementations. Also, as stage lights radiate substantial infrared waves, interference may occur in performance contexts.

2.2 Hall Sensor Version

In this version, a Hall sensor was used to measure the magnetic field strength produce by a magnet located in the bottom of the beam. The field strength generates, according to the Hall Effect, a voltage on the sensor terminals. This

voltage is directly proportional to the field and actually reflects the distance between the sensor and the beam.

Accordingly to the datasheet, the output response in Volts is linear in respect to the magnetic field in Gauss [8]. Otherwise, as the configuration is unipolar (just one magnet as magnetic field generator), the relation between magnetic field and distance is quadratic, i.e., non-linear [9]. A disadvantage encountered on this method is related to the instrument's mechanical features: small translational movements affects the alignment between sensor and magnet and slope error measurements can be expected [9].

2.3 IR and Hall Sensors Solutions

As presented above, IR and Hall sensor responses are based on the distance between the sensor and the beam. The measurements are taken near the fixed end of the beam to avoid disturbing the performer's gestures. From the measurement perspective, this spot is not optimal in terms of displacement range, limiting the input quantity range for the sensors.

In addition, some approximations would be required to consider the relation between the displacement exerted at the free end and the displacement near to the fixed end as a linear ratio. There are two sources of non-linearity when measuring the beam's movements using IR and Hall sensor. The first one is the inner sensitivity of the sensors to theirs primary quantity: infrared radiation (IR sensors) or indirectly the distance sensor-magnet (Hall sensor), which are both related to the distance between the sensor and the beam next to the fixed end. The second one is the ratio between the measured quantity (distance between the sensors and the beam) and the quantity of interest (displacement applied to the free end).

Non-linear measurement functions are generally not appreciated as they demand correction, linearization and high sensitivity along the measuring interval. This non-linear transfer functions might be perceived as an unreliable relation between the performer's gestures and the sound being controlled by the measured signals, when a decent correction process is not executed.

3. STRAIN GAGE TECHNOLOGY

From physics, it's known that once there is a force applied to a certain area, there will be a resulting stress given by $\tau = F/A$, where τ is the stress, F is the force and A is the area. For each particular load pattern, two quantities stand out: strain and displacement. Strain is the relative displacement of rigid body particles, which can be described by its *engineering normal* form as: $\epsilon = \Delta L/L$, where ϵ is the engineering normal strain, L is the original length of the material and ΔL is the length variation. In contrast, displacement (δ) is a deformation that implies change in shape or position.

Materials can react to the stress elastically or plastically, depending on its own characteristics and on the load. In the elastic regime, the relation between stress and strain is linear and there is no residual displacement when the force is released. Under this condition, the linear relation

between stress and strain is given by the Young Modulus or Elastic Modulus (E) given by: $E = \tau/\epsilon$.

In the current study, we will be measuring strain, through the use of strain gages. These sensors are sensitive to strain, which is maximized in the neighbourhood of the fixed end. The strain at this point can be correlated to the displacement at the free end. This ratio, for static loads, small displacements and negligible weight of the beam, can be considered quadratic in respect to the length. This comes from the fact that, regarding these boundary conditions, the displacement along the beam and the strain are respectively:

$$\delta = F * L^3 / E * b * h^3 \quad (1)$$

$$\epsilon = 6 * L * F / E * b * h^2 [3] \quad (2)$$

where b is the beam's width and h is the height of the beam.

3.1 Types of Strain Gages (SG)

There are a variety of models, principles of operation and measuring intervals that should be considering when selecting a certain model for each specific application. Among the electrical-type strain gages, there are the resistive SG (based on resistance changes) and the semiconductor SG (based on conductivity changes) [10]. The first one, also called metallic SG, experiences changes in its resistance when submitted to mechanical forces. The second one varies its resistance according to changes on its resistivity (piezoresistive effect), when strained. Their sensitivity, called Gage Factor (GF), is usually from 1.8 to 4.5 for metallic SG and from 40 to 200 for semiconductor SG [11]. Both sensor types are sensitive to temperature changes, especially the semiconductor one. Therefore, in real-time applications, without strict temperature control, employing semiconductor SG is not advisable, that's why a metallic SG was selected to perform this task.

3.1.1 Metallic Strain Gage Functionality

An electrical resistance of a conductor having length L , area A and resistivity ρ is:

$$R = \frac{\rho * L}{A} \quad (3)$$

If the wire experiences a longitudinal load, both its dimensions, L and A , and resistivity ρ will change at different ratios:

$$\frac{dR}{R} = \frac{d\rho}{\rho} + \frac{dL}{L} - \frac{dA}{A} \quad (4)$$

According to [11], for an isotropic conductor, within the elastic limit, the amount of resistance variations is

$$\frac{dR}{R} = \frac{dL}{L} * [1 + 2\nu + C * (1 - 2\nu)] = GF * \frac{dL}{L} = GF * \epsilon \quad (5)$$

where ν is the Poisson coefficient, C is the Bridgmann's constant.

All sensors are sensitive to so-called secondary quantities [12]. The main secondary sensitivity for metallic strain gages is temperature variation. Temperature affects the

strain gage performance in two ways. First, metal materials face dimension changes as stated by the thermal expansion coefficient.

Due to that, the strain gages are built over a foil material which thermal expansion coefficient is similar to the specimen thermal expansion coefficient. By doing that, no relative expansion between specimen and sensor is expected.

Secondly, temperature variations modify the resistance of the unstrained grid wire and the Gage Factor.

For compensating this effect, two solutions are commonly put in use: to conduct the experiments under a controlled temperature environment (what it is difficult in a stage environment); or to use *dummy* strain gages, that experience thermal strain but no mechanical strain. Thermal effects can then be compensated.

3.2 Strain Gage Application

The four steps for applying strain gages to measure strain in the instrument are: determine the bridge topology, design the conditioning circuit, apply the sensor into the specimen and perform the measurements.

3.2.1 Bridge Topology

Strain gage resistance variation, calculated by $\Delta R/R = GF * \epsilon$, are usually small enough for being measured with a reasonable resolution using voltage dividers. Due to this fact, Wheatstone bridge is employed, that consists on a balanced/unbalanced circuit.

These features are profitable for strain gage measurement as the disposition of elements defines whether they will have adding or subtracting contributions to the differential output voltage. In the current application, a *full bridge configuration for bending purposes* was used. This topology infers that there are four strain gages applied to the specimen: two of them installed on the top, two installed on the bottom.

Considering a one-way bending, two strain gages observe stretch and sense $+\epsilon$, while two observe contraction and sense $-\epsilon$. In addition, all of them experience thermal deformation, that can be considered the same (due to small dimensions) if there is no temperature gradient between top and bottom.

Finally, summing up all strain contributions, it's possible to calculate the resulting bridge output voltage as follows:

$$\begin{aligned} V_{out} &= V_{pw} * GF / 4 * \\ &(\epsilon_m^1 + \epsilon_{th}^1 - \epsilon_m^2 - \epsilon_t^2 + \epsilon_m^3 + \epsilon_t^3 - \epsilon_m^4 - \epsilon_t^4) \\ V_{out} &= V_{pw} * GF * \epsilon_m \end{aligned} \quad (6)$$

where V_{out} is the output voltage, V_{pw} is the power source voltage, GF is the Gage Factor and ϵ_x^n is the strain, where the upper indices mean the element number and the lower indices whether the strain is mechanical or thermal. The solution above takes into account the fact that all thermal strains are equal as well as all absolute values of mechanical strains are equal.

3.2.2 Conditioning Circuit

A conditioning circuit was designed to perform the following functions: zero the bridge, compensate lead wire resistance, amplify and adjust the scale for USB voltage patterns. These tasks are essential when dealing with small signals, like strain gage bridge output voltage. First, an analog zeroing process is executed by using two fixed resistors (tolerance 0.1%) and two trimpots (tolerance 1%). Adjusting coarse and fine trimpot resistors, it's possible to correct small disagreements that could be present when the sensors are unstrained.

