SMC 2015

Maynooth University
National University of Ireland Maynooth

12th SOUND AND MUSIC COMPUTING CONFERENCE



MAYNOOTH, IRELAND

Proc. of the 12th Int. Conference on Sound and Music Computing (SMC-15) Maynooth, Ireland

Joseph Timoney and Thomas Lysaght, Maynooth University

July 30, 31 & August 1, 2015

Published by:
Music Technology Research Group
Department of Computer Science
Maynooth University
http://www.maynoothuniversity.ie/smc15/

ISBN: 9-7809-92746629

Credits:
Cover design: Thomas Lysaght
Logo photo: Front Cover: Lei Pan, Back Cover: Yinya Liu
LaTeX editor: Joseph Timoney, Fionn Collender
using LaTeX's 'confproc' package, version 0.7 (optional: by V. Verfaille)

Printed in Maynooth by Inkjet Printers — July 2015

SMC15 Conference Partners

Movidius

SFI Science Foundation Ireland

SMC15 Conference Patron Sponsors

Dave Smith Instruments

XILINX ALL PROGRAMMABLE

iZotope

MSB Mastering

Ardmore Sound

rocudo music machines

FAW

MAKER.IE

HOTONE Design Inspiration

I-CUBEX

imro Cherishing Music In A Changing World

Music Matters

Fáilte Ireland National Tourism Development Authority

meetinireland.com

Program Committee and Paper Reviewers

Torsten	Anders	Masataka	Goto	Christopher	Raphael
Jose	Antunes	Martin	Gould	Josh	Reiss
Anders	Askenfelt	Kerry	Hagan	Joshua	Reiss
Federico	Avanzini	Kjetil Falkenberg	Hansen	Lauri	Savioja
Roland	Badeau	Mitsuyo	Hashida	Diemo	Schwarz
Stefano	Baldan	Francesco	Grani	Eleanor	Selfridge Field
Zlatko	Baracskai	Matthieu	Hodgkinson	Stefania	Serafin
Alvaro	Barbosa	Risto	Holopainen	Xavier	Serra
Stephen	Barrass	Andrew	Horner	Alan	Smaill
Natasha	Barrett	Ozgur	Izmirli	Tamara	Smyth
Jose Ramon	Beltran	Dariusz	Jackowski	Johan	Sundberg
Stefan	Bilbao	Kristoffer	Jensen	Martin	Supper
Sebastian	Böck	Haruhiro	Katayose	Etienne	Thoret
Jordi	Bonada	Gavin	Kearney	Joseph	Timoney
Eric	Boyer	Damián	Keller	Petri	Toiviainen
Andrew	Brown	Jonathan	Kemp	Alberto	Torin
Ivica	Bukvic	Peiman	Khosravi	Giuseppe	Torre
Emilios	Cambouropoulos	David	Kim-Boyle	Cyril	Touze
Sergio	Canazza	Alexis	Kirke	Caroline	Traube
Baptiste	Caramiaux	Jari	Kleimola	Finn	Upham
Brian	Carty	Peter	Knees	Vesa	Valimaki
Ivan	Cohen	Alessandro	Koerich	Lindsay	Vickery
Fionnula	Conway	Victor	Lazzarini	Rudi	Villing
Leandro	Costalonga	Tom	Lysaght	Anja	Volk
Nicolas	d'Alessandro	Thor	Magnusson	Graham	Wakefield
Christophe	D'Alessandro	Sylvain	Marchand	Ge	Wang
Matthew	Davies	Alan	Marsden	Andreas	Weixler
Amalia	de Götzen	Ricard	Marxer	Jez	Wells
Giovanni	De Poli	Davide Andrea	Mauro	Tillman	Weyde
Philippe	Depalle	Patrick	McGlynn	Katieanna	Wolf
Simon	Dixon	Romain	Michon	Kinhong	Wong
Tony	Doyle	Chikashi	Miyama	Jim	Woodhouse
Tom	Erbe	Damian	Murphy	Matthew	Wright
Cumhur	Erkut	Kia	Ng	Lonce	Wyse
Georg	Essl	Vesa	Norilo	Jiajun	Yang
Gianpaolo	Evangelista	Linda	O Keeffe	Woon Seung	Yeo
Mikael	Fernstrom	Reid	Oda	Steven	Yi
Jonathan	Forsyth	Kjartan	Olafsson	Kazuyoshi	Yoshii
Guillaume	Gales	Sean	O'Leary	Jaeseong	You
Anastasia	Georgaki	Jussi	Pekonen	Massimiliano	Zanoni
Michele	Geronazzo	Darragh	Pigott	Ivan	Zavada
Volker	Gnann	Marcelo	Pimenta	Fengyun	Zhu
Emilia	Gomez Gutierrez	Marcelo	Queiroz	David	Zicarelli

Welcome from Joseph Timoney and Thomas Lysaght, Conference Co-Chairs

The organising committee of the 12th International Sound and Music Computing conference would like to welcome all delegates to Maynooth University for what we hope will be a stimulating event in the calendar of SMC activities. From the Summer School to the Paper and Poster sessions to the Concerts we wanted(want) to provide an interesting program deep in the spirit of SMC but also add a local flavour if we can(could).

Our concept for the Summer School was to have pairs of workshops in the fields of 'Sound and Music' and in 'Computing'. The first workshop is on Foley Sounds by Ardmore studios and it is highly interactive. The SMC conference has had papers in this area previously. To witness the blending of sound effects with film to create an impression of reality is fascinating. It demonstrates the importance of sound manipulation to the medium and also how new technology can work hand in hand with the old. The choice of a workshop on Mastering too gives insight into the music making production process. Lastly, the lecture on Sound Texture generation demonstrates how signal processing is creating new tools for sound designers. The other workshops were chosen to reflect this year's theme of 'High Performance Computing technologies and Audio'. We were fortunate to get the industry support from Microsoft Ireland, Movidius, and Xillinx. Technologies such as GPUs and FPGAs bring new possibilities to the processing and delivery of audio streams. However, they were not necessarily introduced with audio in mind so it is up to our community to investigate and imagine how they can work with the research ideas we want to implement. The final workshop is on DIY electronics by Maker.ie. This emerged out of our own close connections to the DIY community in Ireland over the last 4 years. We thought it would be a fun way to round off the summer school.

Our keynote speakers this year: Prof. Barry Truax will speak about microsound and composition. Dr. Derry Fitzgerald will talk about source separation algorithms. Dr. Stefan Bilbao will explain physical modelling synthesis. We had approximately 100 submissions for the technical program and have just over 70 papers in the final list. These compromise a diversity of areas from the Sound and Music Computing field: Interactive systems, Interfaces, Computational musicology, Multimodality, Spatial audio, and Signal processing. We are really grateful for all the hard work of the reviewers that delivered detailed appraisals to a tight deadline. It is their voluntary support that makes this a true community. We also would like to thank the chairs and those that are selecting the best papers for the prizes. It is important that we say thanks also to all our sponsors as their support is invaluable to the event. In particular, Science Foundation Ireland (SFI) and Movidius. Finally, I want to wish you an enjoyable conference and that your stay in Ireland will be pleasurable. I hope the weather will keep up for us. I'd like to thank everyone in the University that has helped us throughout the year and all the volunteers that are helping us through the hectic days of the conference itself.

Joe Timoney and Tom Lysaght

Conference Chairs of the Summer School and the Technical Program

CONFERENCE PROGRAM

Keynotes

Keynote Speakers

1 *Barry Truax*
Interacting with Inner and Outer Sonic Complexity: from Microsound to Soundscape Composition

3 *Derry Fitzgerald*
Musical Sound Source Separation

5 *Stefan Bilbao*
Perspectives on Physical Modelling Synthesis

Day 1

Oral Session 1 - Interactive Performance Systems I

7 *Iulius A.T. Popa, Jeffrey E. Boyd, David Eagle*
MUSE: a Music-making Sandbox Environment for Real-Time Collaborative Play

15 *Tom Mudd, Simon Holland, Paul Mulholland, Nick Dalton*
Investigating The Effects Of Introducing Nonlinear Dynamical Processes into Digital Musical Interfaces

Poster Craze I - Interfaces for Sound and Music/Multimodality in Sound and Music Computing I

23 *Masahiro Hamasaki, Masataka Goto, Tomoyasu Nakano*
Songrium: Browsing and Listening Environment for Music Content Creation Community

31 *Federico Avanzini, Sergio Canazza, Giovanni De Poli, Carlo Fantozzi, Niccolò Pretto, Antonio Rodà, Ivana Angelini, Cinzia Bettineschi, Giulia Deotto, Emanuela Faresin, Alessandra Menegazzi, Gianmario Molin, Giuseppe Salemi, Paola Zanovello*
Archaeology and virtual acoustics. A pan flute from ancient Egypt

37 *Constantin Popp, Rosalía Soria Luz*
Developing mixer-style controllers based on arduino / teensy micro controllers

43 *Misa Uehara, Takayuki Itoh*
Pop Music Visualization Based on Acoustic Features and Chord Progression Patterns Applying Dual Scatterplots

49 *Nicholas John Kirwan, Dan Overholt, Cumhur Erkut*
BEAN: A digital musical instrument for use in music therapy

55 *Saya Kanno, Takayuki Itoh, Hiroya Takamura*
Music Synthesis based on Impression and Emotion of Input Narratives

Oral Session II - Content Processing of Music Audio Signals

61 *Jordan Smith, Graham Percival, Jun Kato, Masataka Goto, Satoru Fukayama*
CrossSong Puzzle: Generating and Unscrambling Music Mashups with Real-time Interactivity

69 *Stefan Huber, Axel Robel*
Voice quality transformation using an extended source-filter speech model

77 *Anastasia Georgaki, Marcelo Queiroz*
"VIRTUAL TETTIX" : Cicadaas' sound analysis and modeling at Plato's Academy

Oral Session III - Interfaces for Sound and Music and Interactive Performance Systems IIa

85 *Otso Lähdeoja*
An Augmented Guitar with Active Acoustics

91 *Aristotelis Hadjakos, Axel Berndt, Simon Waloschek*
Synchronizing Spatially Distributed Musical Ensembles

99 *Ayaka Dobashi, Yukara Ikemiya, Katsutoshi Itoyama, Kazuyoshi Yoshii*
A Music Performance Assistance System based on Vocal, Harmonic, and Percussive Source Separation and Content Visualization for Music Audio Signals

105 *Tsubasa Fukuda, Yukara Ikemiya, Katsutoshi Itoyama, Kazuyoshi Yoshii*
A Score-Informed Piano Tutoring System with Mistake Detection and Score Simplification

Poster Craze II - Computational Musicology and Mathematical Musical Theory/Models for sound

111 *Cárthach Ó Nuanáin, Perfecto Herrera, Sergi Jorda*
Target-Based Rhythmic Pattern Generation and Variation with Genetic Algorithms

119 *Florian Thalmann*
Harmony of the Spheres: A Physics-Based Android Synthesizer and Controller with Gestural Objects and Physical Transformations

125 *Aidan Breen, Colm O'Riordan*
Capturing and Ranking Perspectives on the Consonance and Dissonance of Dyads

133 *Shun Shiramatsu, Tadachika Ozono, Toramatsu Shintani*
A Computational Model of Tonality Cognition Based on Prime Factor Representation of Frequency Ratios and Its Application

141 *Eyal Alon, Damian Murphy*
Analysis and Resynthesis of the Handpan Sound

147 *Nicholas Jillings, Brecht De Man, David Moffat, Josh Reiss*
Web Audio Evaluation Tool: A Browser-Based Listening Test Environment

Oral Session IV - Multimodality in Sound and Music Computing I

153 *Tatsunori Hirai, Yukara Ikemiya, Kazuyoshi Yoshii, Tomoyasu Nakano, Masataka Goto, Shigeo Morishima*
Automatic Singing Voice to Music Video Generation via Mashup of Singing Video Clips

161 *Federico Fontana, Federico Avanzini, Hanna Järveläinen, Stefano Papetti, Lorenzo Malavolta*
Rendering and Subjective Evaluation of Real vs. Synthetic Vibrotactile Cues on a Digital Piano Keyboard

169 *Marcello Giordano, Ian Hattwick, Ivan Franco, Deborah Egloff, Emma Frid, Valerie Lamontagne, Maurizio Martinucci, Christopher Salter, Marcelo Wanderley*
Design and Implementation of a Whole-Body Haptic Suit for "Ilinx", a Multisensory Art Installation

177 *Satoru Fukayama, Masataka Goto*
Music Content Driven Automated Choreography with Beat-wise Motion Connectivity Constraints

Day 2

Oral Session V - Models for Sound Analysis and Synthesis/Auditory displays and data sonification

185 *Jan C. Schacher, Hanna Järveläinen, Christian Strinning, Patrick Neff*
Movement Perception in Music Performance - A Mixed Methods Investigation

193 *Antonio Goulart, Joseph Timoney, Marcelo Queiroz, Victor Lazzarini*
Psychoacoustic impact assessment of smoothed AM/FM "303" resonance signals

201 *Rosalia Soria Luz*
Multichannel composition using state space models and sonification

Poster Craze III - Interactive Performance Systems/Multimodality in sound and musi computing II

209 *Antonio D. Carvalho Jr., Thomas Mayer*
Sensors2OSC

215 *Tiago F. Tavares, Gabriel Rimoldi, Vânia Eger Pontes, Jônatas Manzolli*
Cooperative musical creation using Kinect, WiiMote, Epoc and microphones: a case study with MinDSounDS

221 *Marcella Mandanici, Antonio Rodà, Sergio Canazza*
The "Harmonic Walk" and enactive knowledge: an assessment report

229 *Dominique Fober, Guillaume Gouilloux, Yann Orlarey, Stéphane Letz*
Distributing Music Scores to Mobile Platforms and to the Internet

235 *Miranda Kreković, Franco Grbac, Gordan Kreković*
Sound My Vision: Real-time video analysis on mobile platforms for controlling multimedia performances

241 *Florian Hörschläger, Richard Vogl, Sebastian Böck, Peter Knees*
Addressing Tempo Estimation Octave Errors by Incorporating Style Information Extracted from Wikipedia

Oral Session VI - Computer Environments for Sound/Music Processing

249 *Jari Kleimola, Oliver Larkin*
Web Audio Modules

257 *Jean Bresson, John MacCallum*
Tempo curving as a framework for interactive computer-aided composition

265 *Andrew J. Lambert, Tillman Weyde, Newton Armstrong*
Perceiving and Predicting Expressive Rhythm with Recurrent Neural Networks

Oral Session VII - Computational Musicology and Mathematical Music Theory I

273 *Roisin Loughran, James McDermott, Michael O'Neill*
Grammatical Evolution with Zipf's Law Based Fitness for Melodic Composition

281 *Georgi Dzhambazov and Xavier Serra*
Modeling of phoneme duaration for alignment between polyphonic audio and lyrics