Furthermore, lead wires resistance, specially on remote measurements, may have a great influence on the output response. For solving that, the six-wire measurement method was designed. Amplification and scaling is performed by a low-power differential operational amplifier. Note that the bridge can deliver positive and negative differential voltage outputs depending on the unbalance direction. For this reason, a voltage reference is summed up with the bridge output bringing the reference voltage output to 2.5 V (suitable for USB interfaces).

3.2.3 Sensor Application into the Specimen

Applying the strain gages into the specimen requires attention due to strain gage's delicate structure. Also, the installing process requires dirt-free environment and tools, as any impurity may degrade the strain gage grid and pads or affect the strain transfer from the specimen to the sensor.

Once specimen and sensor are cleansed and dried, the application point is selected. For selecting the point, there is a compromise between greater load concentration and maximum strain damage prevention. After selecting the point, signs are drawn to indicated the correct position to apply (a maximum alignment error of four degrees between top and bottom strain gages positions are acceptable).

Afterwards, the glueing process is done based on cyanoacrylate cold cure adhesive. Finally, in order to close the bridge, the pads are connected using wires with the same specifications and lengths. Ultimately, a resistance measurement test is done to verify the correct installation of the sensor.

4. EXPERIMENTAL DEVELOPMENT

The present experimental work consists on analog and digital design as presented in the Figure 3. Hardware and software were developed in order to efficiently evaluate the sensor output signals.

4.1 Measurement Chain

The measurement chain of the system is presented in the Figure 3 and described in the next sessions.

4.1.1 Hardware

- sensors: strain gages, Hall sensor and IR sensor;
- conditioning circuits:
 - **zero circuit:** zeroes the voltage output when the beam is unstrained, before executing measurements;

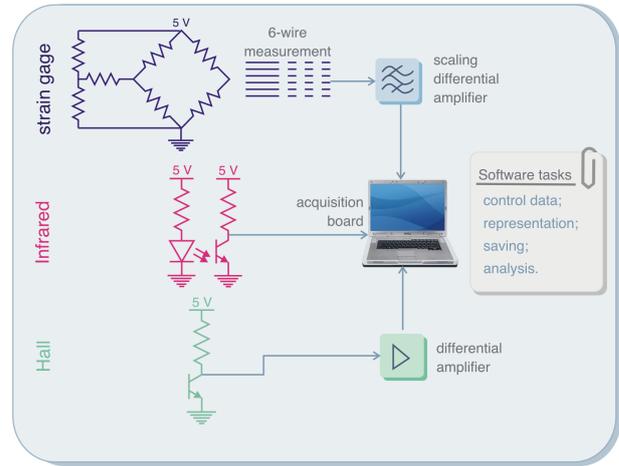


Figure 3. Measurement Chain

- **amplification:** amplifies the Wheatstone Bridge output and Hall sensor output using differential amplifiers;
- **scale adjustment:** adjusts the full-differential bridge output to the range (0 to 5) V, in order to be suitable for USB interfaces;
- **lead wire compensation:** compensates voltage losses through the power source wire by measuring the power source voltage through a six-wire bridge connection.
- acquisition board analog interface from National Instruments (8 channels, 24 bits, up to 100 k samples/s). Five channels were used:
 - power source voltage;
 - sensing of power source voltage through the bridge;
 - strain gage conditioning circuit output;
 - Hall conditioning circuit output;
 - IR sensor output.

4.1.2 Software

A .vi code (Labview Virtual Instrument software) was implemented for processing the data from the acquisition board. The software incorporates the following blocks and functionalities:

- acquisition board settings;
- data manipulation settings (user);
- input and output data;
- simple mathematical manipulations;
- data representation (graphics): before and after manipulation;
- data saving interface.

The possibility of saving the data allows the researcher to export this data to a specialized mathematical application where one can perform more complex data manipulations.

4.2 Measurement Procedure

As soon as the hardware is setup: wiring, zeroing and gain adjusting, the measurement procedure can be started. The steps followed for each set of measurement are briefly described above:

1. perform the zeroing procedure by software;
2. bend monotonically the beam towards one direction, take measurements of position and sensor outputs at each desired point;
3. once the last measurement point is reached, slightly exceed this point before starting the bending operation toward the other direction;
4. bend monotonically the beam toward the other direction, take measurements of position and sensor outputs at each desired point;
5. once the last measurement point is reached, exceed slightly this point before starting the bending operation toward the other direction;
6. repeat four times the operations 2 to 5.

5. RESULTS

5.1 Qualitative Comparison

Table 1 presents the qualitative indicators for each sensor. Some of these indicators require discussion as follows:

| QUALITATIVE COMPARISON | | Sensors | | |
|---------------------------|--|-------------------|----------------|-----------------|
| | | Strain Gages | Hall Sensor | IR Sensor |
| desired characteristics | linearity | linear | non-linear | non-linear |
| | large displacement (15 to 60) mm sensitivity | medium (constant) | low (variable) | high (variable) |
| | small displacement (0 to 15) mm sensitivity | medium (constant) | low (variable) | low (variable) |
| | force sensitivity | high | none | none |
| | mechanical robustness | low | medium | high |
| undesired characteristics | stage light sensitivity | negligible | negligible | high |
| | temperature sensitivity | medium | negligible | negligible |
| | assembly difficulty | high | medium | low |

Table 1. Qualitative Comparison among Sensor's Performances

force sensitivity: strain gages measure strain which is directly related to the force applied to the beam;

mechanical robustness: in this sense, mechanical robustness indicates the property of maintaining the instrument features through time and use. As strain gages are applied to moving parts, as well as their lead wires, this can imply fatigue or connection problems. In contrast, IR and Hall sensor don't have any moving parts in the sensing system. Besides, the Hall sensor was considered as having medium mechanical robustness because its operation are vulnerable to translational movements that can be performed resulting in nonalignment between Hall sensor and magnet;

assembly difficulty: it represents how difficult it is to set up data acquisition from the three sensor types. While strain gages require expertise to manipulate, to install and to have their signals conditioned, Hall sensors usually demand conditioning circuits and IR sensors' receptor can be directly digitalized.

5.2 Quantitative Comparison

The comparison results presented in this session are based on several measurements regarding the following characteristics:

- static and constant force;
- positive and negative displacements in relation to the steady-state position;
- the displacement dynamic cycle is a triangle waveform.

5.2.1 Measurement Function

This function is obtained by using both ascending and descending displacements. This function represents the output sensitivity to the input, where the input is the displacement (δ) and the output is the voltage signal value (V). This equations provide information about linearity, resolution and hysteresis.

When loading the data into a mathematical software, a graphical representation helps to estimate what is the best approximation for the function: linear, quadratic, exponential, sine waves summation or others. This procedure was done and yielded the best approximation curve for each sensor type, regarding the R-squared factor (R_{sqr}).

- **Strain Gage** — linear — $R_{sqr} = 0.9987$
 $V_{SG} = 0.01524 * \delta + 2.507;$
- **Hall Sensor** — quadratic — $R_{sqr} = 0.9670$
 $V_{HL} = -2.840 * 10^{-5} * \delta^2 - 2.131 * 10^{-3} * \delta + 0.2001;$
- **Infrared** — quadratic — $R_{sqr} = 0.9918$
 $V_{IR} = -4.937 * 10^{-4} * \delta^2 + 1.505 * 10^{-2} * \delta + 4.538.$

Some remarks about the low R_{sqr} for the Hall sensor approximation are necessary. As it will be explained in Session 5.2.5, this sensor yields great hysteresis, i.e., it is possible to observe multiple parallel curves shifted from each other in relation to the δ axis. Therefore, although the Hall sensor has a better approximation considering each

half-cycle separately (monotonic bending increasing), due to its hysteresis, the approximation tends to an average between the shifted curves. The best approximation curves for each sensor output are presented in the Figure 4.

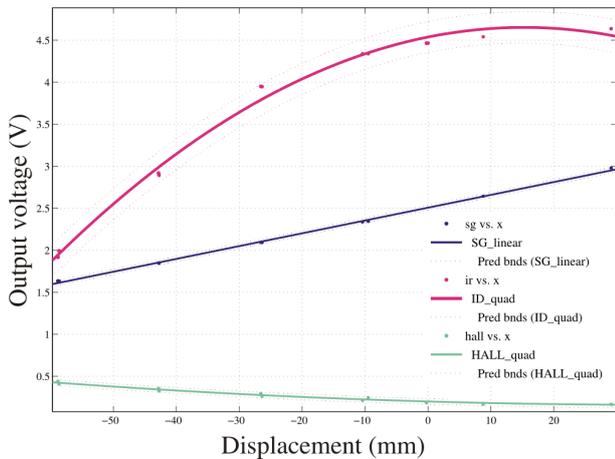


Figure 4. Best curve fitting for each sensor output

Both IR and Hall output response are non-monotonic, i.e., their output don't maintain a given order. Therefore, there is an ambiguity concerning displacements within the approximate ranges [0 to 30] mm for the IR sensor, and [-60 to -15] mm, for the Hall sensor, approximately (observe Figure 5). Unfortunately, this undesired condition is hard to compensate.

5.2.2 Linearity

A linear sensor output response is commonly desired for simplicity. A non-constant sensitivity along the measurement range requires a lookup table giving input and correspondent output values in order to obtain a verification curve. Unfortunately, usually it's hard to obtain a suitable control of the input to read as many points as necessary to obtain a reasonable error/uncertainty scenario.

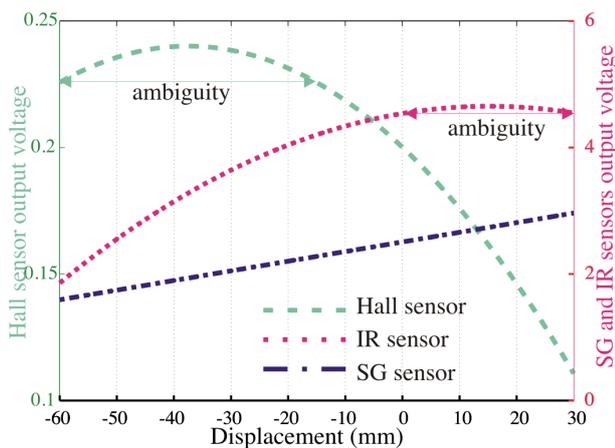


Figure 5. Quadratic Non-monotonic response of IR and Hall sensors

Fortunately, the measurement functions, once obtained by verification or calibration (better solution), can be loaded

in the firmware (microcontroller) or in the real-time sound controller software.

As the actual measurements are taken in the analog domain where no correction is applied, it's possible to observe that, for the IR sensor, the output is noticeably less sensitive next to the steady-state position, as show in the Figure 5.

5.2.3 Sensitivity

As a consequence of the measurement functions, it's possible to calculate the sensitivity for each sensor technology. Figure 6 presents the sensitivity for the three sensor types, i.e. the amount of variation observed in the output (in V) when a unitary variation of displacement (in mm) is performed. As it is expected, the IR and the Hall sensor have a non-constant sensitivity across the measurement range.

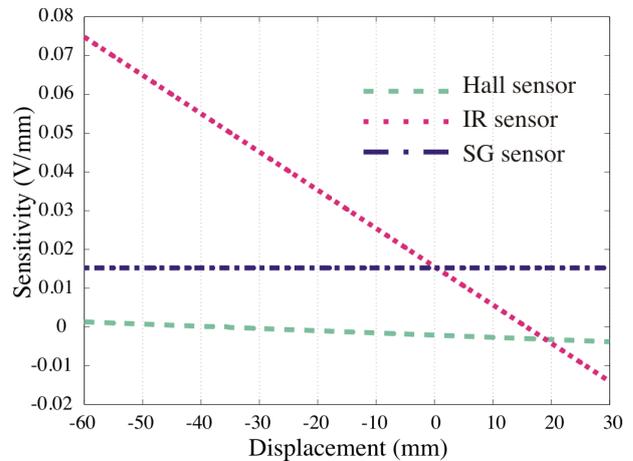


Figure 6. Strain Gage, IR and Hall sensor sensitivities across the measurement range

It's observable that depending on the beginning point for the displacement, the sensitivity varies for the IR and the Hall sensor, while the strain gage has a constant response across the measurement range. The IR presents a good performance concerning its sensitivity, especially within the range (-50 to 0) mm. Comparing the three sensitivities, the Hall sensor presents a poor sensitivity as its value is non-constant and it's the lowest one among the three sensor responses.

5.2.4 Resolution

The resolution indicates the "smallest change in a quantity being measured that causes a perceptible change in the corresponding indication" [13]. The following calculation is made based on the fact that the DMI will be connected to portable acquisition device whose analog to digital converter is, in general, not greater than ten bits resolution. High value for the resolution, i.e. a poor resolution means low value for the sensitivity. For example, the IR's low sensitivity within the interval (2 to 30) mm causes the IR sensor to have a worse resolution than the SG in this range. Figure 7 shows the resolution considering a 10-bit ADC computer interface. In order to better present this comparison, the lower graphic in the Figure 7 is a scaled version

of the upper one, where just resolutions lower than 5 mm are shown.

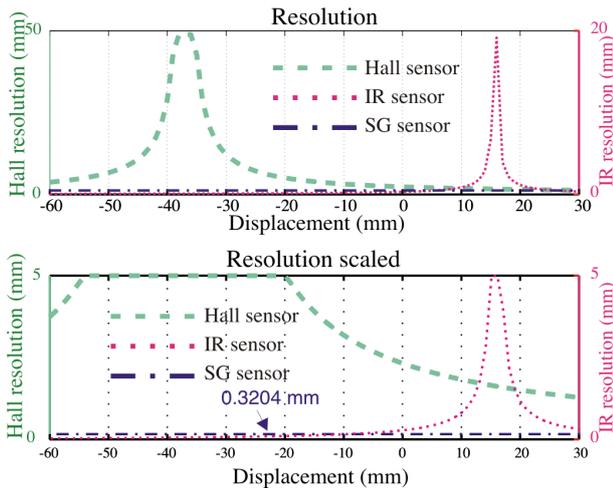


Figure 7. Strain Gage, IR and Hall sensor resolutions across the measurement range

5.2.5 Hysteresis

There are two hysteresis phenomena on this instrument. The first one is due to its mechanical conception causing the beam not to come back exactly to the same place when the force is relieved. This might be related to the poor affix at the fixed end and/or to the plastification of the beam's material. This problem is untreatable in this version of the instrument, demanding a new mechanical version with a clamped fixed end.

Below is represented the maximum hysteresis value ($\Delta\delta_{hist}$) for each sensor:

- **Strain Gage:** $\Delta\delta_{hist} = 1.01 \text{ mm}$;
- **Hall Sensor:** $\Delta\delta_{hist} = 16.86 \text{ mm}$;
- **Infrared:** $\Delta\delta_{hist} = 4.64 \text{ mm}$;

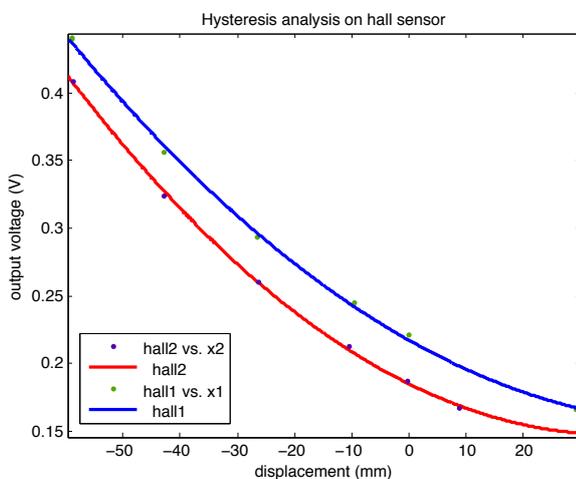


Figure 8. Hall Sensor Hysteresis Effect

The second one comes from hysteresis on the sensor operation. Different from the other sensor types, the Hall sensor presented a considerable hysteresis shift on its values (Figure 8). For calculating the maximum hysteresis value, the first and the last measurement points were disregarded.