287 *Luca Andrea Ludovico, Adriano Baratè*
Generalizing Messiaen's Modes of Limited Transposition to a n-tone Equal Temperament

295 *Chunyang Song, Marcus Pearce, Christopher Harte*
SynPy: a Python Toolkit for Syncopation Modelling

Poster Craze IV - Computer environments for sound/music processing/Content processing of music

301 *Abdullah Onur Demir, Hüseyin Hacıhabiboğlu*
MEPHISTO: A Source to Source Transpiler from Pure Data to Faust

309 *Alan Del Piccolo, Stefano Delle Monache, Davide Rocchesso, Stefano Papetti,*
To "Sketch a Scratch"

317 *Stephen Brown, Jorge Oliver*
Inter-track synchronisation for transmitting live audio streams over digital radio links

323 *Tatsunori Hirai, Shoto Sasaki, Shigeo Morishima*
MUSICMEAN: Fusion-based music generation

329 *Antonio Pošćić, Gordan Kreković, Ana Butković*
Desirable aspects of visual programming languages for different applications in music creation

337 *F. Canadas-Quesada, P. Vera-Candeas, N. Ruiz-Reyes, P. Alonso, J. Ranilla*
Online Harmonic/Percussive Separation Using Smoothness/Sparseness Constraints

Oral Session VIII - Multimodality in Sound and Music Computing

343 *Anis Haron, Matt Wright*
Wave Voxel Synthesis

351 *Maddalena Murari, Antonio Rodà, Osvaldo Da Pos, Emery Schubert, Sergio Canazza, Giovanni De Poli*
Mozart is still blue: a comparison of sensory and verbal scales to describe qualities in music

359 *Gareth William Young, Dave Murphy, Jeffrey Weeter*
Vibrotactile Discrimination of Pure and Complex Waveforms

Day 3

Oral Session IX - Sound and music signal processing algorithms/Music Information Retrieval

363 *Humberto Corona, Michael O'Mahony*
Mining Lyrics for Mood Classification in the Million Songs Dataset

371 *Jose J. Valero-Mas, Justin Salamon, Emilia Gómez*
Analyzing the influence of pitch quantization and note segmentation on singing voice alignment in the context of audio-based Query-by-Humming

379 *Giorgio Presti, Davide Andrea Mauro, Goffredo Haus*
TRAP: Transient Presence detection exploiting Continuous Brightness Estimation (CoBE)

Poster Craze V - Music Information Retrieval/Music Performance Analysis and Rendering/Perception

387 *Yading Song, Simon Dixon*
How Well Can a Music Emotion Recognition System Predict the Emotional Responses of Participants?

393 *Juan Li, Lu Dong, Jianhang Ding, Xinyu Yang*
Exploring the General Melodic Characteristics of XinTianYou Folk Songs

401 *Ryo Nomura, Takio Kurita*
Non-negative Sparse Decomposition of Music Signal using Pre-trained Dictionary of Feature Vectors of Possible Tones from Different Instruments

407 *Tobias Großhauser, G. Troester, A. Thul, M. Bertsch*
Sensor and Software Technologies for Lip Pressure Measurement in Trumpet Playing - from Lab to Classroom

413 *Jason Cullimore, David Gerhard*
The Virtuoso Composer and the Formidable Machine: A Path to Preserving Human Compositional Expression

419 *Malte Nogalski, Wolfgang Fohl*
Acoustically Guided Redirected Walking in a WFS System: Design of an Experiment to Identify Detection Thresholds

Oral Session X - Music Performance Analysis and Rendering

427 *Jérôme Nika, Dimitri Bouche, Jean Bresson, Marc Chemillier, Gérard Assayag*
Guided improvisation as dynamic calls to an offline model

435 *Tetsuro Kitahara, Kosuke Iijima, Misaki Okada, Yuji Yamashita, Ayaka Tsuruoka*
A loop sequencer that selects music loops based on the degree of excitement

439 *Muhammad Hafiz Wan Rosli, Andres Cabrera, Matthew Wright, Curtis Roads*
Perceptually guided Granular model of Multidimensional Spatial Sonification

Oral Session XI - Computational musicology and Mathematical Music Theory II/Sound and Music signal processing algorithms

447 *Hanlin Hu, Brett Park, David Gerhard index*
On the Musical Opportunities of Cylindrical Hexagonal Lattices: Mapping Flat Isomorphisms Onto Nanotube Structures

455 *Mario Martins, Carlos N. Silla Jr.*
Irish Traditional Ethnomusicology Analysis Using Decision Trees and High Level Symbolic Features

463 *Alex Wilson, Bruno Fazenda*
Navigating the mix-space: theoretical and practical level-balancing technique in multi-track music mixtures

471 *Diemo Schwarz, Sean O'Leary*
Smooth Granular Sound Texture Synthesis by Control of Timbral Similarity

Poster Craze VI - Auditory displays and data sonification/Content processing of music audio signals

477 *Stephen Roddy, Dermot Furlong*
Embodied Auditory Display Affordances

485 *Joseph Newbold, Nadia Bianchi-Berthouze, Nicolas Gold, Amanda Williams*
Musically informed sonification for self-directed chronic pain physical rehabilitation

491 *Francisco Jose Rodriguez-Serrano, Jonatan Menendez-Canal, Antonio Vidal, Francisco Jesus Canadas-Quesada, Raquel Cortina*
A dtw-based score following method for score-informed sound source separation

497 *Ben Robertson, Jonathan Middleton, Jens Hegg*
Multi-channel spatial sonification of chinook salmon migration patterns in the snake river watershed

503 *Katie Crowley, James McDermott*
Mapping brain signals to music via executable graphs

509 *Sérgio Freire, Pedro Cambraia*
Analysis of musical textures played on the guitar by means of real-time extraction of mid-level descriptors

Oral Session XII - Interactive Performance Systems IIb

515 *Takuya Kurihara, Naohiro Kinoshita, Ryunosuke Yamaguchi, Tetsuro Kitahara*
A Tambourine Support System to Improve the Atmosphere of Karaoke

521 *Marcello Giordano, Marcelo M.Wanderley*
Follow the Tactile Metronome: Vibrotactile Stimulation for Tempo Synchronization in Music Performance

527 *Jonas Fehr, Cumhur Erkut*
LICHTGESTALT: Interaction with sound through swarms of light rays

533 **List of Authors**

Barry Truax

Professor,
School of Communication,
Simon Frasier University,
Vancouver, Canada
Website: www.sfu.ca/~truax

Interacting with Inner and Outer Sonic Complexity: from Microsound to Soundscape Composition

Abstract:

It is possible to think of the two extremes of the world of sound as the inner domain of microsound (less than 50 ms) where frequency and time are interdependent, and the external world of sonic complexity, namely the soundscape. In terms of sonic design, the computer is increasingly providing tools for dealing with each of these domains, such as granular synthesis, convolution and the creation of virtual acoustic spaces through multi-channel soundscape composition utilizing computer-controlled spatial diffusion. The models of interaction involved with the complexity of each of these domains are instructive, and characterized by a blurring of the distinction between timbre and space. The presentation will include examples drawn from the composer's practice, such as the octophonic soundscape works Temple, Chalice Well, Aeolian Voices, and Earth And Steel.

Biographical Note:

Barry Truax is a Professor Emeritus in the School of Communication and formerly the School for the Contemporary Arts at Simon Fraser University where he taught courses in acoustic communication and electroacoustic music. He has worked with the World Soundscape Project, editing its Handbook for Acoustic Ecology, and has published a book Acoustic Communication dealing with all aspects of sound and technology. As a composer, Truax is best known for his work with the PODX computer music system which he has used for tape solo works and those which combine tape with live performers or computer graphics. In 1991 his work, Riverrun, was awarded the Magisterium at the International Competition of Electroacoustic Music in Bourges, France, a category open only to electroacoustic composers of 20 or more years experience. Truax's multi-channel soundscape compositions are frequently featured in concerts and festivals around the world.

Dr. Derry Fitzgerald

Nimbus Centre
Cork Institute of Technology,
Ireland
Website:http://nimbus.cot.ie/author/derry/

Musical Sound Source Separation

Abstract:

This talk focuses on presenting an overview of techniques for performing sound source separation with a particular focus on music recordings. Sound Source Separation attempts to extract individual sound sources or instruments from a recording containing multiple sources. In the case of recorded music, there are typically more sources than signals and so the music sound source separation problem is typically underdetermined. This has resulted in the development of a number of different model-based approaches such as matrix factorisation-based techniques and Bayesian methods. These are introduced using a real-world case study of using sound source separation techniques to create stereo upmixes from mono to stereo as an example. Following on from this, a number of recent developments in source separation algorithms will be presented, including Kernel Additive Modelling, and Spatial Projection-based methods. The talk will also highlight potential future directions for sound source separation research.

Biographical Note:

Dr Derry FitzGerald is a senior Post-Doctoral Researcher in Nimbus. He was a Stokes Lecturer in Sound Source Separation algorithms at the Audio Research Group in DIT from 2008-2013. Previous to this he worked as a post-doctoral researcher in the Dept. of Electronic Engineering at Cork Institute of Technology, having previously completed a Ph.D. and an M.A. at Dublin Institute of Technology. He has also worked as a Chemical Engineer in the pharmaceutical industry for some years. In the field of music and audio, he has worked as a sound engineer and has written scores for theatre. He has recently utilised his sound source separation technologies to create the first ever officially released stereo mixes of several songs for the Beach Boys, including 'Good Vibrations', 'Help me Rhonda' and 'I get around'. His research interests are in the areas of automatic music transcription, sound source separation, tensor factorizations, and music information retrieval systems.

Dr. Stefan Bilbao

Audio and Acoustics Group,
University of Edinburgh
Website: www.acoustics.ed.ac.uk/group-members/dr-stefan-bilbao/

Perspectives on Physical Modelling Synthesis

Abstract:

Physical modelling synthesis has now been around for quite a while---and in the mainstream for more than 20 years. And yet, only recently has it become possible to perform simulations for relatively complex musical instrument designs in a reasonable amount of time. There are various different approaches to physical modeling---some can be viewed as descending from standard abstract methods such as additive/table lookup methods for the synthesis of waveforms, but others have their roots in simulation techniques for the dynamics of vibrating systems. The first part of this talk is concerned with examining the different approaches to physical modelling in this light, in order to highlight both the differences and unifying features---particularly with regard to computational cost, which is the main downside to working with physical models relative to other synthesis techniques. The remainder of the talk is devoted to an exploration of the possibilities of physical modelling synthesis for some more elaborate instrument constructions, including: brass instruments, percussion, guitar models, modular instrument construction environments, and, finally, the computationally “big” problem of embedding physical models in a surrounding 3D space. Sound and video demonstrations will be presented.

Biographical Note:

Dr. Stefan Bilbao is currently a Reader in the Music subject area at the University of Edinburgh and the co-director of the Acoustics and Audio Group. His background is in Physics (BA, Harvard, 1992) and Electrical Engineering (MSc., 1996, PhD, 2001, Stanford University). He was previously a lecturer at the Sonic Arts Research Centre, at the Queen’s University Belfast (2002-2005), and a postdoctoral research fellow at the Stanford Space Telecommunications and Radioscience Laboratory (2001-2002). He is currently the PI of a NESS project, concerned with the development of large scale physical modelling synthesis algorithms on parallel hardware for musicians.

MUSE: a Music-making Sandbox Environment for Real-Time Collaborative Play

Iulius A. T. Popa
University of Calgary,
Alberta, Canada
juliuspopa@gmail.com

Jeffrey E. Boyd
University of Calgary,
Alberta, Canada
jboyd@ucalgary.ca

David Eagle
University of Calgary,
Alberta, Canada
eagle@ucalgary.ca

ABSTRACT

This paper reports the concept, design, and prototyping of *MUSE*, a real-time, turn based, collaborative music making application for users with little to no formal music education background. *MUSE* (a Music Sandbox Experience) is a proof-of-concept, web application running exclusively in the *Chrome* web browser for four players using gamepad controllers. First, we outline the proposed methodology with respect to related research and discuss our approach to designing *MUSE* through a partial gamification of music using a player-centric design framework. Second, we explain the implementation and prototyping of *MUSE*. Third, we highlight recent observations of participants using our proof-of-concept application during a short art/installation gallery exhibition. In conclusion, we reflect on our design methodology based on the informal user feedback we received and look at several approaches into improving *MUSE*.

1. INTRODUCTION

Computer-mediated, real-time collaborative music applications and interfaces could be categorized as either *serious musical instruments* or *playful musical toys*.

Serious musical instrument applications and interfaces provide their users the means to produce a quality, original music content in real-time. They usually feature an increased human-computer interaction complexity due to the large amount of musical controls they offer. This characteristic gives their users extensive creative freedom and the opportunity for skill mastery which in return leads to long-term engagement with the instrument. However, the quality of both the interaction and the musical result is directly related to the amount of time and effort invested by each user in mastering the required individual skills as well as acquiring collaborative experience as a musical ensemble. Moreover, the approach taken in designing most of these instruments relies on a substantial amount of either traditional or genre specific music notation, terminology, and/or concepts. Users lacking this knowledge will often create unsatisfactory, poor quality musical content resulting in a loss of interest in using that particular collaborative application or interface.

Musical toys on the other hand, don't require any musical education knowledge or group playing experience in order to interact with them successfully. These applications usually abstract most of the musical concepts they implement by presenting users with a small number of simple, intuitive controls. Apart from making the learning process easy, musical toys often produce a musical output of a consistent, pleasant quality, regardless of their users' individual or group proficiency. By making their users feel musically "competent" in virtually no time, applications and interfaces in this category appeal to large demographics. However, even when they do present some skill mastery opportunities (e.g. eye-hand coordination), musical toys often do not allow users to produce original content and express their musical creativity.

The main characteristics of these two categories place them at rather opposite ends of what appears to be a continuous spectrum. We felt that there was a potential middle-ground area worth exploring found between a *serious musical instrument* and a *playful musical toy*. The challenge was to merge two seemingly contrasting attributes: (1) the ease of use of a musical toy and (2) the interaction depth that allows for creative expression usually found in a musical instrument. The resulting application would therefore have to provide users with easy-to-grasp functionality and controls while allowing for producing original, pleasant musical results in a collaborative setting.

2. RELATED WORK

The initial goal of this research project was to provide users with little to no musical education background a platform for creating collaborative music in real-time. Primarily, we focused on addressing central issues found in a number of co-operative music-making applications and interfaces: (1) a lack of creative freedom found in many commercial games that we consider *musical toys*, such as *RockBand*™, *GuitarHero*™ [1] or *Rocksmith*™ [2], and (2) Real-time musical output of inconsistent quality identified in academic research projects approaching collaborative music-making from a *serious musical instrument* perspective [3, 4].