6. DISCUSSION

After unveiling the characteristics of the three types of sensors and after comparing them, various parameters can be useful for sensor selection: responsiveness, robustness, price, availability, usability/complexity, compatibility to real-time and performance circumstances.

Concerning responsiveness, the main metrological indicator are the sensitivity and linearity. The IR sensor presents the best sensitivity over 75% of the measurement range, although this value is non-constant and decays drastically for small displacements. This implies that small performed vibrations and the damped oscillation of plucked movements are hardly sensed by the system resulting in poor possibilities of composing and playing under these conditions. The strain gage has a medium constant sensitivity over the measurement range, i.e its output is linear, while the Hall sensor has a small quadratic sensitivity. In terms of mapping sensor signals to control music, the non-linearity of Hall and IR sensor is a drawback as it requires algorithms to correct the output response and the ambiguity.

Referring to robustness, we have reports about the two first versions, based on IR and Hall sensors, given by technicians, composers and performers. When using Hall sensors, the mechanical design permits some translational movements that compromise the alignment between the magnet and sensor, changing the sensitivity. With IR sensors, they mention a strong interference of stage light on the response up to the point that it was necessary to cover the sensor area with a black fabric to diminish the problem. In the case of strain gages, we don't have substantial performance experience yet, but we can expect that some problems relating to connections and moving parts can occur and sensor and lead wires that are installed directed on the beam are subject to fatigue.

As IR and Hall sensors are more used on low-technology applications, they tend to have low price and good availability. Furthermore, they are straightforward to setup. The strain gages, however, present a high initial cost with accessories, tools and chemical products but the sensor itself is not expensive once one has all the materials for the application. The strain gages availability may also be an issue since they are sold directly by the manufacturer.

In terms of usability and complexity, strain gages are the most difficult to use, requiring some skills and special material to apply them to the instrument. The other two sensors, disseminate over both industrial and DIY contexts, are relatively easy to handle with. However, one should consider that for improving the performance of these sensors, some complex techniques of correction for linearity and ambiguity are required. Finally, these sensor should be carefully selected for a determined distance range.

7. CONCLUSIONS

As the hypothesis estimate and the qualifying test preview, strain gage sensors have a linear response that makes them an interesting sensing option for The Rulers. On the other hand, for large displacements the infrared sensor has a better sensitivity, but this behavior is undermined by its small sensitivity for small displacements and by its non-monotonic response. Finally, the drawbacks that exclude the Hall sensor as an optimal solution are its high hysteresis, wide ambiguity interval (non-monotonic) and low sensitivity, although they can be useful for measuring other distance ranges and for detecting movement directions. Ultimately, the selection should be done according to the composer and performer needs. After the comprehension of important characteristics of each sensor, this selection is easier and more conscious about the features, advantages and limitations of the selected method.

Acknowledgments

The first author would like to thank Capes/Brasil for a doctoral scholarship and also IDMIL / McGill and GRANTE / UFSC researchers for discussions. This work is partially funded by NSERC Discovery Grant and CFI.

8. REFERENCES

- [1] S. Ferguson and M. M. Wanderley, "The McGill Digital Orchestra: An Interdisciplinary Project on Digital Musical Instruments," *Journal of Interdisciplinary Music Studies*, vol. 4, pp. 17–35, 2010.
- [2] J. Malloch, D. Birnbaum, E. Sinyor, and M. M. Wanderley, "Toward a New Conceptual Framework for Digital Musical Instruments," *Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx-06)*, 2006.
- [3] A. S. Kobayashi, *Handbook on Experimental Mechanics*. John Wiley & Sons, 1993.
- [4] J. Dias Pereira, P. Silva Girao, and O. Postolache, "Fitting Transducer Characteristics to Measured Data," *Instrumentation Measurement Magazine, IEEE*, vol. 4, no. 4, pp. 26–39, Dec. 2001.
- [5] D. Patranabis, S. Ghosh, and C. Bakshi, "Linearizing transducer characteristics," *Instrumentation and Measurement, IEEE Transactions on*, vol. 37, no. 1, pp. 66–69, Mar. 1988.
- [6] H. Erdem, "Implementation of Software-based Sensor Linearization Algorithms on Low-cost Microcontrollers," *ISA Transactions*, 2010.
- [7] S. Khan, A. Alam, S. Ahmmad, I. Tijani, M. Hasan, L. Adetunji, S. Abdulazeez, S. Zaini, S. Othman, and S. Khan, "On the Issues of Linearizing a Sensor Characteristic Over a Wider Response Range," in *Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on*, May 2008, pp. 72–76.
- [8] *Datasheet SS49E/SS59ET Series: Economical Linear Position Sensor*, Honeywell.
- [9] "Hall Effect Sensing and Application," Honeywell, Application Note.
- [10] E. R. Miranda and M. M. Wanderley, *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. A-R Editions, 2006, ISBN 0-89579-585-X.
- [11] R. Pallas-Areny and J. G. Webster, *Sensor and Signal Conditioning*, 2nd ed. NY, USA: John Wiley & Sons, 2001.
- [12] P. Stein, "The Unified Approach to the Engineering of Measurement Systems for Test and Evaluation - a Brief Survey," in *Instrumentation and Measurement Technology Conference, 1996. IMTC-96. Conference Proceedings. 'Quality Measurements: The Indispensable Bridge between Theory and Reality'*, *IEEE*, vol. 1, 1996, pp. K1–28 vol.1.
- [13] *Vocabulary of Basic and General Terms in Metrology (VIM)*, Joint Committee for Guides in Metrology (JCGM) Std., Rev. 3rd, 2008.

BEATLED - THE SOCIAL GAMING PARTYSHIRT

Tom De Nies

Ghent University - IBBT

tom.denies@ugent.be

Thomas Vervust

Ghent University - CMST

thomas.vervust@ugent.be

Michiel Demey

Ghent University - IPEM

michiel.demey@ugent.be

Rik Van de Walle

Ghent University - IBBT

rik.vandewalle@ugent.be

Jan Vanfleteren

IMEC/Ghent University - CMST

jan.vanfleteren@ugent.be

Marc Leman

Ghent University - IPEM

marc.leman@ugent.be

ABSTRACT

This paper describes the development of a social game, BeatLED, using music, movement and luminescent textile. The game is based on a tool used in research on synchronization of movement and music, and social entrainment at the Institute of Psychoacoustics and Electronic Music (IPEM) at Ghent University. Players, divided into several teams, synchronize to music and receive a score in real-time, depending on how well they synchronize with the music and each other.

While this paper concentrates on the game design and dynamics, an appropriate and original means of providing output to the end users was needed. To accommodate this output, a flexible, stretchable LED-display was developed at CMST (Ghent University), and embedded into textile.

In this paper we analyze the characteristics a musical social game should have, as well as the overall merit of such a game. We discuss the various technologies involved, the game design and dynamics, a proof-of-concept implementation and the most prominent test results.

We conclude that a real-world implementation of this game not only is feasible, but would also have several applications in multiple sectors, such as musicology research, team-building and health care.

1. INTRODUCTION

In today's games, the social aspect has become more than just an extra feature. Game developers are incorporating social interaction as a key feature into their games, and are researching alternative ways for users to interface with the gaming platforms. BeatLED is a so called "social" game, and this paper will describe its key features, internal workings and applicability.

First we narrow down the general concept of a "social game", and how this concept relates to earlier work. We also motivate why the game was developed, and how it can be applied in different fields. Next, we present the various technologies and algorithms used to accommodate the game. The game dynamics are described, followed by an insight into the proof-of-concept implementation.

Copyright: ©2011 Tom De Nies et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

We also discuss the set-up and results of the various tests that were performed. These tests include algorithm performance tests and general test sessions with actual end users. Finally, the future plans and possibilities for the project are suggested and a conclusion is made.