Mainstream co-operative music games favor eye-hand coordination skill over the musical creative freedom of

Copyright: © 2015 Iulius A. T. Popa et al. 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.

their users; players usually have to match visual cues they receive on screen with the hand / finger position on the input device they control [1]. The success of these types of music games for broad audiences is based on two main design choices: first, they use prerecorded hit-songs which ensure a recognizable, engaging, and ultimately satisfactory musical output every time they are played, and second, they are easy to learn and use. The long-term engagement of players with these games does not stem from a creative pool of musical curiosity but from the ongoing challenge/reward system for successful coordination of player input with visual cues in real time. From a music creativity perspective these games do not offer any real musical choice to the player. All interaction happens at the gameplay, non-musical level only. *Rocksmith*™ uses a real electric guitar as input device and tracks the player's input proficiency against a predetermined musical score. The overall music is comprised of a pre-recorded mix of a hit-song with the player's instrument track muted; in this case the player produces his own sounds in real-time and is rewarded as long as he correctly plays his part according to the score. The game thus expands more into the *serious musical instrument* territory by providing an opportunity for mastery leading to long-term engagement usually associated with traditional musical instruments. However, just like in *RockBand* or *GuitarHero*, the player's creativity is entirely restricted due to a lack of access to creating original music output [2]. Although several players can play as a group, since the design of these games disregards creative choice, the possibility for collaborative inspiration and creativity to occur is non-existent.

Unlike commercial games, a number of academic projects addressing real-time collaborative music such as *TOUCHtr4ck* [3] and *The Music Pattern* [4] use a *serious musical instrument* approach in designing their interfaces. They focus more on musical creativity and long-term engagement but somehow overlook the importance of setting limits to the users' level of creative input in their design choice [5]. We found that by giving users control to a large amount of musical parameters without taking a structured approach to the design of the system, these applications often produce real-time musical results of questionable quality, especially when used by people with no music education background.

reacTable™ [6], a tangible interface using a modular synthesizer approach designed for producing live electronic dance music, managed to significantly increase the overall quality of the ongoing musical output by narrowing down to a specific musical genre, *electronica*. *reacTable* allows one or more users to freely control pre-determined musical loops and sound parameters represented by physical objects placed onto a tangible tabletop display. The interface provides users with a clear visual feedback of the individual music objects' state and of the overall music system status at any time that leads to an engaging human-computer interaction. Since it features a considerable amount of musical and sound parameter controls, *reacTable* belongs mostly to the *serious musical instrument* category. Due to the high degree of creative freedom it offers, in the case of a multi-user setup, the actions of one player directly influence the other players' musical choices, thus greatly enhancing the collaborative experience. However, the ease of use of a *musical toy* is compromised due to the heavy reliance on electronic music concepts, such as modular synthesis music making techniques and its associated graphic symbols and metaphors. Because *reacTable* was designed primarily as a solo musical instrument for DJs, the quality of the overall music output is directly impacted by the users' knowledge and mastery of the afore-mentioned techniques and their group playing experience.

3. CONCEPT

Following the findings presented above, and based on our proposition that there is potential for merging collaborative *musical toys* with *serious instruments*, we set some preliminary design goals for our interface, in that it should: (1) produce an overall pleasant musical output, (2) allow users to express their musical creativity without relying on any music education or experience prerequisites, (3) provide users with an easy-to-use interface, and (4) offer users the opportunity for mastering musical skills leading to long-term engagement with the interface. Overall, we envisioned a co-located, real-time gameplay where the contribution of one player would change the ongoing musical output to a certain degree, which in return would act as a creative stimulus for the musical choices of the other players.

Based on our literature review we concluded that providing novice users with an increased level of musical controls while presenting them with a non-structured musical environment usually leads to inconsistent and sometimes unsatisfactory musical results. Since our first design requirement was to produce "an overall pleasant musical output" we set on designing a system capable of producing a consistent, pleasurable sound output with no sudden drops in the quality of the music. Although we wanted to offer increased creative freedom to our users compared to other collaborative *musical toys*, we believe that what leads to the players' long-term engagement with an interface is not the mere presence of countless affordances [7] but that of a system of rules in which those affordances coexist. This system of rules is what differentiates playing with toys (*paidia*) from playing games (*ludus*). Whereas *paidia* relates to spontaneity, excitement, and improvisation, *ludus* refers to rule-based, system-defined, organized play [8]. The long-term engagement found in rule based games resides in the vast amount of possibilities within the boundaries of the game. Similarly, by defining a clear set of rules, a *musical toy* can be elevated to the *playful platform for serious play* status, or simply put, a *game of music*.

4. DESIGN FRAMEWORK

From a design perspective, games can be either games of *progression* or *emergence* [9]. In a game of *progression* the designer has absolute control of the game's flow of events and challenges. This is achieved through carefully scripted level design. Games of progression are a good approach to designing narrative-driven computer games. However, replayability is usually very low due to the

scripted flow of the game combined with a limited amount of player options, and that of an already known outcome of the game on subsequent plays. Our interest was to approach *MUSE* as a game of *emergence*, as most rule-based games fall under this category and since we wanted *MUSE* to be played in real-time. In a game of emergence, the game state is continuously and directly influenced by the interactions between players with the game rules and components. Even a small set of relatively simple rules can lead to a vast number of possible states the game can be in at a given time. Games of emergence have a considerable replay value due to the high probability space they offer. Following our hypothesis that a *musical toy* governed by a small set of gameplay rules may result in a game of musical emergence, we approached *gamification* as a suitable solution to designing *MUSE*.

4.1 Gamification of Music

Gamification is the process through which game design elements are implemented in non-game contexts [10]. Gamification aims to increase user activity, quality of experience, and social interaction by adding a layer of gamefulness to a given design. We considered two approaches in gamifying *MUSE*: *complete gamification* – translating an existing tabletop or computer game system into the music domain, and *partial gamification* – implementing discrete game mechanics[1] by matching them with particular music concepts, controls, or parameters.

4.1.1 Complete Gamification - hypothesis

Since most board games and many computer games are games of emergence, we initially approached the gamification of *MUSE* by looking at translating an existing tabletop or computer game system into the music domain. After paper-prototyping some ideas, we realized that a successful integration of a complete game system with music – hence of a unified, coherent set of several game mechanics – may fail due to a number of issues we discuss here.

Within a given game system, the choice of actions does undeniably elicit an emotional response in the player, and their intensity fluctuates in direct relation with the impact a particular action has on the internal economy of the game world. For instance, in a game of *Chess*, the less decisive opening moves have less impact on the outcome of the game – thus producing less intense emotional response in players – compared to the game-changing, tension filled middle-game moves. In the same way, actions executed while performing music elicit a vast range of emotional responses in players based on many factors such as the setting of the performance, musical context, overall disposition of the participants, the specific role each player has in the ensemble, and many others.

All the possible interactions a player can have with an artifact – also called *motivational affordances* [12] – are the perceived opportunities that lead to a rewarding game experience, as perceived by the player in relationship with the game. It follows that the internal economy of a game system shapes the motivational affordances out of which the players' intrinsic motivation stems. We can then conclude that motivational affordances and the user satisfaction they convey cannot exist outside the internal economy of the system that generates them.

From a theoretical standpoint, a matching of the economics of the game system with the economics of real-time collaborative music is required in order to preserve the motivational affordances responsible for generating engaging, emergent gameplay. Even if this approach proves effective, the translation of the time domain from one system to another would pose a significant design challenge. The passing of time, as dictated by game rules, can render a game system either "fun" to play, where players enter a state of psychological "flow" [13], or turn it into a completely tedious activity. The timing of what is perceived as good "flow" in a particular tabletop game may negatively impact the creation and perception of real-time music. We therefore concluded that a successful translation of a complete game system into the music domain was not feasible due to the unique idiosyncrasies of these two mediums.

4.1.2 Partial Gamification of MUSE

We implemented several discrete game mechanics into the real-time music-making domain. A few of the game rules, interface, and game elements we tested revealed novel and interesting ways of interacting with music. However, the vast majority of the mechanics lost their gameplay effectiveness when translated into music. This finding confirmed Deterding's view on gamification [14] that game elements that are successful within a game system do not necessarily maintain their attributes when translated to a new context. He suggests that designers should conceptualize these perceived opportunities as *situated motivational affordances*, in that they are both *artifactual* – object specific– and *situational* – context or system specific. It became obvious to us that grouping a number of successful but discrete game mechanics together will not result in a coherent, logical real-time music system. Consequently, we focused on the overall quality of the system we wanted to produce instead of trying to build a unified system out of disconnected mechanics.

The approach we took into designing *MUSE* is based on the *MDA* model as described by Hunicke [15]. *MDA – mechanics-dynamics-aesthetics* – is a game development model which takes a player-centric, top-down approach to the design process. Instead of starting with the game mechanics to be implemented in the design (feature-driven) the focus is instead shifted towards the player's experience (aesthetics-driven). The usual steps of action are: (1) identifying the aesthetics the designer hopes to achieve, (2) defining the dynamics that may lead to those game experiences, and (3) creating the mechanics that produce the envisioned dynamics. In other words, we established the desirable emotional responses we wanted players to experience upon interacting with the game, we looked at what the game system's behavior should be at run-time based on the player's input, and lastly, we identified the game

[1] Game mechanics are comprised of game rules, interface, and game components [11].

components, interface, and rules necessary for the specified dynamics to emerge. *MUSE's* overall aesthetic goal was to provide inexperienced users with a pleasurable, engaging musical experience in a collaborative setting in real-time. The emotional response we were most interested in can be also referred to as plain "fun", but the term lacks specificity. Hunicke [15] proposes a relatively broad taxonomy containing eight categories of emotions (Table 3) usually experienced by users while playing games.

We decided that *MUSE*'s proof-of-concept main aesthetics should incorporate *Sensation*, *Expression*, and *Submission* (in this particular order of importance). If the proof-of-concept proved feasible, secondary aesthetics could include *Discovery*, *Fellowship*, and *Challenge*. The game dynamics generating these emotions thus had to first: produce real-time pleasant musical output (game as sense-pleasure), second: provide users access to musical self-expression (game as self-discovery), and third: allow for a curiosity-driven, unrushed exploration of musical possibilities in real-time (game as pastime).

Sensation Game as sense-pleasure	**Fantasy** Game as make-believe
Narrative Game as unfolding story	**Challenge** Game as obstacle course
Fellowship Game as social framework	**Discovery** Game as uncharted territory
Expression Game as self-discovery	**Submission** Game as pastime

Table 3. Hunicke's eight categories of emotions experienced during play, part of the MDA design framework.

The last step in outlining *MUSE*'s design structure was to identify the mechanics (components, interface, and game rules) that allow for the desired dynamics to take place. Several preliminary mechanics were considered as a direct result of the dynamics presented above: (1) limiting players' access to low-level musical controls such as individual pitches, velocities, etc. in order to ensure a pleasant overall sound output and minimize downtime, (2) abstracting musical concepts into easy-to-grasp interface elements with intuitive controls, and (3) creating an open-ended, sandbox music environment with no incentives other than musical ones (see Table 4).

Desired Dynamic	***Design Approach***
Produce real-time pleasant musical output	Limit the users' access to low-level musical controls
Provide access to musical self-expression	Abstract musical concepts into easy-to-grasp interface
Allow for ongoing exploration of musical possibilities in real-time	Create an open-ended, sandbox environment based on musical incentives only

Table 4. The design approaches leading to the desired game dynamics in *MUSE* (based on the MDA approach)

In designing *MUSE* we aimed at producing a music-output continuum ranging from *non-disturbing – interesting* to *pleasant – exciting* musical qualities. Although prone to subjective interpretation and in significant need of user testing, we assessed the quality of the music output in regards to our target audience. In terms of musical genre, we decided to gravitate around a combination of minimalistic [16] and popular music within a western music tradition context. A similar project, *Polymetros* [17], had used a minimalistic music approach and received a good audience response with regards to the musical genre. Within these musical boundaries, we approached designing *MUSE* focusing on maximizing the quality of the musical output of the system towards the *pleasant - exciting* end of the spectrum for most of the playing time.

5. IMPLEMENTATION

Based on the design choices outlined in Table 4 we implemented the game's interface, components, and mechanics (game rules) as presented below. After creating a number of musical game components and controls and testing them on several different game-board layouts, we realized that the amount of low-level user control still had to be significantly reduced for the following reasons: 1. the more fine control we had as designers over the "*no-wrong-notes*" system, the better the musical quality of the overall sound output, resulting in an enjoyable gameplay, and 2. in a real-time musical environment, small, low-impact musical actions/changes do not carry enough emotional response in the user to be worthwhile for novice players to perform. These fine alterations can only produce interesting musical results when handled by proficient musicians only; even then, since these changes are musically low-impact, they need to happen quickly and quite often to be musically effective.

We chose to design a very simple graphical interface, with no visual clutter, to allow users to grasp the connection between the overall music output and the visual representation of the music system at any time. The input control uses all gamepad buttons except the two 2-axis analog joysticks. For intuitive interaction and consistent control, colored gamepad buttons – Y X, A, and B – control graphic elements of the same color in the game.

In terms of duration of play, being in the proof-of-concept stage, *MUSE* is now completely open-ended, with no time limitation imposed by the gameplay. There are no points given to the players or any other reward systems in place, since we wanted to observe users' interactions within the game's controlled environment and quantify the emotional responses elicited in players based only on their musical actions, interactions, preferences, and perceptions. These observations could potentially lead to establishing a *musical hierarchy of motivational affordances* within *MUSE*. This could be later used as foundation for implementing more complex game mechanics leading to higher levels of interaction based on the intensity hierarchy of these responses.

5.1 *MUSE*'s System Description

The sound output of *MUSE* is comprised of four distinct 8-note loops that play continuously throughout the game session. Each player is assigned a musical loop of eight repeated notes at the beginning of the game. Each loop uses a distinct musical timbre (instrument) for the entire duration of the game. The sound banks of the loops were sampled from a digital synthesizer and span eight octaves in range. They are asynchronously loaded [18] into the web browser's cache memory during web page loading. For *MUSE*'s proof-of-concept sound banks we sampled two plucked string instruments and two pitched percussion sounds. To lower the memory load on the browser and to increase the timbral color based on the sample's velocity, we recorded one stereo sample (16bit / 44100Hz) at three distinct velocity levels – soft, medium, and loud corresponding to fixed MIDI velocity values – for every octave. All intermediate notes are transposed at run time in *MUSE*. Each sound bank thus holds one sample times eight octaves times three velocity layers to get 24 samples.

5.2 Graphical User Interface

MUSE is comprised of an eight by eight grid in which rows correspond to octaves and columns to speeds. Each player's loop can occupy one tile of the grid at any given time. The position on the grid affects the range (octave) and the playing speed of the loop. Every row corresponds to a different octave with loops sounding higher if positioned high on the grid. The speed distribution ranges from very slow on the left of grid to very fast to the right, with loops doubling or halving their speed depending on their position on the grid (see Figure 1).