2. SOCIAL GAME

2.1 What/Why?

Interaction - on-line or otherwise - has become a must for all recent games. Studies show that a large part of the latest console gaming generation considers the social aspect as the most important motivation to play [1]. Typical examples of *social games* are those developed for the Nintendo Wii consoles. When we examine these games, we are able to identify the following features:

Multiplayer Naturally, in order to be called "social", a game should facilitate more than one player. Ideally, easy, even dynamic expansion of the group should be possible. In the best case, one can choose the number of players arbitrarily.

Interactivity In order to be challenging and fun, the game should include a tight action-perception coupling, both between the players and the game, as among the different players themselves. A slow responding game, where users need to wait for feedback based on their actions, is not an option.

Intuitivity The game itself, its rules and its controls, should all be intuitive and easy to comprehend. During the gameplay, players should not be focusing on how to control the game, or try to comprehend its rules. Instead they should only be focusing on the general idea of the game, and on the other players. Keeping the controls natural not only allows the players to be fully absorbed by the game, but also makes it easy to introduce new players to the game, hence contributing to the social factor.

Motivation (to play in group) The game should not only encourage participation, it should also easily attract more players. For example, certain game modes could only be made available when a predefined number of players is reached. To make the game inherently more challenging with more players would be even better.

Social bonding The game should incorporate a certain *social bonding* factor. This means playing the game should be beneficial to the relationships between the players. This factor is also present in many of the team-building games, often seen in larger businesses.

During development of this game, it is important to keep these features in mind. They are also recognized in many console games, especially those intended for Nintendo Wii or Microsoft Kinect. We will attempt to incorporate all of these features in our social game.

The game we are building is centered around a widespread social activity, namely dancing to music. Dancing is by nature an intuitive and social activity, it allows you to interact with other people without conversation, and the larger the group, the more fun is guaranteed. All five of the above factors can already be discerned in that activity. Therefore, building a social game using music and movement is a logical step.

We will build a game that allows players to dance to music, and to receive a certain score, depending on how well they synchronize with the music, and with each other.

2.2 Similar Games & Applications

To describe all games based on music, dance and/or movement is beyond the scope of this text. We will highlight the most relevant and well-known examples.

2.2.1 Similar Games

Dance Dance Revolution DDR, by Konami, is probably the best-known dance game. Players receive instructions on a screen in the form of a sequence of arrows, each arrow corresponding with a specific movement. When they make the correct movement at the correct time, they receive a positive score. Their movements are recorded using a proprietary dance mat, as shown in figure 1. The number of players is limited, with a maximum of 2 or 4 players. Players are also very limited in movement, partly because they are obliged to follow the directions of the game and partly because of the limited space they have to dance.

Just Dance The spatial limitations of DDR are partially solved by Just Dance for Nintendo Wii. Here, players interact only with the controller, but again, they must watch the screen and follow the specific instructions of the game, as shown in figure 2.

Dance games for Kinect Another approach to dance games is presented by the Kinect, a camera-based game system recently developed by Microsoft. These games are very similar in functionality to the previously discussed games. Players have to follow the instructions provided on the screen, only now these instructions represent more complex movements. Using its depth-sensitive camera, the Kinect is able to track detailed movements of the players and match them to the instructions. However, this also implies that players need to remain inside the field of view of the camera.



Figure 1. DDR dance mat: players have to match their movements to arrows shown by the game.



Figure 2. Just Dance 2 for Nintendo Wii screen interface. The hand that holds the wii remote is highlighted.

The major difference between all currently available games and ours, is that our game takes into account the *mutual synchronization* between the movements of players. It also imposes *no limitations in space*.

2.2.2 Exergames

The games described above are so called *exergames*, since many people use them as a motivating means of physical exercise. DDR was even deployed in some schools, as part of the exercise program [2]. This shows that a social game such as the one discussed in this paper can also be applied as a means of exercise .

2.3 Previous Research

The previous (and first try at) implementation of this game, Sync-In Team [3], was used as a research tool, used to aid in research toward synchronization of movement and music. The game also allowed researchers to study social interaction between dancing people, a phenomenon known as *entrainment* [4].

This game captured the movements of 4 players, divided into 2 teams, using accelerometers. The players were dancing to a series of audio tracks. Using a simple Fourier Transform based algorithm, the tempo of the movements was calculated, and compared to the Beats Per Minute (BPM) of the music.

A team whose tempo lay close to the BPM, got an increase in score, while the score decreased for a team whose tempo was too far off.

Scores were projected onto the floor, using growing and shrinking patterns in different colors, each color corresponding with a team, as pictured in Figure 3.

This approach had some limitations. The algorithm used to calculate the tempo was based on an FFT (Fast Fourier Transform), using a time-window of at least 4 seconds.

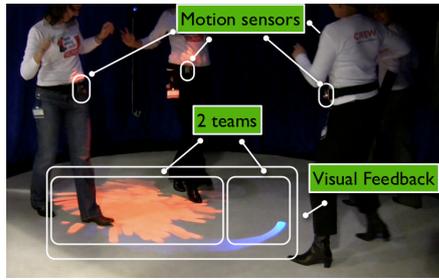


Figure 3. Sync-In Team score visualization using colored projection on the floor

This resulted in a poor response-time, and other limitations, such as no record of phase, or varying tempo. The game was limited to 4 players, and 2 teams. Also, the manner of output for scores limited the players in space and movement, forcing them to look down.

2.4 Goal

Our goal is to resolve the issues that presented with the original Sync-In-Team Game, while developing a new and improved interactive social game, BeatLED.

The game should meet the following requirements:

- Allow at least 4, and preferably an arbitrary number of players to dance to a series of audio tracks, divided into a number of teams.
- Capture their movement data and synchronize it with the selected music, using a synchronization algorithm that performs in nearly real-time.
- Output a score in an original way, not limiting the movement space of the players.

This last aspect will be realized using the hardware developed at CMST, namely a flexible, stretchable LED-Matrix, embedded into a t-shirt.

2.5 Motivation & Applications

A social game like ours can have numerous applications in all fields. Next to personal entertainment, the game's social bonding and motivating characteristics could be applied to business, interpersonal relationships and even medicine. Applications could include team-building sessions (corporate or treatment-wise) or rehabilitation.

Apart from this, developing such a game represents an interdisciplinary challenge, and is bound to uncover techniques that can be applied in other applications.

3. TECHNOLOGIES

3.1 Accelerometers

In order to synchronize the movements of the players to the music, we need a device that captures these movements. For this we use accelerometers. An accelerometer is a device that measures proper acceleration (relative to free fall). Acceleration values are measured on three axes, so each movement direction can be represented, as can be

seen in Figure 4, where the accelerometer is embedded in a Nintendo Wii remote. For use during the development stage, these Nintendo Wii remotes are chosen for their easy connectivity via bluetooth, their high availability and low cost.

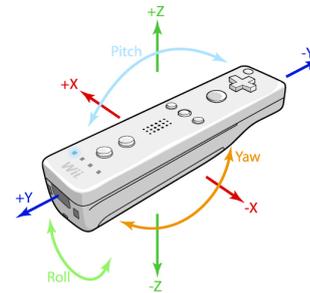


Figure 4. Nintendo Wii remote with acceleration axes x, y and z

3.2 LED-Display

For the visual output of the game, a flexible, stretchable LED-matrix was developed at the Center of Microsystems Technology at Ghent University.

The novelty lies in the integration of the LED-display into textile, and the possibility to send data to this display wirelessly. So far, all commercially available textile that includes electronics lacks one or more of the key properties available with the CMST LED-Display. CMST has access to an in-house lab with great facilities and know-how, which were applied to obtain a stretchable LED display integrated in a T-shirt, as shown in figure 5(a).

The main focus of the design, beside stretchability, is on size and power consumption, since this is a battery application. All the electronics needed to control the display and the wireless communication need to be small in comparison to the display itself.

The final circuit design was assembled on a flexible design and molded in silicone, rendering it some rigidity, as shown in Figure 5(b). This mold could then be attached inside the sleeve of the game T-shirt.