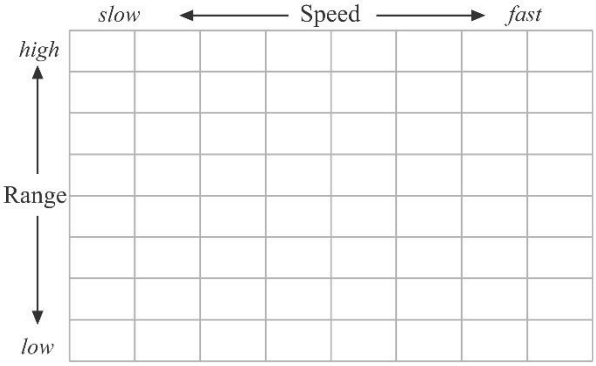


Figure 1. The eight speeds by eight ranges grid in *MUSE*

5.3 Game Components

A geometric shape represents a player's musical loop within the game. The player's name is displayed in the top-left corner of the tile. This representation of the loop is referred to as a *BLOCK* in the game (see Figure 2).

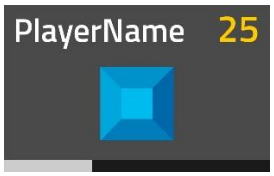


Figure 2. A player's *BLOCK* in *MUSE*.

The starting shape for all players is the *SQUARE*. This shape is the graphical representation of an eight note loop playing the same pitch at one fixed velocity. Other shapes available throughout the game are: *TRIANGLE* – same pitch, various velocities, *CIRCLE* – various pitches, same velocity, and *PENTAGON* – various pitches and velocities. All pitch/velocity contours are randomly generated during gameplay. Because the shapes' pitch/velocity complexity increases from *SQUARE* to *PENTAGON*, in order to further control the quality of the musical output we limited the amount of shapes available in the game as follows: maximum four *SQUARES*, three *TRIANGLES*, two *CIRCLES*, and one *PENTAGON*.



Figure 3. The graphic representation of all the available pitch/velocity contours in *MUSE*

5.4 Gameplay

Each player is automatically positioned on a tile close to the middle of the grid at the start of the game. Users begin playing by taking turns in activating (turning ON) their loops / *BLOCKS*. Since we wanted users to easily grasp the relationships between their actions and the musical results they produced, we implemented the turn-taking game mechanic in *MUSE*. Turn taking also allows a player to see what other players' musical contributions and preferences are, clearly hear the musical result produced, thus learning through observation [19]. Each player has a time counter displayed at the top-right corner of their *BLOCK* (see Figure 2) and receives 25 seconds of time each turn, time they can use on their turn to perform a game action of their choice. Once the timer reaches zero, the active player's turn ends immediately. If a player wishes to have more than 25 seconds available on their turn, they can *End Turn* before the timer runs out. Since they receive 25 seconds each turn, if players wish they can *End Turn* several times and accumulate up to a maximum of 60 seconds of available time.

On their turn, players can choose to either perform an action affecting only their own *BLOCK* or actions controlling all players' *BLOCKS*. The actions available for their own *BLOCK* are: (1) *Move Block*: allows players to move their BLOCK on the grid, thus changing their loop's register and speed in real time, (2) *Change Shape* which instantly changes the melodic / rhythmic contour of their *BLOCK* as per the geometrical shapes' musical characteristics described in section 5.3, and (3) *End Turn* which allows the current player to carry over to the next turn the

amount of unused time. The *End Turn* action passes the turn to the next player in the game.

The actions controlling all players' *BLOCKS* are: (1) *Move All Blocks*: similar to *Move Block* but moving all players on the grid, (2) *Refresh Blocks*: this action randomly generates new melodic / rhythmic contours for the current shapes in the game, (3) *Swap Blocks*: randomly swapping either the position of all the *BLOCKS* on the grid or the *SHAPES* among the players, (4) *Play Blocks*: allows users to perform a real-time chord progression for the entire musical output, and (5) *End Turn*: identical with the one presented above.

A special condition applies to the players' movement on the grid: if an active player moves their *BLOCK* onto a tile occupied by a passive player, the passive player is temporarily removed from the *GRID* and their loop sound is muted. An active player can thus remove all other players from the game for a significant musical change. Removed players take turns as usual having the following options: (1) *Get Back*: allows them to select any tile on the grid and re-enter the game (placing their BLOCK on the grid and activating the loop's sound – mute OFF) and (2) *End Turn*: as described above. The time needed for a removed player to get back on the grid is not accounted for. Upon re-entering the game the player has access to the usual (i.e. timed) game actions: *My Block* or *All Blocks*.

When moving on the grid, players can continuously move in any direction, since the *GRID* is designed so that loops/*BLOCKS* can "circle" through either octaves or speeds. In other words, a player's *BLOCK* located on the top row (highest sounding register) can still move up, resulting in the *BLOCK* being placed on the bottom row of the same column (lowest register). The same approach applies to speeds. Figure 4 shows a gameplay screenshot of *MUSE*.



Figure 4. Screenshot of *MUSE* in progress.

5.5 Game Platform

MUSE is a one page, proof-of-concept web application running in the *Chrome* web browser. We chose to develop for the web platform for portability. Although browser based, *MUSE* is designed for co-located play. Co-location of users in collaborative games significantly improves the overall perceived quality of gameplay [20-22], therefore *MUSE* was designed for up to four players sharing one screen display. Players use game pad controllers as input devices (we tested *MUSE* with both wireless gamepads - *Logitech F710* - and wired ones - *XBOX one*). The real-time, audio processing of *MUSE* is built using the Web Audio API (application programming interface), a high-level JavaScript API for processing and synthesizing audio in web applications [23]. The application is controlled through game pad controllers, currently supported in some web browsers through the Gamepad API [24].

6. PLAY TESTING

Fourteen participants tested *MUSE* during an open gallery exhibition at the University of Calgary. On several occasions two-player groups played the game, with each player controlling two musical *BLOCKS* while on two occasions a complete group of four-players joined a game session. The system setup consisted in one laptop running *MUSE* in the *Chrome* web browser with four *Logitech F710* wireless controllers connected to it. Players were seated on stools positioned around a small coffee table. The visuals were projected on the wall surface using a short-throw projector placed under the table. The audio output was played through two loud speakers positioned to the left and right of the projected image. A gameplay video recording of a one-player game session of MUSE is available online [25].

6.1 Observations and Reflections

6.1.1 Game as Sense-Pleasure

None of the users expressed aural discomfort during any of the most extreme *BLOCKS'* configurations on the grid. Due to the timed turn taking, players could quickly change the music output if they considered it less than satisfactory. Moreover, we observed some participants nodding their head or tapping their feet to the beat. By not giving users access to individual pitch or velocity controls, we found the proposed "*no-wrong-notes*" environment leading to an overall "*pleasant musical output*" to be effective. Many of the participants expressed their satisfaction with *MUSE* as being "*really fun*" or "*awesome*" while some even expressed their desire to come back and play it again.

6.1.2 Game as Self-Expression

Although players had the liberty to perform any action they wished on their turn, the ever-going, continuous sound output provided them with a musical context to which they either contributed further, abruptly changed or erased it almost completely. We noticed that soon after getting familiar with the interface, some players started making game choices of a more interesting musical quality within the given mood and characteristics of the music output at the time of their turn. We believe that our design approach allows creative self-expression to migrate from being an isolated, independent action to becoming a fundamental component of a unified, collaborative experience.

Players in two-player groups (controlling two loops each) became more interested in organizing the *BLOCKS*

into a particular configuration of shapes/positions in order to produce a pre-conceived musical result. This finding suggests that one-to-many actions – similar to the *Swap Blocks* action where a single press of a button is needed to arrange several players in a different configuration – would strongly appeal to users with an interest in musical arranging.

Some players found that using the *Move Block* action repeatedly on their own *BLOCK* in combination with continuously changing its shape led to a very rewarding "solo" experience. They expressed disappointment however, when by accident they moved onto a passive player's tile, thus removing that player from the grid and silencing it, which resulted in losing the harmonic support that inspired them to perform the solo in the first place. It seemed obvious to us the need to implement a "solo" mode in which the removal mechanic of the passive player is disabled. We are also considering making use of the two 2-axis analog joysticks as expressive controls mapped to several musical or sound parameters in the future.

Although players reported having a good experience with *MUSE* and were observed feeling in control of their musical contributions for most of the time, we feel that the level of long-term engagement and replay value of *MUSE* is still not as high as we would like it to be and needs to be addressed in the future. As mentioned before, the main reason why musical instruments offer long-term engagement to their users is the opportunity for skill mastery [26]. With that understanding in mind, we are looking at ways in which we can provide users with ways of improving their musical skill in *MUSE* without compromising the "*ease-of-use*" that characterizes most *musical toys*. We elaborate more on this insight in section 6.1.6.

6.1.3 Game as Submission (pastime)

The implementation of the turn-taking mechanic together with that of limited turn action time confirmed our expectations regarding the impact this will have on the flow of the game: players were more attentive to their choice of action and looking forward to their next turn. On subsequent turns, some players went back to the action they were previously performing while others explored the game's options further. While waiting for their turn, passive players keenly observed and listened to the musical changes performed by the active player, this essentially building up their excitement in anticipation of their own turn [13].

6.1.4 Game as Discovery

Since the tutorial does not mention the gameplay mechanic of temporarily silencing and removing other players from the game by moving onto their tiles, players were pleasantly surprised to discover this feature by themselves. Subsequently, this game mechanic got used progressively more. Since there is no other gameplay purpose for removing a player other than a pure musical one, players made use of this mechanic rather creatively. This led us to consider implementing more "hidden" musical/gameplay features in *MUSE* in the future and make this "surprise" component known to players early in the game. We anticipate that adding events and controls of uncertain quality but with a definite chance of occurring at a later time in the game, will keep the players' engagement and excitement levels high for longer periods of play time.

6.1.5 Game as Fellowship

Players who knew each other before playing the game, tended to discuss the game's controls and features more than players who never met before. The interaction at communication level, also noted in other games' play testing sessions [27], makes for an overall relaxed atmosphere and adds to the enjoyment of the participants when playing the game [20]. Future features of *MUSE* taking advantage of this knowledge may include short real-time challenges where participants have to actually work together towards a common musical goal, thus addressing one of the secondary dynamics discussed earlier, that of *Fellowship*.

6.1.6 Game as Challenge

One of the approaches to long-term engagement we strongly consider implementing in the future, is the design of two distinct, main game modes: a basic or *learner* mode, and an advanced or *performer* mode. The *learner* mode could use a level-design approach (usually found in computer games) that gradually increases in complexity as the player progresses through the game levels. Users would have to master the set of musical concepts and controls of a given game level before being allowed access to higher levels. Moreover, their learning progress during the *learner* mode could be saved in player profiles within the game. When playing in *performer* mode, players would load their profiles and have instant access to any game skills / controls they managed to learn so far. Having players with higher skill levels playing together with less advanced users while in *performer* mode, may encourage the latter ones to master the basics faster in order to gain access to the more expressive musical features and controls showcased by advanced players.

7. CONCLUSION

The initial overall positive user feedback gave us confidence in our design approach to creating a collaborative, real-time music-making sandbox environment. Nevertheless, further formal testing such as usability studies and rigorous playtests are required to gather the data needed to clearly establish the design approaches most suitable for developing real-time collaborative music interfaces for broad audiences. Based on our informal evaluations we believe that our design approach considerably fulfilled our proposed proof-of-concept expectations.

Acknowledgments

This work was kindly supported by the Computational Media Design program at the University of Calgary. Iulius A. T. Popa is funded by a Queen Elizabeth II scholarship from the University of Calgary.

8. REFERENCES

[1] K. Miller, "Schizophonic Performance: Guitar Hero, Rock Band, and Virtual Virtuosity," *Journal of the Society for American Music,* vol. 3, pp. 395-429, 11, 2009.

[2] M. Kalinauskas, "Gamification in fostering creativity," *Social Technologies,* pp. 62-75, 2014.

[3] A. Xambó, R. Laney, and C. Dobbyn, "TOUCHtr4ck: Democratic collaborative music," in *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction,* 2011, pp. 309-312.

[4] C. Ning and S. Zhou, "The music pattern: A creative tabletop music creation platform," *Computers in Entertainment (CIE),* vol. 8, pp. 13, 2010.

[5] M. D. Thibeault, "The Power of Limits and the Pleasure of Games: An Easy and Fun Piano Duo Improvisation," *General Music Today,* vol. 25, pp. 50-53, 2012.

[6] S. Jordà, "The reactable: Tangible and tabletop music performance," in *CHI'10 Extended Abstracts on Human Factors in Computing Systems,* 2010, pp. 2989-2994.

[7] J. J. Gibson, "The theory of affordances," *Hilldale, USA,* 1977.

[8] R. Caillois, *Man, Play and Games.* Free Press, 1961.

[9] J. Juul, "The open and the closed: Games of emergence and games of progression." in *CGDC Conf.* 2002.

[10] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: Defining gamification," in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments,* Tampere, Finland, 2011, pp. 9-15.

[11] B. Brathwaite and I. Schreiber, *Challenges for Game Designers.* Rockland, MA, USA: Charles River Media Inc., 2008.

[12] P. Zhang, "Technical Opinion: Motivational Affordances: Reasons for ICT Design and Use," *Communications ACM,* vol. 51, pp. 145-147, 2008.

[13] S. Deterding, "Situated motivational affordances of game elements: A conceptual model," in *Gamification: Using Game Design Elements in Non-Gaming Contexts, a Workshop at CHI,* 2011.

[14] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience.* Harper & Row, 1990.

[15] R. Hunicke, M. LeBlanc, and R. Zubek, "MDA: A formal approach to game design and game research," in *Proceedings of the AAAI Workshop on Challenges in Game AI,* 2004.

[16] J. W. Bernard, "The minimalist aesthetic in the plastic arts and in music," *Perspectives of New Music,* pp. 86-132, 1993.

[17] B. Bengler and N. Bryan-Kinns, "Designing collaborative musical experiences for broad audiences," in *Proceedings of the Ninth ACM Conference on Creativity & Cognition,* 2013, pp.234-242.

[18] (April 22, 2015). *AJAX: Asynchronous JavaScript and XML.* Available: https://developer.mozilla.org/en-US/docs/AJAX/Getting_Started.

[19] R. Paradise and B. Rogoff, "Side by side: Learning by observing and pitching in," *Ethos,* vol. 37, pp. 102-138, 2009.

[20] E. Jakobs, A. H. Fischer, and A. S. Manstead, "Emotional experience as a function of social context: The role of the other," *J. Nonverbal Behav.,* vol. 21, pp. 103-130, 1997.

[21] Y. A. De Kort and W. A. Ijsselsteijn, "People, places, and play: player experience in a socio-spatial context," *Computers in Entertainment (CIE),* vol. 6, pp. 18, 2008.

[22] B. H. Thomas, "The Future of Entertainment: How Play and Engaging Experience Can Contribute to the Society," *Computer Entertainment,* vol. 8, pp. 22:1-22:3, 2010.

[23] (April 22, 2015). *Web Audio API.* Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API.

[24] (April 22, 2015). *Gamepad API.* Available: https://developer.mozilla.org/en-US/docs/Web/Guide/API/Gamepad.

[25] "MUSE gameplay (a Music Sandbox Experience for collaborative play in real time), " *YouTube.* n.d. [Online] Available: http://youtu.be/ySg728q_bwA [June 17, 2015].

[26] S. Holland, K. Wilkie, P. Mulholland and A. Seago, *Music and Human-Computer Interaction.* Springer, 2013.

[27] K. Isbister and N. Schaffer, *Game Usability: Advancing the Player Experience.* CRC Press, 2008.

INVESTIGATING THE EFFECTS OF INTRODUCING NONLINEAR DYNAMICAL PROCESSES INTO DIGITAL MUSICAL INTERFACES

Tom Mudd
Centre for Research in Computing
The Open University
tom.mudd@open.ac.uk

Simon Holland
Centre for Research in Computing
The Open University
simon.holland@open.ac.uk

Paul Mulholland
Centre for Research in Computing
The Open University
paul.mulholland@open.ac.uk

Nick Dalton
Centre for Research in Computing
The Open University
nick.dalton@open.ac.uk

ABSTRACT

This paper presents the results of a study that explores the effects of including nonlinear dynamical processes in the design of digital musical interfaces. Participants of varying musical backgrounds engaged with a range of representative systems, and their behaviours, responses and attitudes were recorded and analysed. The study suggests links between the inclusion of such processes and the affordance of exploration and serendipitous discovery. Relationships between musical instruments and nonlinear dynamics are discussed more broadly, in the context of both acoustic and electronic musical tools. Links between the properties of nonlinear dynamical systems and the priorities of experimental musicians are highlighted and related to the findings of the study.

1. INTRODUCTION

This paper explores the complicated relationships between artists, tools and creative output. Worth [1] highlights a distinction between two perspectives on engagement with musical tools. The first — referred to as *idealist* — focuses on the tool as a device for realising an artistic idea formed in the mind of a composer or musician. In this case the tool is ideally a transparent medium for realising this idea with as little mediation as possible. This is essentially a communication-oriented model where a message needs to pass from A to B, and distortion of the message is undesirable. This is contrasted with a more material approach in which the tool plays a significant role in forming ideas, and the creative process is seen as a back-and-forth engagement with the tool.

Worth examines this latter attitude in the work of electronic musicians associated with the Mego label, but similar attitudes can be found in other musical practices, notably free improvisation where instruments are variously referred to as "allies" [2], things with which to have "relationships" [3], things with their own "intentions" [4], and where the performer may be "played by" the instrument [5, p 57]. Keep [6] discusses similar attitudes in experimental music, where the exploration of inherent sonic properties plays a significant role. Gurevich and Treviño [7] discuss the tendency towards a communication-oriented model in the New Instruments for Musical Expression community, noting that the term *expression* seems to include a tacit assumption that the performer's role is to communicate something "extramusical", and that this assumption risks excluding alternative modes of engagement such as those found in experimental musical practices. Musicians concerned with a more material-oriented approach often seem to value instabilities and unpredictable elements in their engagement with a given tool [3, 6, 8, 9].

A central motivation for this research is considering tool design with the latter interaction model in mind: if tools are something to form a dialog with, to have a relationship with, and to collaborate with, how do different designs facilitate or impede this approach?

Copyright: ©2015 Tom Mudd 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.

2. NONLINEAR DYNAMICAL SYSTEMS

This paper links the material approach outlined above to the properties of nonlinear dynamical systems (NLDSs), and examines connections between the inclusion of such processes in musical tools and particular approaches to engaging with these tools. NLDSs are systems in which the state at any given time is at least partly determined by previous states via feedback of some kind, and in which the determination of successive states is not a linear combination of current inputs and previous states. From an interaction perspective this means that timing can be a crucial element; *when* something is done can be as important as what is done. Such systems can at different times be stable and unstable, cyclical and unpredictable, chaotic but deterministic, and exhibit a range of complex behaviours.

NLDSs have been explicitly employed by composers and musicians in a variety of ways. [10] links their properties to compositional approaches to pitch and rhythm. Many others, including [11], [12], [13], [14], and [15] have implemented systems as structuring elements, synthesis ele-

ments, mapping elements, or combinations of these. Such systems exist in more subtle ways in many other musical practices however. Feedback has been used by a broad range of musicians in different musical areas [16–18], whether with microphone and loudspeaker or with feedback loops inside electronic systems. Many acoustic instruments themselves incorporate nonlinear dynamical processes, such as in the feedback relationship between reeds and resonating air columns [19], and in bowed strings [20].

The exploration of instruments and musical tools that takes place in many areas of free improvisation and experimental music [6, 21, 22] appears to be reflected in the choice of tools in these domains, as there is a tendency towards engaging with the unstable, unpredictable aspects of instruments [3, 6, 9] and often an explicit acknowledgement of a more material-oriented approach [6, 8]. The term "experimental" is used here in a very specific context, referring to an approach in which the outcome of an action or method is genuinely not known or unpredictable, associated particularly with post-Cagean musical practices as discussed by [23].

3. MAPPING AND DYNAMICAL PROCESSES

The study presented in this paper examines the ways in which different participants react to systems that include nonlinear dynamical processes, and considers whether this can be related to the participants' own practice regarding music making and engagement with musical tools.

As such, the present study is related closely to studies into the effects of different parameter mappings for musical tools, such as the work done by [24] and [25]. The study conducted by Hunt and Kirk [24] into the effect of complex cross-coupled mappings on musical engagement is of particular relevance. The study found that although the isolation of individual parameters in a controller through one-to-one mappings allowed for accuracy in completing very simple sonic tasks, the complex mappings were better suited to producing more complicated gestures, and perhaps more importantly, were often seen as more fun and potentially more interesting to use over longer periods. Menzies [25] extended this work through an investigation of the inclusion of linear dynamical processes in controlling musical systems, arguing that we are used to engaging with dynamical processes in our everyday life — moving limbs, manipulating objects, playing sports, etc. — and that dynamics lend a richness to these interactions.

The extension into *nonlinear* dynamics is perhaps counter-intuitive from the communication-oriented perspective described in section 1; the nonlinear element provides scope for chaos and bifurcations, making direct, predictable control potentially difficult. However, it may open the door to the kinds of relationships discussed in relation to the material-oriented perspective. As an example, consider the response of a reed instrument where too much pressure is applied to the reed, producing a sharp high-pitched squeak. In terms of interaction design, this result is very unpredictable, and can be difficult for beginners to control and remove from their playing. In the domain of more experimental music however, this bifurcation point becomes a potentially interesting site for investigation and experimentation, and can provide a means to find new and unexpected situations, even after many years of studying an instrument (see for example John Butcher describing his relationship with the reed in his saxophone playing [9]).

The study presented in this paper questioned participants about control, surprise, and potential for exploration in relation to a range of systems designed to differentiate the impact of the nonlinear dynamical elements.

4. STUDY METHODOLOGY

The study itself involved 28 participants of differing musical backgrounds each using four different representative digital interfaces (described in detail in the following section), all of which were controlled via a simple MIDI controller consisting of two dials and a slider. The participants were recruited such that half of the group were musicians consistently engaged in experimental musical practices. Each participant was asked to spend a period of 4-8 minutes trying out a given interface, before making a short recording of 1-4 minutes. The order in which the interfaces were presented was randomised for each participant, and no information was given as to how they worked, what each input might do, or how they would differ from each other. Data from the controller was logged from both activities. Participants then answered a range of Likert-scale questions (detailed in 4.2) before repeating the process with the remaining interfaces. After completing this process with all four interfaces, they provided information on their musical background (level of experience, instrument(s) played, experience with electronic musical tools, experience of free-improvisation, and a short overview of their musical practice), and conducted a short, semi-structured interview. The results presented here focus primarily on the data from the Likert-scale questionnaire with some context provided by the interviews.

4.1 The interfaces

A musician's experience and engagement with a particular musical tool may be affected by a wide range of factors: the specific affordances of the tool, the range of sound worlds available (e.g. the possibility for tonal, timbral, and rhythmical control and differentiation). The many different decisions to be made regarding the nature of the input device, the mappings and sound engine will all combine and interact with the user's own background, experience and taste. The specific design of the four interfaces in this study attempts to address some of these considerations, differing along two key variables: whether or not the interface incorporated a nonlinear dynamical process as a core aspect (NLDS vs static), and whether the mappings from the inputs to the parameters of the system were continuous or discontinuous (summarised in table 1). The former is the central concern in this study, whilst the latter provides a useful control, to test to what extent differences in the participants' responses were determined exclusively by the inclusion of nonlinear dynamics. Audio excerpts from the four interfaces can be heard at `http://tommudd.`

Interface	Nonlinear Dynamical	Mapping	Audio Engine
1	Yes	Continuous	Resonated Duffing Oscillator
2	Yes	Discontinuous	Resonated Duffing Oscillator
3	No	Discontinuous	Resonated Oscillator
4	No	Continuous	Audio Sample Based

Table 1. The four interfaces used in this study

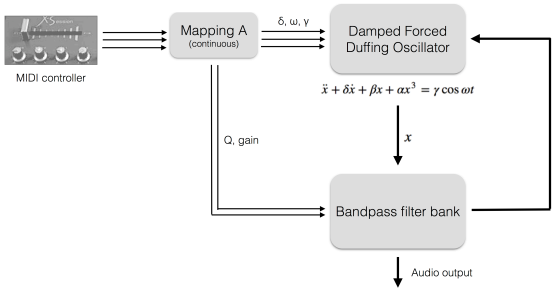


Figure 1. Interface 1. A damped forced Duffing oscillator coupled with a bank of linear resonators. The user interacts with the system via three MIDI controls.

co.uk/smc2015-examples/. A demonstration version of the MaxMSP software is also available at the same URL for reference. Each interface is discussed below in more detail.

4.1.1 Interface 1: Nonlinear dynamical system with continuous mappings

Both interfaces 1 and 2 are based on a damped forced Duffing oscillator [26], shown below in equation 1 as a discrete map. This is a nonlinear dynamical system that models the forced vibrations of a beam that is fixed at one end.

$$\begin{aligned} x_{n+1} &= y_n \\ y_{n+1} &= -\delta y_n - \beta x_n - \alpha x_n^3 - \gamma sin(\omega t) \end{aligned} \quad (1)$$

This equation is implemented at sample rate (44.1kHz in this instance) and coupled with a set of resonators such that the x_n term is passed through the filter bank, and the output of the filter bank is used in its place in the above equation. This combination of a nonlinear function coupled with a linear resonator bears a close resemblance to the structure of many acoustic instruments [19] and hence to many physical models [20]. The specific structure of interface 1 is shown in Figure 1.

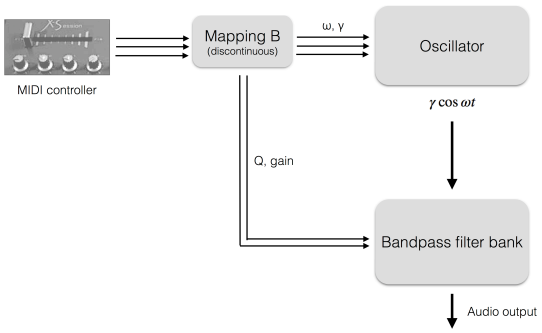


Figure 2. Interface 3. Duffing system and feedback are removed, leaving an oscillator and resonant filter bank. The discontinuous mapping is otherwise preserved from interface 2.

4.1.2 Interface 2: Nonlinear dynamical system with discontinuous mappings

Interface 2 differs from interface 1 only in terms of the mapping from the MIDI controls to the system parameters: interface 1 uses continuous mappings, whilst interface 2 uses discontinuous mappings that cause jumps in the parameters at particular points. This distinction was included to assess how significant the nonlinear dynamical component was in comparison with the static discontinuities in the mapping. In other respects this interface is the same as interface 1.

4.1.3 Interface 3: Static system with discontinuous mappings

Interface 3 is very similar to interface 2, but with the Duffing system removed as shown in Figure 2, rendering the interface non-dynamical and linear. The discontinuous mapping is retained however. Although the system is similar to interface 2 and to a lesser extent interface 1 in terms of the processes involved, the range of possible sounds is very different.

4.1.4 Interface 4: Static System with continuous mapping based on audio recording of interface 1

Interface 4 attempts to preserve the sound world of the Duffing systems by basing the interface around a two minute audio file recorded from interface 1. The system is therefore not a nonlinear dynamical system, but retains a very similar sound world to interfaces 1 and 2. The inputs are mapped to positions in the sample, playback rate and overall volume respectively.

4.2 Data Collection

The key data from the study presented in this paper comes from the questionnaire data and the MIDI control data, with some contextualisation provided by the interviews. The questionnaire asked each participant to what extent they agreed or disagreed with the following six questions for each interface (each on a five point Likert-scale):

1. "I felt in **control** of the sound"
2. "I found it straightforward to **recreate** particular sonic events"
3. "I was often **surprised** by the instrument's response"
4. "I feel that there are many areas that I could still **explore** and discover"
5. "I found a way of using the system that I felt fitted well with my own musical **practice**"
6. "I felt that my actions were **significant** in determining the final (recorded) result"

These questions will be referred to by the terms in bold text for the remainder of this paper. Participants were also asked to rank the four interfaces in terms of which they found the most satisfying to use.

5. RESULTS

The results presented in this paper form an initial evaluation of the data from this experiment, but there are some significant trends that emerge from this initial analysis. This section details some of the key findings both in terms of how the variation in the interfaces affected the participants' responses, and how participants of differing musical background reacted to variations in the interface.

5.1 The influence of nonlinear dynamics

Figure 3 presents the questionnaire data provided by the 28 participants. Two statistically significant trends emerge from this data:

- The responses to the first two questions regarding control and ease of recreating sonic events both correlated with the nature of the mapping, with the discontinuous mappings for interfaces 2 and 3 seeming to elicit less agreement with the two statements (as determined by an ANOVA with $F(1,27) = 9.45$, $p < 0.01$ and $F(1,27) = 7.18$, $p < 0.025$ for control and recreate respectively).
- The responses to the third and fourth questions regarding surprise and scope for exploration and discovery correlate with the inclusion of the nonlinear dynamical processes, with interfaces 1 and 2 being linked more closely with these statements ($F(1,27) = 13.11, p < 0.01$ and $F(1,27) = 11.81, p < 0.01$ for surprise and explore/discover respectively).