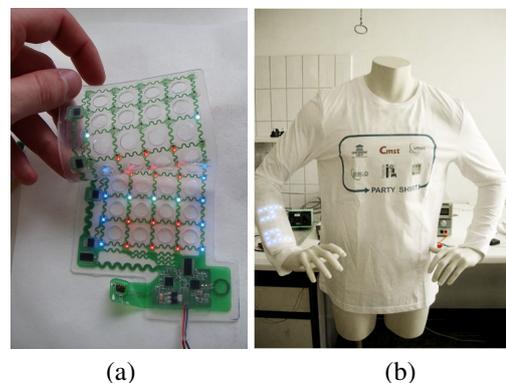


Figure 5. The finalized LED-display (a) silicone mold (b) embedded in T-shirt

4. SYNCHRONIZING MUSIC & MOVEMENT

4.1 Input Data Processing

Before any processing is performed, the input data needs to be shaped into a format which is usable by the synchronization algorithm. The acceleration data needs to be captured and filtered, and the music needs to be annotated, detecting the beats in the audio track.

4.1.1 Acceleration Data

The acceleration signal $A(x)$ provided by the accelerometers described in 3.1 is sampled at a fixed *sampling rate*, with each sample consisting of three real values, one for each axis, normalized between -1 and +1.

Due to the high sensitivity, the signal is bound to contain several involuntary, irrelevant movements, as well as some noise. To counter this, the signal is passed through a *low-pass filter* before any further processing. The simplest type of linear digital filter, a *Finite Impulse Response (FIR)* filter is used, designed to attenuate all frequencies higher than 10Hz.

4.1.2 Beat Detection

In contrast to the original Sync-In Team, where only the average Beats Per Minute (BPM) of the song was calculated, our application's algorithm needs more specific data. This data includes the locations of each beat within the track. Because developing a custom algorithm for this is not the focus of this research, we opt to utilize third party software for this, namely Beatroot [5].

Beatroot tracks the onsets of all beats in a .wav file, and outputs these onset timings (in ms) to a plain text file, which can be read by our synchronizing algorithm. Once the audio is analyzed by Beatroot, the obtained data can be used to calculate the *average BPM*.

$$bpm = \left\lfloor \frac{60}{I_b} \right\rfloor \quad (1)$$

In this formula, I_b represents the *mean beat interval*, defined as the average time between 2 successive beats, calculated as

$$I_b = \frac{L}{B} \quad (2)$$

with $L = \text{length of audio track in seconds}$ and $B = \text{number of beats detected}$.

4.2 Synchronizing Algorithm

The data described in the previous paragraph is now used to associate a score to the synchronization between the movement data and audio. To do this, a newly developed *peak detection algorithm* is used.

4.2.1 Score Array

First, the beat data supplied by the beat detection algorithm (4.1.2) is converted to a *score array* S .

We associate a positive score with every sample of the audio track within a certain acceptable tolerance o before and after a beat. While doing this, the same sampling rate f_s as the accelerometers is used, thus creating an array with

each element $S[t]$ corresponding to a sample of the movement data (with t the time of the sample). The acceptable tolerance is calculated as

$$o = \frac{I_{b_samples}}{m} \quad (3)$$

where $I_{b_samples}$ represents the average number of samples between two successive beats, calculated as $I_{b_samples} = \frac{f_s * 60}{BPM}$ and m an adjustable parameter used to determine the tolerance in which a positive score is given. The further away from the actual beat position, the lower the associated score will be. The same principle is used for the *off-beat* locations, exactly between two successive beats, since people tend to synchronize to these as well. A simple example of the creation of the score array is shown in figure 6. Note that this illustration is simplified for visibility reasons. The tolerance in the figure is chosen quite high (100 ms) and the sampling rate very low at 20 Hz. In the real implementation the sampling rate would be much higher (ca. 100 Hz).

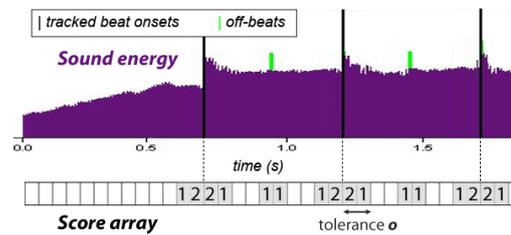


Figure 6. Creation of the score array from the detected beat and off-beat locations (simplified example)

4.2.2 Peak Detection

Next, the low-pass filtered acceleration signal is processed by a peak detection algorithm, for each axis. To detect peaks in a single accelerometer signal X (one axis) of fixed length, the following algorithm is used:

1. Calculate the average μ and the standard deviation σ of the sampled signal. (For computational reasons, an estimation for the average is used)

$$\mu = \frac{\max(X) + \min(X)}{2} \quad (4)$$

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=0}^N (X[i] - \mu)^2} \quad (5)$$

(Taking into account Bessel's correction for sampled standard deviation)

2. Create a binary form X_{bin} of the input signal with

$$X_{bin}[t] = \begin{cases} 1, & \text{if } X[t] > \mu + K \cdot \sigma \\ 0, & \text{else} \end{cases} \quad (6)$$

with K a parameter adjusted along the amount of noise present in the signal. This process is illustrated in figure 7.

3. For every continuous interval where $X_{bin} = 1$ a peak is detected at the location in the middle of the interval.

This process is illustrated in figure 7.

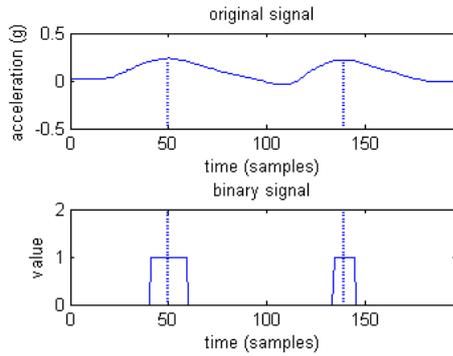


Figure 7. Original (filtered) acceleration signal (above) and its binary form (below), with detected beats represented by vertical, dotted lines

4.2.3 Scoring Algorithm

The final step is to assign a score to the detected peaks, depending on whether they correspond to a beat in the audio track. The complete synchronizing algorithm is described by the following steps:

1. Filter the input signal with the low-pass filter described above.
2. Buffer the signal into frames of $k \cdot I_b$ samples.
3. For each frame of $k \cdot I_b$ samples:
 - (a) Detect a maximum of k peaks $p_i, i = 1, \dots, k$ (only for the acceleration data in the current frame)
 - (b) Store the location l_{pmax} of the highest peak p_{max}
 - (c) The score is given by $s_{frame} = S[startindex_{frame} + l_{pmax}]$ with $startindex_{frame}$ the index of the first sample of the current frame, and S the score-array described above.

In this algorithm, k is a parameter specifying the length of the frame. In the proof-of-concept implementation, k is chosen as $k = 2$, meaning 2 beats can be expected each frame.

This algorithm is executed in parallel for each acceleration axis, detecting only k peaks, namely those leading to the highest scores.

5. GAME DYNAMICS

The game dynamics are an important factor to the gameplay and overall fun-factor of the game. For this project, a gameplay is implemented using *absorbing teams*, but thanks to the modularity of the synchronization algorithm, other gameplays are possible in future applications.

5.1 Absorbing Teams

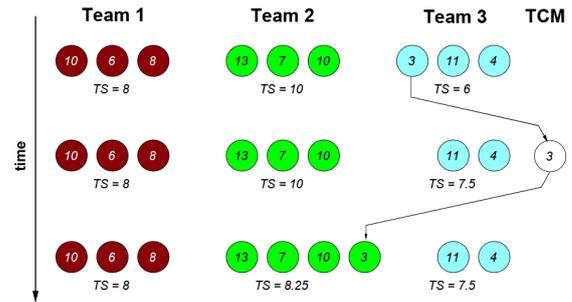


Figure 8. Example of Absorbing Team Game Dynamics: the player from team 3 is absorbed by team 2

The game mode we propose has all features of a social game as described in the beginning of this paper.

At the start of the game, all players are divided randomly into a user-specified number of teams, with a minimum of 1 player per team. It is clearly relayed to the players to which team they are assigned, using the color of the LED-display.