In certain respects these results are not surprising: it seems natural for a mapping that may abruptly change at a certain threshold to be deemed uncontrollable, and for a chaotic system to be linked with surprise and discovery. The more interesting aspect is that the nature of the mapping *does not* seem to impact upon the questions regarding surprise and exploration ($F(1,27) = 3.81$ and $F(1,27) = 0.06$ respectively, $p >> 0.05$) and — significantly for this paper — that the inclusion of nonlinear dynamical processes

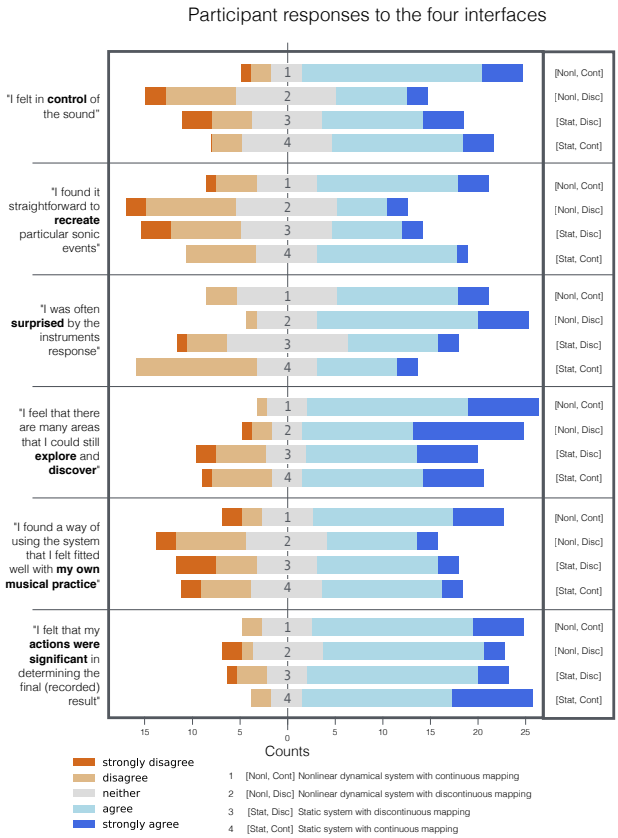


Figure 3. Participant agreement with six different statements in Section 4.2 as they apply to the four different musical interfaces described in Section 4.1.

does not seem to affect perceptions of control and repeatability ($F(1,27) = 0.06$ and $F(1,27) = 0.12$ respectively, $p >> 0.05$).

5.2 Interface preferences

The responses to the question "which interface did you find the most satisfying to use?" which asked participants to rank the four interfaces are shown in Table 2. The overall scores for each interface are calculated by awarding +2, +1, -1 and -2 for ranks of 1st, 2nd, 3rd and 4th respectively. This shows little clear consensus between participants, with only minor differences in rankings, with the scores all averaging out very close to zero. There was generally no correlation between the responses to the six statements detailed in Section 4.2 and interface preference. The only correlations found were for interface 1 (NLDS with continuous mappings), where participants who ranked this interface highly in terms of satisfaction also tended to feel in control, able to recreate sonic events, and that their actions were significant in determining the sounding result.

5.3 Differences between participants

The twenty eight participants can be grouped into many different categories based on the questionnaire and interview data, but as discussed in section 2, a concern for this research is whether there is a specific link between approaches to engagement and experimental musical prac-

,

Interface	Rated Most Satisfying	Rated Least Satisfying	Overall score
All participants			
1	10	7	2
2	5	7	-1
3	6	5	-1
4	7	9	0
Experimental group			
1	6	4	-1
2	2	2	1
3	2	3	-1
4	4	5	1
Non-experimental group			
1	4	3	3
2	3	5	-2
3	4	2	0
4	3	4	-1

Table 2. "Which interface did you find the most satisfying to use?" Columns 2 and 3 are counts. Overall score is calculated by awarding +2, +1, -1 and -2 for rankings of 1st, 2nd, 3rd and 4th respectively

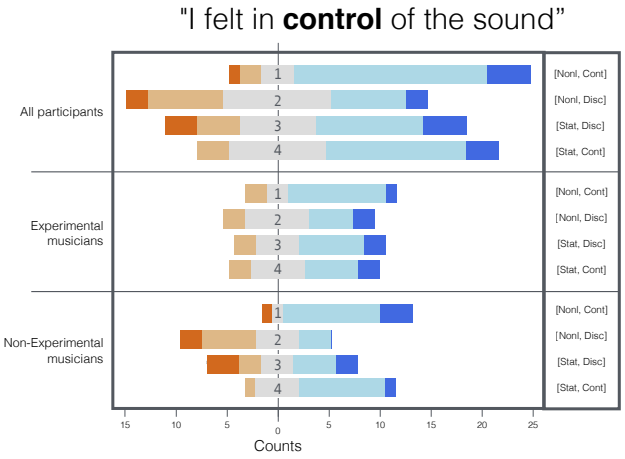


Figure 4. Comparison of response counts from musicians with and without experimental music backgrounds. The correlation between sense of control and the use of a continuous mapping (interfaces 1 and 4) is only significant for the non-experimental music group.

tices. Grouping the participants by whether or not they have a background in experimental music – in the narrow sense defined in Section 1 – highlights a number of differences in participant engagement. Figures 4 and 5 show how the responses to different questions varied according to whether a participant was considered to be in this group or not, with the two groups being comprised of 14 participants each.

A notable result is that there was less variation in the responses from the experimental music group for each interface. Neither of the two points presented above in section 5.1 are significant for this group alone, whilst they remain significant for the non-experimental group (see table 3).

Table 2 divides the preferences for each interface by the two groups. The interfaces are still difficult to distinguish on this basis however. Interface 1 appears to be more polarising for the experimental music group; despite six out of fourteen of the experimental music group finding interface 1 the most satisfying, four out of fourteen found it the least satisfying, and the overall score comes to only -1 indicating that overall there was no clear preference for the interface amongst this group.

6. DISCUSSION

6.1 Control vs Exploration

The link between nonlinear dynamics and both *surprise* and *scope for exploration* is a potentially interesting one for several reasons. Firstly, it is of potential interest to musical systems designers interested in creating interfaces that allow for surprise and exploration for either their own use or for others to use. A similar mechanism for achieving such a response might be through the use of stochastic systems, but there is a fundamental difference between chance processes and the chaotic-but-deterministic nature of nonlinear dynamics. [5, p 1] claims that "randomness does not produce a sense of surprise, but rather confusion, dismay, or disinterest". The fact that the systems are deterministic means that although they are unpredictable and allow for exploration, they still allow for actions to be repeated, and as [6] puts it "to re-access fruitful results."

The fact that the inclusion of nonlinear dynamical processes did not have a statistically significant effect on the participants' sense of control, while the inclusion of discontinuous mappings did have an effect, initially seems to be a surprising result. Both systems incorporate relatively abrupt transition points, where a small change in an input control leads to a drastic change in the resultant sound. In the case of the discontinuous mapping these transition points are absolute: when the input value crosses a certain point, the resultant sound will jump. The abrupt transitions due to the nonlinear dynamical processes however are more flexible: the transition point will vary according to the state of the other inputs, and may in fact vary depending on the history of the input, and therefore the timing of the controller movements (again, analogous the complex range of factors that lead to an abrupt squeak in a reed instrument). With certain settings, the abrupt transition may not occur at all. Several participants noted in their interviews that the discontinuous mappings limited the range of input values that were available if one wanted to avoid such transitions (a problem no doubt compounded by the already limited resolution of the MIDI controls).

The conditional nature of the response of the nonlinear dynamical elements could explain the link between these elements and the scope for exploration: the fact that each input control can affect the behaviour of the other controls, coupled with the fact that the history of the input may also play a part in determining the state of the system provides a broad landscape of possibilities to be explored.

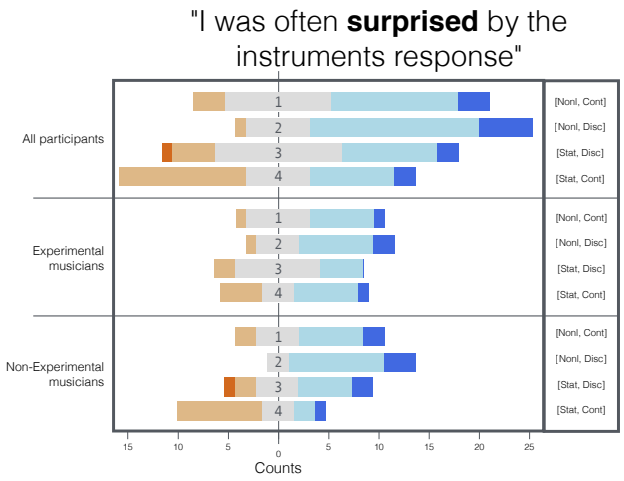


Figure 5. Comparison of response counts from musicians with and without experimental music backgrounds. The correlation between surprise and the inclusion of a nonlinear dynamics (interfaces 1 and 2) is only significant for the non-experimental group.

6.2 Other Implementations of NLDSs

These results are not necessarily easily generalisable. A great many other decisions are made in the process of creating musical interfaces, all of which may affect participant engagement, and the nonlinear dynamical elements themselves may be implemented in many different ways. A useful next step might be to consider these possibilities in more detail, and to examine the affect that each has on participant engagement. For example, whether the systems are responsible for synthesis directly, whether they are an aspect of the mapping process (as with [25]), or whether they cannot easily be classified in these terms. The systems may also be implemented at different rates: sample rate, control rate, or perhaps iterating only at user defined moments. Investigating how attitudes towards the interfaces shift when used for longer periods of time may also be productive, as the short 5-12 minute sessions for each interface may not be sufficient for participants to adequately answer the questionnaire and interview questions.

6.3 Contextual Complexity

The complexities of the musical (and social) situations in which musical tools are used make it very difficult to describe concrete cause-and-effect links between specific design decisions, and specific changes in engagement.

The interviews conducted with participants at the end of each session provide some useful contextualisation for the participants' questionnaire responses, particularly with regard to their qualitative attitude to aspects such as control and surprise. The musical situation in which a participant imagined themselves when using the interfaces seemed to have a strong influence on these aspects. For instance, in an imagined studio context, many participants expressed the desire to be surprised by the response of the tool, and that this might be a useful creative relationship. In a hypothetical concert situation however, participants often said that they would be less enthusiastic about surprises, or would distinguish between different kinds of surprises with some being more acceptable than others (some participants with a strong level of engagement with free improvisation provided notable exceptions however).

,

Variable	Question	F(1, 27)	p value
Experimental music group			
mapping	control	0.17	$n.s.$
mapping	recreate	1.44	$n.s.$
NL dynamics	surprise	3.47	$n.s.$
NL dynamics	explore	3.74	$n.s.$
Non-experimental music group			
mapping	control	15.83	< 0.01
mapping	recreate	6.12	< 0.05
NL dynamics	surprise	10.35	< 0.01
NL dynamics	explore	8.16	< 0.025

Table 3. Analysis of variance results examining how the impact of the mapping decisions and the inclusion of nonlinear dynamical processes on responses to questions on *control*, *recreate*, *surprise*, and *exploration* differed when considering the experimental music group and the non-experimental group separately.

6.4 Distinctions between participant groups

The links that were sought and not found between the grouping of participants into experimental and non-experimental and their preferences for the different systems may also hint at the complexity of the domain under consideration. There are perhaps many over-simplifications in the idea that experimental musicians will tend to find more exploratory interfaces more satisfying, and such links might be highly context dependent. The categories themselves involve large generalisations and do not take into account the range and complexity of individual musicians' attitudes and musical practices.

The experimental music group's lack of any statistically significant differentiation between the different interfaces noted in Section 5.3 does seem to suggest a significant difference in engagement and attitude however, although clear interpretations of this result are difficult. One possible explanation may be that the experimental group were more accepting of the specifics of each interface (in line with the material-oriented mindset outlined in Section 2), and were less inclined to try and realise pre-formed musical ideas. To give a more specific example, having a sense of control with a tool may relate to one's expectations: if unpredictable interactions are familiar, then one may feel in control despite the unpredictable nature of the interface. Similarly if one is comfortable with surprises from an instrument, then the interfaces may not seem so surprising.

7. CONCLUSIONS

As stated at the outset, the purpose of this research is to investigate the relationships between musicians, their tools, and their musical practice. The paper presented a study into the specific influence of nonlinear dynamical components on the ways in which musicians respond to, and engage with, a range of digital musical interfaces. Links were found between the inclusion of such elements and the perceived scope for exploration and discovery within the interface, as well as the potential for the results to surprise the musician. Links were also found between the continuous nature of the input mappings and the sense of control felt by the musicians, and their perception of their ability to repeat particular sonic gestures. These findings were discussed in the context of different musical approaches, particularly in terms of experimental musicians who often prioritise exploratory engagements with musical tools, although no clear links between such practices and the nonlinear dynamical elements were found in this study.

8. REFERENCES

[1] P. Worth, "Technology and ontology in electronic music: Mego 1994-present," Ph.D. dissertation, The University of York, Music Research Centre, 2011.

[2] D. Bailey, *Improvisation: Its Nature and Practice in Music*. NY: Da Capo Press, 1992.

[3] T. Unami, "What are you doing with your music?" in *Blocks of Consciousness and the Unbroken Continuum*, B. Marley and M. Wastell, Eds. Sound 323, 2005.

[4] P. Hopkins, "Amplified gesture documentary," 2012, samadhisound llc.

[5] D. Borgo, *Sync or Swarm: Improvising Music in a Complex Age*. Continuum International Publishing Group Inc, 2007.

[6] A. Keep, "Improvising with sounding objects in experimental musivc," in *The Ashgate Research Companion to Experimental Music*. Ashgate Publishing Limited, 2009, pp. 113 – 130.

[7] M. Gurevich and J. Treviño, "Expression and its discontents: Toward an ecology of musical creation," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2007, pp. 106–111.

[8] E. Prévost, "Free improvisation in music and capitalism: Resisting authority and the cults of scientism and celebrity," in *Noise and Capitalism*. Eritika, 2008.

[9] D. Warburton, "John Butcher interview," 2001, paris Transatlantic, March 2001, available at: www.paristransatlantic.com/magazine/interviews/butcher.html.

[10] J. Pressing, "Nonlinear maps as generators of musical design," *Computer Music Journal*, vol. 12, no. 2, pp. 35–46, 1988.

[11] I. Choi, "Sound synthesis and composition applying time scaling to observing chaotic systems," in *Proceedings of the Second International Conference on Auditory Display*, 1994, pp. 79–107.

[12] D. Dunn, "Autonomous and dynamical systems," 2007, new World Records.

[13] R. Ikeshiro, "Audiovisualisation using emergent generative systems," Ph.D. dissertation, Goldsmiths, University of London, 2013.

[14] D. Slater, "Chaotic sound synthesis," *Computer Music Journal*, vol. 22, no. 2, pp. 12–19, 1998.

[15] J. Bowers and S. O. Hellström, "Simple interfaces to complex sound in improvised music," in *Proceedings of CHI' 2000 extended abstracts. The Hague, The Netherlands*. ACM Press, 2000, pp. 125–126.

[16] D. Sanfilippo and A. Valle, "Towards a typology of feedback systems," in *Proceedings of the 2012 International Computer Music Conference*, 2012.

[17] T. Mudd, "Flexibility, subtlety, spontaneity in new instrument design: The feedback joypad," in *Proceedings of the 2012 International Computer Music Conference International Computer Music Conference*, 2012, pp. 614–617.

[18] S. Waters, "Performance ecosystems: Ecological approaches to musical interaction," in *Proceedings of Electroacoustic Music Studies Network Conference 2007*, 2007.

[19] M. E. McIntyre, R. T. Schumacher, and J. Woodhouse, "On the oscillations of musical instruments," *Journal of the Acoustical Society of America*, vol. 74, no. 5, pp. 1325–1345, 1983.

[20] J. O. Smith, *Physical Audio Signal Processing*. http://ccrma.stanford.edu/ jos/pasp/, 2010, online book, accessed 10 January 2014.

[21] E. Prévost, *The First Concert: An Adaptive Appraisal Of A Meta Music*. Copula, Matchless, 2011.

[22] C. Cardew, "Towards an ethic of improvisation," 1971, in Treatise Handbook. London: Edition Peters.

[23] C. Cox and D. Warner, *Audio Cultures: Readings in Modern Music*. Continuum International Publishing Group Ltd., 2004.

[24] A. Hunt and R. Kirk, "Mapping strategies for musical performance," in *Trends in Gestural Control of Music*, M. Wanderley and e. M. Battier, Eds. Ircam - Centre Pompidou, 2000, pp. 231–258.

[25] D. Menzies, "Composing instrument control dynamics," *Organised Sound*, vol. 7, no. 3, pp. 255–266, 2002.

[26] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer, 1983.

Songrium: Browsing and Listening Environment for Music Content Creation Community

Masahiro Hamasaki **Masataka Goto** **Tomoyasu Nakano**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{masahiro.hamasaki, m.goto, t.nakano}[at]aist.go.jp

ABSTRACT

This paper describes a music browsing assistance service, *Songrium* (http://songrium.jp) that enables visualization and exploration of massive user-generated music content with the aim of enhancing user experiences in enjoying music. Such massive user-generated content has yielded *"web-native music"*, which we defined as musical pieces that are published, shared, and remixed (have derivative works created) entirely on the web. Songrium has two interfaces for browsing and listening to web-native music from the viewpoints of scale and time: *Songrium3D* for gaining community-scale awareness and *Interactive History Player* for gaining community-history awareness. Both of them were developed to stimulate community activities for web-native music by visualizing massive music content spatially or chronologically and by providing interactive enriched experiences. Songrium has analyzed over 680,000 music video clips on the most popular Japanese video-sharing service, *Niconico*, which includes original songs of web-native music and their derivative works such as covers and dance arrangements. Analyses of more than 120,000 original songs reveal that over 560,000 derivative works have been generated and contributed to enriching massive user-generated music content.

1. INTRODUCTION

Since many amateur musicians started releasing their new original songs on video sharing services, a new type of music content that is born, listened to, and distributed on the web becomes popular. Many derivative works, such as cover versions and music video clips of those songs, have also been actively created and shared by other creators. Such music content is called *web-native music* [1] and has ever been increasing. This is different from commercially distributed songs that are originally released on the market and then copied to video sharing services.

Creators and listeners of web-native music form an interesting community that grows up and keeps on updating its own history dynamically on the web. This dynamics makes it difficult for people to grasp the whole picture of the community. Although ranking and recommendation are powerful and typical ways to find popular and similar music content in usual, they are not effective enough to grasp the whole picture. The goal of this research is to enable people to efficiently browse and listen to web-native music while grasping its nature and history.

In this paper we therefore propose to extend our web service called *Songrium* (http://songrium.jp) [1, 2] [1] by adding two interfaces, *Songrium3D* and *Interactive History Player*, that help people to be aware of the scale and history of the music content creation community through visualizing music content. Songrium3D visualizes the whole content in three dimensional space to gain

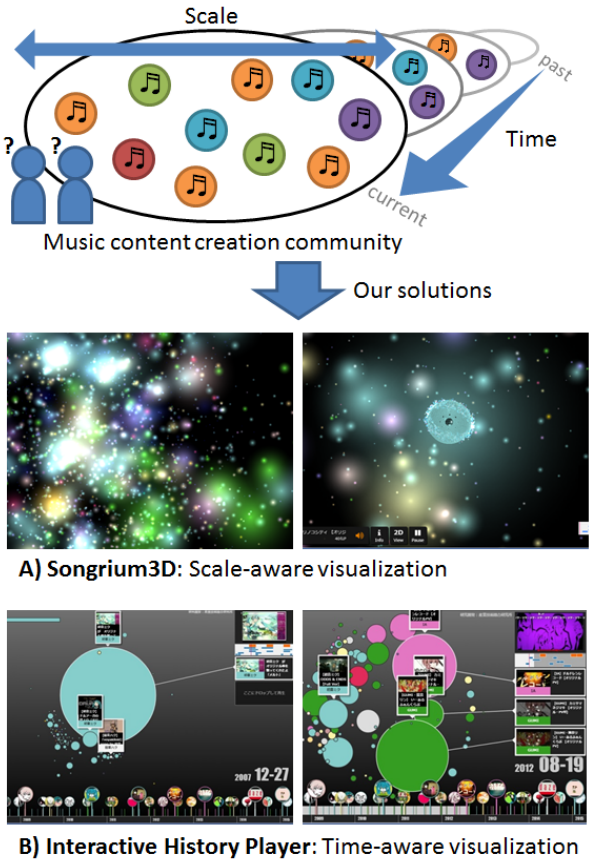


Figure 1. Songrium3D and Interactive History Player. The former is scale-aware visualization and the latter is time-aware visualization for music content creation community.

Copyright: ©2015 Masahiro Hamasaki 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] A demonstration video of Songrium is available at https://staff.aist.go.jp/masahiro.hamasaki/SMC2015/ .

community-scale awareness (Figure 1-A). Interactive History Player visualizes the whole content chronologically to gain community-history awareness (Figure 1-B). The current target content of Songrium is original songs using the singing synthesis technology *VOCALOID* [3] and their derivative works on Niconico, the most popular Japanese video-sharing service.

Songrium3D visualizes original songs as if they are stars in a planetarium.. Their positions are automatically arranged so that songs with similar moods can be closely located and be easily listened to by the user. Also, it seamlessly visualizes overviews of the whole content and details of each content. Furthermore, Songrium3D shows automatically synthesized visual effects for each user-generated music content during music playback. The effects are composed of predefined elements and their compositions are based on the analysis of music structure, both contributing to the high quality visuals.

Interactive History Player exhibits the growth in popularity of songs, arranged by published date, in an animated display. This feature enables users to experience a group of songs in one continuous movie, providing a clear, intuitive picture of how trends on video-sharing services have changed.

We launched Songrium in August 2012 and over 147,000 users have used our service. Since more than 120,000 original songs and 560,000 derivative works have already been registered and new songs are also automatically registered every day, Songrium is the only large-scale web service that can provide a comprehensive overview of the music content creation community for VOCALOID.

This paper is organized as follows. Section 2 explains music content creation community within the VOCALOID community on Niconico. Section 3 introduces basic functions provided by Songrium. Section 4 describes Songrium3D and Section 5 describes Interactive History Player. Section 6 presents our experiences with Songrium and Section 7 summarizes this paper's contributions.

2. WEB-NATIVE MUSIC ON NICONICO

Niconico is an extremely popular video communication service in Japan today. As with similar services (YouTube, etc.), users are able to upload and view videos. User-generated music content is the subject of many videos. Among them, music contents related to VOCALOID are popular, and are different from the rest as explained below.

VOCALOID is a singing synthesis technology [3] that forms a subset of the music content creation community on Niconico. This technology is used to synthesize the main vocal melody of songs. Many examples have been published as original works on the website. Despite the impressive technology used for the songs, the vocals produced by VOCALOID are readily identifiable as not of human origin, meaning that both creators and listeners naturally accept that these songs are first published on the web. Niconico therefore serves as a forum for VOCALOID creators and listeners to gather and share their relations.

Many different products are based on VOCALOID; each has a different vocal timbre. Most products have an associated character image, with Hatsune Miku [2] being the most well-known. Soon after releasing Hatsune Miku, Crypton Future Media (the developer of Hatsune Miku) officially started allowing users to reuse its character for derivative works with their original license: Piapro Character License [3]. Subsequently, users started to create music videos, such as promotion videos for musicians, with such original songs and drawings. Some users even went so far as to create 3D models of Hatsune Miku and create 3D animation videos [4, 5]. Thereafter, many songwriters published karaoke (full song without vocals) versions of their own original songs, prompting some users to sing these songs and to publish derivative works recorded in video clips.

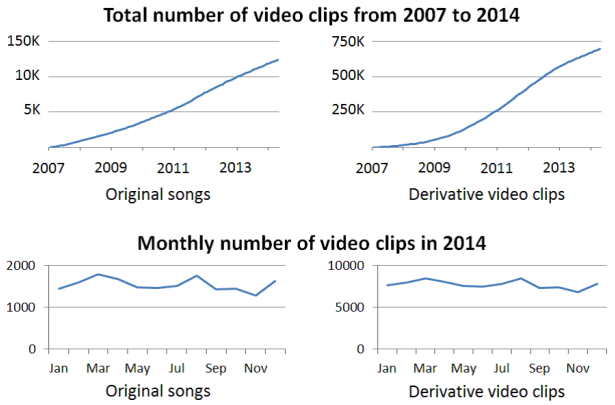


Figure 2. Two top graphs shows that the total number of published original songs and their derivative works in the period September 2007 – December 2014. The two bottom graphs shows the monthly number of published original songs and their derivative works in 2014.

We designated music having such characteristics as *web-native music* [1] and defined the conditions of web-native music as shown below.

(1) It is generally assumed that new original songs are first released on the web (without CD release or radio play, for example), with a unique URL identifying the source and release date.

(2) Creators do not hesitate to create and release derivative works of original songs on the web.

(3) After releasing original or derivative works, their creators can publicly receive feedback on the web and be encouraged to create more related materials.

Under the conditions mentioned above, web-native music naturally encourages to create derivative work. In fact, many derivative works are uploaded on Niconico. Figure 2 shows the number of total published original songs and their derivative works from September 2007 through December 2014 and the number of monthly published items in 2014. As described herein, we define the term 'derivative work' as a video clip that reuses a part of or whole of

[2] http://www.crypton.co.jp/mp/pages/prod/vocaloid/cv01_us.jsp

[3] From December 2012, they use Creative Commons license for foreign users.

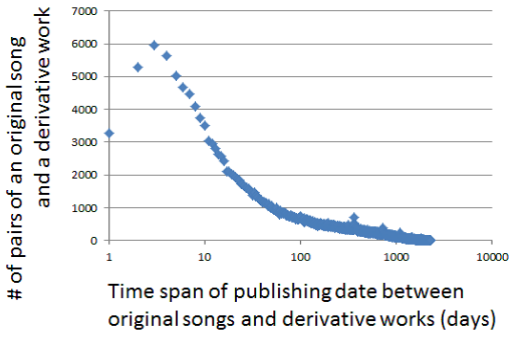


Figure 3. Distribution of time spans between publishing dates of original songs and their derivative works.

video clip of VOCALOID original song. According to this figure, the number of uploaded original songs and derivative works has been increasing rapidly. In 2014, about 1,500 original songs and 7,000 derivative works were published month by month.

Figure 3 depicts the distribution of time spans of publishing dates between an original song and its derivative works. Actually, 80% of derivative works are published after one month of the original publication date, whereas 40% are published after one year, showing that derivative works extend the longevity of original works. Furthermore, 14% of original songs have a derivative work with page views higher than original work, indicating that derivative works are also attractive contents.

3. SONGRIUM

3.1 Overview

Songrium is a music browsing assistance service that facilitates the understanding of the massive user-generated music content within the VOCALOID community on the Niconico service. Figure 4 presents an overview of Songrium. Songrium automatically gathers information related to original songs and their derivative works that grow day after day. It then classifies these contents and estimates the relations between original songs and derivative works. Songrium visualizes using results of music understanding.

By visualizing the web-native music, Songrium improves a user's understanding of various relations in the web-native music and an enriched interactive experience with the Web of Music. It was difficult for people listening to original songs to notice that there exist various derivative works of them, such as cover versions, singing or dancing video clips, and music video clips with 3D animations. By providing people with easy, intuitive access to those derivative works, Songrium enables them not only to find interesting music video clips but also to understand and respect the creators of music and video clips.

Songrium uses web mining and music understanding technologies together with advanced data visualization techniques to achieve unique function, such as a Music Star Map, Songrium3D (in Section 4), and Interactive History Player (in Section 5).

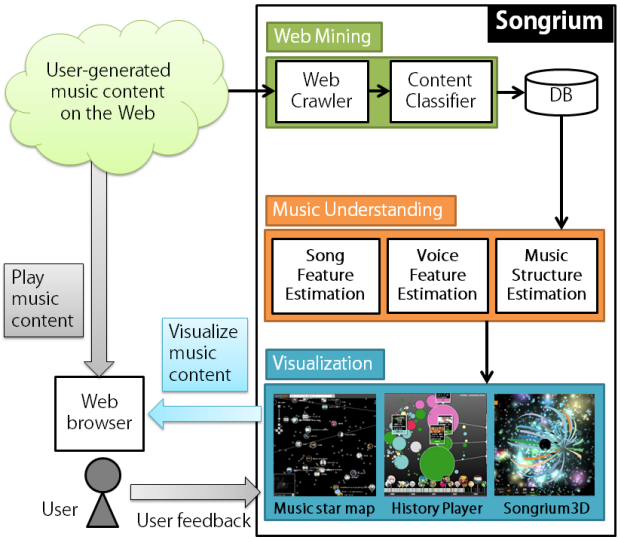


Figure 4. System overview of Songrium.