During the game, players are shown 2 scores: their *individual score* and their *team score*. The individual score represents how well this player is synchronizing to the music. The team score represents the mutual effort of the entire team a player is currently assigned to, and is calculated as the average of the scores of all team members.

After a predetermined time interval, the *lowest scoring player* of the *lowest scoring team* is removed from his/her team and transferred to *Team Change Mode (TCM)*. This will cause the team's average score to drastically improve (since they lost the low-scoring player).

After another, shorter time interval, the player in *TCM* is added to the *highest scoring team*, resulting in a drop in the team score of this team.

This way the highest scoring team figuratively *absorbs* the low scoring player from the lowest scoring team. This action then *equalizes* the team scores to a certain extent.

The game finishes when the music stops, or when all players have been absorbed by a single team.

An illustration of these game dynamics for 9 players is given in figure 8. In this example, the blue team (Team 3) is the lowest scoring team, and their lowest scoring player is removed from the team and set to TCM. This immediately affects the team score of team 3, changing it from 6 to 7.5. The changing player is then added to the highest scoring team, in this case the green team (Team 2), lowering the score of this team to 8.25.

5.2 Score Display

The scores are displayed using the LED-display developed at CMST. The outer rows represent the individual score, while the inner rows correspond to the team score, as shown in figure 9.

Scores are represented using a decimal system, with each colored led representing 10 points, and the position of the white led indicating the units. For example, in figure 9,

an individual score of 58 and a team score of 73 is shown. However, it is important to note that players shouldn't concern themselves with these scores, they are merely intended as an indication of how well they are doing. Players should only make the simple association: *more color = better score*.

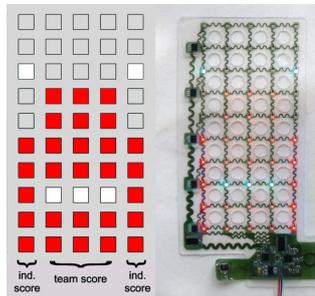


Figure 9. Scores: schematic (left) and real display (right). An individual score of 58 (outer 2 columns) and a team score of 73 (inner 3 columns) is shown.

6. PROOF-OF-CONCEPT IMPLEMENTATION

6.1 Technical Details

For the practical implementation of BeatLED, Java was chosen as a programming language, because of its portability, compatibility and ease of GUI design.

As mentioned before, the accelerometers embedded in Nintendo Wii remotes were chosen, which were easily interfaced via Bluetooth. However, because Wii remotes send their acceleration data in *bursts*, a resampling module had to be written, and was applied before inputting the data into the game.

Because the game was designed in parallel with the LED-display, the flexible visualization matrix was not immediately available. We opted to show preliminary output using projections on the ground, and later using rigid versions of the display.

6.2 Progress & Results

We succeeded in creating and testing a game with up to 10 players, and 5 teams. In fact, with the current game design, the number of players and team can be arbitrarily chosen at the start of the game, limiting the maximum number of players to the amount of Wii remotes that can be connected.

Because Bluetooth only supports up to 7 devices on 1 machine, we integrated the Open Sound Control¹ (OSC) protocol into the application. This way, additional accelerometers could be connected to a different machine, and transmit the acceleration data through a network connection to the main gaming machine.

This also allows for easy integration with other types of accelerometers. Future developers could easily create a separate module to send (correctly formatted) data coming from different accelerometers over the network to the

¹ <http://opensoundcontrol.org/>

existing game.

We also opted to send the scores in the OSC format, to allow for easy connectivity of additional score displays, and improve scalability and modifiability. Figure 10 shows a schematic view of the set-up, where one laptop is used for I/O, and one for processing. Note that there can also be one all-in-one machine, or several I/O computers and one processing computer.

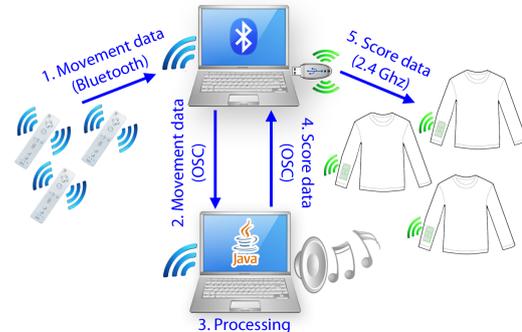


Figure 10. Schematic view of the data-chain using Open Sound Control (OSC). (During the game, the wii remotes are attached to/held by the players.)

Three different synchronizing algorithms were tried out: the original FFT algorithm, an algorithm based on adaptive oscillators, and a peak detection algorithm (described above). The peak detection proved to be the best performing and was left in the implementation. Its parameters were optimized, ensuring a realistic representation of the level of synchronization of music and movement.

All components are modular and can be interchanged with new implementations, leaving room for future adaptations of the game.

7. TESTING

Two types of tests were performed in order to evaluate the proof-of-concept implementation. First, the accuracy of the synchronizing algorithm had to be determined. When the final implementation was completed, a series of test sessions were organized, allowing real end users to play and evaluate the game.

7.1 Algorithm Tests

The accuracy of the peak detection was tested using the *peak detection error rate*, defined for a fixed length frame of samples i as

$$E_{peaks}(i) = \frac{|N_{detected}(i) - N_{annotated}(i)|}{N_{annotated}(i)} \quad (7)$$

with $N_{detected}(i)$ the detected number of peaks in the frame, and $N_{annotated}(i)$ the number of peaks visible in the data.

We obtain the *average peak detection error rate* by averaging these results for all frames of a set of test data.

$$E_{peaks-avg} = \frac{1}{F} \sum_{i=1}^F E_{peaks}(i) \quad (8)$$

A test set of accelerometer data was used, sampled at 200Hz with movements at varying tempo, with each visible peak carefully annotated. The results are visible in table 1. We can clearly discern that even at a frame size of 200 samples (corresponding with 1 second), usable results are obtained. This is clearly an improvement over the original FFT algorithm, which required a frame size of 4 seconds.

| frame size (samples) | 100 | 200 | 400 | 800 |
|----------------------|-----|------|------|------|
| $E_{peaks-avg}$ | - | 3.9% | 1.5% | 1.8% |

Table 1. Average error rate of detected number of peaks for data with varying tempo

7.2 Test Sessions

7.2.1 During development

During development, monthly test sessions were organized, using the latest version of the soft- and hardware. User input influenced the following design decisions:

Location of the motion sensors It became quickly apparent that users who held the motion sensor in their hand moved significantly less than users with the sensor attached to their body in an unobtrusive place. This was applied in all later test sessions.

Game and song duration Using trial and error, an average length of 1 minute was chosen for the audio fragments in the final implementation. User input showed that a sequence of 4 tracks proved ideal.

Cumulative score vs. Sliding window A choice can be made between a *cumulative score* and a *sliding window* score. Cumulative scores are simply added to a player's previous scores, throughout the entire game, while a sliding window calculates a player's score as the average of his/her last N scores, with N the window size. The choice of scoring method greatly affects the strategies applied by the players. Users were inconclusive about which they liked best.

7.2.2 With finalized software

The tests with the final proof-of-concept software were carried out using a less costly, rigid version of the LED display, allowing us to test the final game thoroughly, accurately and without risk of damaging the more expensive flexible demonstrators.

A short questionnaire (10 questions) was presented to the users of the final test sessions. Although it was filled in by a limited number of people, we were easily able to discern some trends. The answers clearly indicated that most aspects of the game are positively received. User opinions were very positive about the *game rules*, *feedback delay*, *game and audio duration* and *team change speed*. However, they were rather divided about the *score calculation* method (as mentioned above) and the *score visualization*. While half of the users thought the scores were clearly presented, the other 50% thought the exact opposite. This

shows that there might be a need to further examine the score representations.

8. FUTURE WORK

In the future, research in this topic could be continued by exploring other algorithms for synchronization, and by inventing new game modes. The results and opinions gathered from further test-sessions can be used to improve user experience.

Our software allows musicologists to research natural synchronization of people with music and each other, in a social context. More players could be added, and one could even imagine a mass-player game, where real results about social behavior would become apparent.

The LED visualization could be further expanded, firstly by adding more LEDs, and secondly by improving wireless communication, allowing us to show more complex images or even video. The user feedback indicated that a simpler visualization, which is easier to understand, might also improve user experience.

9. CONCLUSIONS

We developed a viable and usable social game, which meets all of the discussed requirements. We used readily available equipment, combined with state-of-the-art, newly developed technology.

While this is a proof-of-concept implementation, we showed that a social game such as ours could offer many possibilities, not only for research, but also in the entertainment sector and even health care.

Acknowledgments

We would like to thank all participators in this project, especially the supervisors at IPeM, MMLab and CMST. Also, special thanks are in order to the Belgian Industrial Research & Development (BiR&D) committee for their financial support, allowing this project to become a reality.

10. REFERENCES

- [1] N. Games, *Video Gamers In Europe*. Interactive Software Federation of Europe (ISFE), 2008.
- [2] S. B. Yang, S. and G. Graham, "Healthy video gaming: Oxymoron or possibility?" in *J. of Online Education - Vol. 04*, 2008.
- [3] M. Leman, M. Demey, M. Lesaffre, L. van Noorden, and D. Moelants, "Concepts, technology, and assessment of the social music game "sync-in-team"," in *Proceedings of the 2009 International Conference on Computational Science and Engineering - Vol. 04*, Washington, DC, USA, 2009, pp. 837-842.
- [4] M. Leman, *Embodied Music Cognition and Mediation Technology*. The MIT press, 2007.
- [5] S. Dixon, "Evaluation of the audio beat tracking system beatroot," in *Journal of New Music Research - Vol. 36*, 2007, pp. 39-50.

AUTHOR INDEX

- Arzt, Andreas 214
Avanzini, Federico 304
Bailey, Nicholas 91
Bank, Balázs 273
Beller, Gregory 253
Benetos, Emmanouil 19, 25
Berdahl, Edgar 83
Bigand, Emmanuel 128
Bresin, Roberto 70
Bullock, Jamie 227
Cadoz, Claude 83
Campana, Ellen 169
Camurri, Antonio 335
Canazza, Sergio 64, 109, 284, 304
Ceolin, Elena 48
Civolani, Marco 273
Cont, Arshia 190
Cooperstock, Jeremy R. 233
Dack, John 259
de Götzen, Amalia 388
De Nies, Tom 526
De Poli, Giovanni 109
del Bello, Valentina 273
Delbé, Charles 128
Delle Monache, Stefano 265
Demey, Michiel 526
Dimitrov, Smilen 290
Dixon, Simon 19, 25
Dolhansky, Brian 298
Doppler, Jakob 381
Doval, Boris 486
Dravins, Christina 70
Dykiert, Mateusz 450
Eigenfeldt, Arne 504, 510
Elblaus, Ludvig 141
Erichsen, Matthias 492
Falkenberg Hansen, Kjetil 70, 141
Fencott, Robin 259
Flexer, Arthur 247, 279
Florens, Jean-Loup 83
Fontana, Federico 273
Foote, Gordon 233
Foresti, Gian Luca 64
Freitas, Alan R. R. 346
Friberg, Anders 122
Frostel, Harald 214
Fukamizu, Akiyoshi 77
Fukayama, Satoru 362
Garrido, Sandra 323
Gasser, Martin 279
Genevois, Hugues 486
Gerhard, David 471
Ghomi, Émilien 486
Gillian, Nicholas 354
Giordano, Marcello 368
Gold, Nicolas 450
Goodman, Janel 169
Gopala Krishna, Koduri 33
Goto, Masataka 183
Goudard, Vincent 486
Grachten, Maarten 115
Gräf, Albert 375
Grill, Thomas 279
Guimarães, Frederico G. 346
Gulati, Sankalp 33
Halpern, Andrea 323
Hamano, Takayuki 77
Hashida, Mitsuyo 415

Hedblad, Anton 122
 Henriques, Tomás 464
 Hirata, Keiji 415
 Hirota, Keiko 77
 Holland, Simon 400
 Holzapfel, Andre 247
 Hoover, Amy K. 161
 Høvin, Mats 421
 Hughes, Craig 400
 Ikeda, Masahiro 439
 Ingalls, Todd 169
 Ishigaki, Asako 408
 Jensenius, Alexander Refsum 427
 Jylhä, Antti 220
 Kaniwa, Teruaki 439
 Katayose, Haruhiro 362, 415
 Kim, Sungyoung 439
 Kim, Youngmoo E. 177, 298, 310
 Kitahara, Tetsuro 362
 Kleimola, Jari 479
 Knapp, R. Benjamin 354
 Ko, Doyuen 233
 Kontogeorgakopoulos, Alexandros 492
 Kotsifa, Olivia 492
 Kreutz, Gunter 323
 Leman, Marc 526
 Maguire, Liam P. 444
 Makino, Shoji 77, 439
 Marsden, Alan 5
 Matsubara, Masaki 408
 Maupin, Steven 471
 Maxwell, James 510
 Mazzarino, Barbara 335
 McGinnity, Thomas M. 444
 McGlynn, Patrick J. 479
 McKinney, Curtis 457
 McPherson, Andrew 298, 310
 Mearns, Lesley 25
 Medeiros, Carolina Brum 518
 Michailidis, Tychonas 227
 Morishima, Shigeo 183
 Murofushi, Sora 183
 Nagata, Noriko 362
 Nakano, Tomoyasu 183
 Neukom, Martin 340
 Niedermayer, Bernhard 41
 Nordahl, Rolf 329
 Novati, Maddalena 304
 Nymoen, Kristian 421, 427
 O'Modhrain, Sile 354
 Okuno, Takatoshi 444
 Olmos, Adriana 233
 Paiva, Rui Pedro 431
 Panda, Renato 431
 Papetti, Stefano 273
 Park, Brett 471
 Pasquier, Philippe 504, 510
 Percival, Graham 91
 Pires, André S. 11
 Pisano, Silvia 335
 Polotti, Pietro 133
 Prockup, Matthew 177
 Queiroz, Marcelo 11, 198
 Raffaseder, Hannes 381
 Ramirez, Rafael 239
 Rao, Preeti 33
 Renaud, Alain 457
 Reuter, Christoph 41
 Ritsch, Winfried 394
 Robertson, Andrew N. 498
 Rocchesso, Davide 265
 Rodà, Antonio 64, 109, 284, 304
 Romano, Filippo 64
 Rubisch, Julian 381
 Rushka, Paul 233
 Rutz, Hanns Holger 155

Sagayama, Shigeki 362
Saito, Hiroaki 408
Salvati, Daniele 284
Scattolin, Francesco 64
Schmidt, Erik M. 177
Schubert, Emery 323
Schwarz, Diemo 105
Scott, Frederick 99
Scott, Jeffrey 177
Serafin, Stefania 290, 329
Shiv, Vighnesh Leonardo 317
Skogstad, Ståle A. 421
Smyth, Tamara 99
Sprenger-Ohana, Noémie 148
Stanley, Kenneth O. 161
Szerlip, Paul A. 161
Takahashi, Yuta 77
Terasawa, Hiroko 77, 439
Thomaz, Leandro Ferrari 198
Tiffon, Vincent 148
Tisato, Graziano 48
Trento, Stefano 388
Trochidis, Konstantinos 128
Truman, Sylvia 56
Turchet, Luca 329
Tzanetakis, George 91
Unander-Scharin, Carl 141
Valle, Andrea 206
Vamvakousis, Zacharias 239
Van de Walle, Rik 526
Vanfleteren, Jan 526
Varni, Giovanna 335
Vervust, Thomas 526
Volpe, Gualtiero 335
Wallis, Isaac 169
Wanderley, Marcelo M. 368, 518
Wermelinger, Michel 400
Widmer, Gerhard 41, 115, 214, 247
Woszczyk, Wieslaw 233
Yamada, Takeshi 77, 439
Zanolla, Serena 64
Zattra, Laura 48