3.2 Web mining of Songrium

Every music video clip on Songrium is classified automatically as an original song or a derivative work. Niconico supports social tags for each clip and tags of some kinds such as "Original Song" and "be enshrined in the Hall of Fame song" are usually put on original songs on Niconico. Therefore, these tags are reliable. However, even if some original songs have no such tag, Songrium automatically classifies them correctly by crawling a set of related web sites to generate the "white list" of VOCALOID original songs. In the case of derivative works, these can be readily identified when the description text of the video clip includes a hyperlink to the original video clip from which it was derived. These hyperlinks almost always exist on Niconico because users like to acknowledge the original video clip.

When a derivative work is incorporated, its relation to the original song is estimated automatically. The derivative works are classified into the predefined categories. We defined six categories of derivative works: (a) Singing a song, (b) Dancing to a song, (c) Performing a song on musical instruments, (d) Featuring 3D characters in music video, (e) Creating a music video for a song, and (f) Others. The first three categories are derived from official categories used by Niconico; the other two categories are derived from our previous work [4, 5]. "Others" includes, for example, videos which review or rank existing videos, or which use VOCALOID songs as the background music to other video contents. With the exception of category *Others* all the remaining categories are extremely popular. All have their own unique social tags on Niconico. Using these tags, Songrium can produce a reliable classification of derivative works. Table 1 presents classification results of 564,623 derivative works of 128,044 VOCALOID original songs that we gathered from Niconico.

Table 1. Classification results of 564,623 derivative works. Some derivative works have multiple categories. Therefore, the total number of classifications is greater than the number of derivative works.

category	# of works
(a) Singing	379,342
(b) Dancing	30,159
(c) Arranging and Performing	35,584
(d) Featuring 3D characters	33,270
(e) Creating Music video	9,062
(f) Others	84,137

Moreover, Songrium enables users to report an error in any of the above classification of video clips, extraction of links, or estimation of relations easily to improve the user experience further.

3.3 Visualization of Songrium

Songrium has various functions of visualization for music content [2] [1]. *Music Star Map* is a function that visualizes original songs. Original songs are embedded in a two-dimensional space, mapped automatically based on audio feature similarity. The position of a song on the map is such that songs in proximity have similar moods (estimated by audio feature analysis). Figure 5-(A) portrays a screenshot of this function. Furthermore, when a user clicks an original song on Music Star Map, its derivative works appear as colorful icons and orbit the selected song. We designate this view as the "Planet View." Figure 5-(B) presents a Planet View screenshot.

In Figure 5-(B), each circle icon denotes a derivative work with attributes represented by the icon orbit, size, color, and velocity. The distance from the center is indicative of the publishing date, with the most recent work in the outermost orbit. The size of each icon reflects the number of page views; the color indicates one of the following derivative categories: Blue (Singing), Red (Dancing), Green (Arranging and Performing), Purple (3D characters in music video), Yellow (Creating music video), and White (Others). Finally, the velocity (orbit speed) of an icon represents how many times the content has been favorited by users of the system.

The official embedded video player of the Niconico service, shown at the upper-right corner, can play back a video clip of the selected original song (Fig. 5-(C)). Our music-listening interface has a chorus-search function for trial listening, *SmartMusicKIOSK* [6], which is shown below the embedded player (Fig. 5-(D)). Songrium has an original social tagging framework called the 'Arrow Tag' that allows to annotate a relation between music content [2]. Figure 5-(E) shows a list of Arrow Tags.

4. SONGRIUM3D

Songrium3D is a novel visualization interface based on the *Music Star Map* of Songrium . The Music Star Map visualizes original songs and their derivative works in two-dimensional space, but Songrium3D visualizes them in three-dimensional space. Using three-dimensional visualization, Songrium3D (1) visualizes whole contents, and (2) visualize songs, derivative works, and music structure seamlessly.



Figure 6. Screenshot of the "Songrium3D". (A) Users can search songs using keywords. Similarly, users can search playlists in Niconico using keywords or a URL. When users choose a Mylist, it starts auto play. (B) It shows a playlist. (C) This spherical object indicates an original song. Some objects and ribbons near the song are visual effects that synchronized to a song. (D) A song is encompassed with many colorful particles which mean its derivative works. (E) Other original songs can be seen way out there. (F) Embedded video player of Niconico for video playback and SmartMusicKIOSK for trial listening.

Figure 6 presents screenshot of the Songrium3D. The spherical object represents an original song in the center of the figure. When it plays a song, this object and peripheral objects move rhythmically that synchronized to a song. Many colorful circumjacent materials indicate derivative works of an original song. Color means a category of derivative works in the same manner as PlanetView (Fig. 5-(B)).

In the above, we describe that Songrium3D visualizes music structure of a song, derivative works of a song, and other songs at a time. The important point is that it is not only visualizing them at once but also visualizing them seamlessly. Figure 7 shows the transition from the top page to a user-specified song. First, all original VOCALOID songs are visualized in a three-dimensional space where songs which sound similar are positioned in proximity, similar to stars in a cosmos (Fig. 7-1). The colors of these stars correspond to VOCALOID characters, with brightness indicating popularity (number of plays). When a user chooses a song, a camera starts to move to the song (Fig. 7-2). Users can gradually see more of the song and its derivative works (Fig. 7-3,4). Songrium3D depicts derivative works as planets orbiting a star (original song). The planet color corresponds to the type of derivate work (singing, dancing, musical cover etc.) It displays all derivative works of the song using massive particles.

After arriving at the song, it displays visual effects that synchronized with sounds of the song (Fig. 7-3,5). Derivative works and other songs are shown in the background

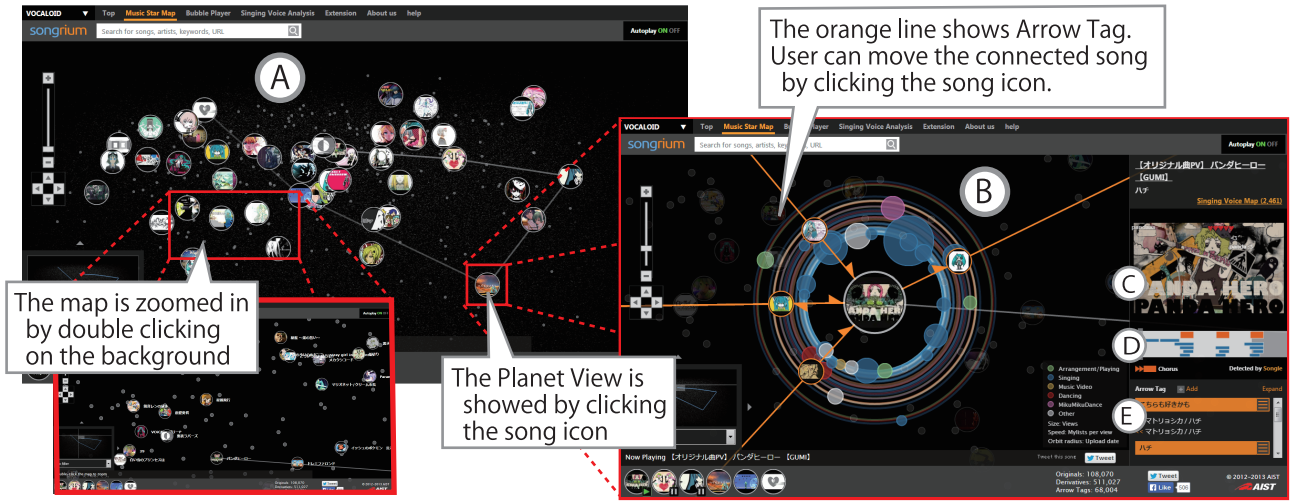


Figure 5. Screenshot of the (A) "Music Star Map" and (B) "Planet View" interface in Songrium. The former visualizes original songs; the latter visualizes their derivative works. Both are connected seamlessly. (A) All original songs are arranged in a two-dimensional space with similar songs positioned in proximity. Any particular area can be expanded for viewing by double clicking. It can then be scrolled to by dragging. (B) After selecting an original song on Music Star Map, users can view its derivative works rotating around the selected song in the center. (C) Embedded video player of Niconico for video playback. (D) Playback interface for trial listening (SmartMusicKIOSK). (E) Social annotated relations called Arrow tags [2] to and from this song instance.

when it shows visual effects that are a visualization of music structure. In this manner, Songrium3D visualizes all contents seamlessly. That is an important benefit of three-dimensional visualization. It helps users to have awareness of greatness of this music content creation community.

Songrium3D visualizes musical facets such as the beat and phrase structure, supported by signal processing and music understanding technologies. Figure 8 presents that how Songrium3D generates visual effects that synchronized to a song. Many music players have visual effects that are synchronized to audio signals. However, in the Songrium3D, visual effects synchronized to music structure, that is chorus section and repeated sections. One visual effect is mapped to one repeated section. Songrium3D has only six patterns of visual effects, however each songs has a different music structure. Then Songrium3D can generate various visual effects synchronized to a song. Handcrafted visual effects can be reflected by the deep meaning of a song, but it is high-cost. On the other hand, a signal visualization approach is low-cost, but it can visualize only shallow meaning of a song. Our approach is a combination of handcraft and automated generation. It is middle-cost and it can be reflected by the meaning of a song.

5. INTERACTIVE HISTORY PLAYER

Interactive History Player visualizes the history of VOCALOID songs. It plays groups of associated songs published within a user-specified time frame on continuous playback. The interface exhibits the growth in popularity of songs, arranged by published date, in an animated display. It plays songs whose play count is high in the period automatically. Consequently, this feature enables users to experience a group of songs in one continuous movie, providing a clear, intuitive picture of how trends on video-sharing services behave. If users become curious about some songs, users can play them with drag-and-drop operations.

Figure 9 portrays a screen shot of Interactive History Player. It displays groups of songs published during the specified period in chronological order, giving the user a full perspective on the trends and transitions in published song groups over time. Each song is represented by a "bubble" (a colored circle). New song bubbles appear in accordance with their respective published dates and congregate in an animation. The colors of the bubbles correspond to the voice synthesis library used in the VOCALOID software, whereas the sizes of the bubbles indicate play counts. On the left side of the screen, the bar chart presents a summation of play counts of bubbles in the same VOCALOID characters.

Users can choose a song for listening, change a period, and filter songs by VOCALOID solely by mouse operation. Furthermore, Songrium plays truncated chorus sections of certain songs that satisfy conditions specified by the user, making it easy to see what kinds of songs were being published as bubbles increasingly appear. These interactive functions help users to browse music contents.

The Interactive History Player has two different versions, "Singing derivative works" version and "Dancing derivative works" version. They display derivative works with the same interface. The bubble colors correspond to their original songs. A bar chart shows a trend of original songs for derivative works.

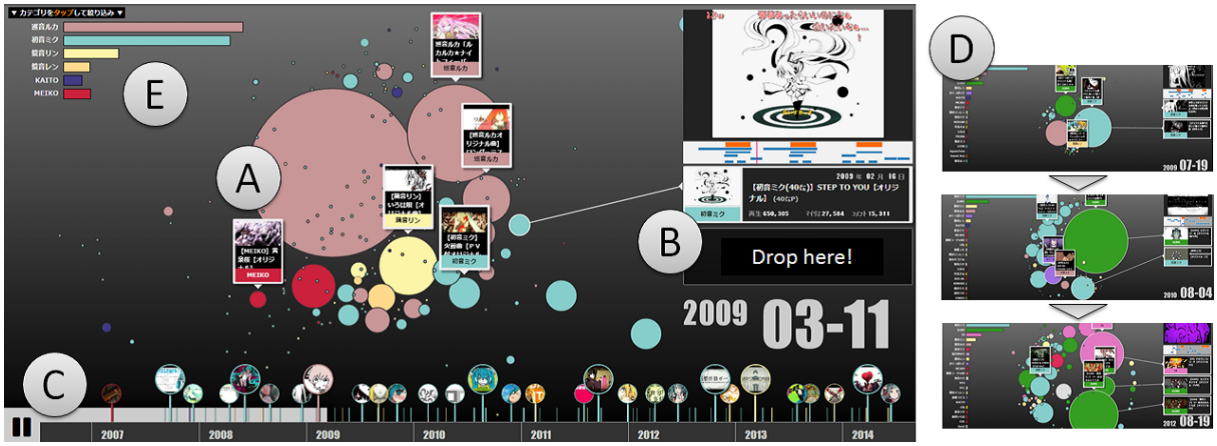


Figure 9. Screenshot of the "Interactive History Player" in Songrium. It visualizes the history of VOCALOID songs. (A) A bubble means music content. Its size indicates play counts and its color indicates VOCALOID. When users click a bubble, its thumbnail and metadata are shown. (B) Users drug and drop a bubble here, then the song is added to the playlist. (C) Timeline displays the current time and popular content in each period. When users click on the timeline, it jumps to the clicked period. (D) It displays groups of songs published during the specified period in chronological order, automatically. (E) Bar chart shows summation of play counts of bubbles.

6. EXPERIENCES WITH SONGRIUM

6.1 Songrium on the Web

Users can use all functions of Songrium and can watch the latest music contents everyday merely by the web browser. The web crawler of Songrium checks updated music video clips related to VOCALOID on Niconico automatically on a daily basis. The user interface of Songrium is implemented using HTML5, SVG, JavaScript, the JavaScript library D3.js, threeJS, and the embedded video player of the Niconico service.

Songrium service was released to the public at http://songrium.jp on August 7, 2012. In addition to the web service, the Songrium extension for Google's Chrome browser was released on February 28, 2013. As of April 2015, 128,044 original songs and 564,623 derivative works have been registered in Songrium. More than 147,000 users have visited our web site. More than 2,500 users have installed the browser extension.

6.2 Songrium3D on the live stage

Animation of Songrium3D was used as a back screen movie on the live stage of Hatsune Miku in the "SNOW MIKU 2015 LIVE!" held four times in February 7th-8th, 2015. It was hosted by Crypton Future Media Inc. for an audience of over 7,000.

Figure 10 shows the live performance with Songrium3D. The centerpiece of the figure of Hatsune Miku on the DILAD screen and the movie generated by Songrium3D shown over her head. At the bottom, one can see the many light sticks swung by audiences members. The live show used the prerecorded singing voice and prerecorded dancing animations of Hatsune Miku. Only the backup band performed live on the stage.

We produced a prerecorded animation of Songrium3D to avert problems deriving from internet connections or real-time rendering. We captured screenshots of Songrium3D in 29.97 fps and combined them to produce a single movie. The movie of the live performance [4] and the animation of Songrium3D for the live [5] are published on the Web.

6.3 Interactive History Player in public events

The Interactive History Player has been used at two big public events, *Niconico Chokaigi 3* [6], *Niconico Chokaigi 2015* [7] and *Magical Mirai 2014 in Osaka* [8]. The first twice were public events for Niconico hosted by Niwango, Inc. Chokaigi 3 held April 26-27, 2014. The total number of attendees was 124,966. Similarly, Chokaigi 2015 held April 25-26, 2015 and the total number of attendees was 151,115 that is up 20% over last year. The last one was a public event for Hatsune Miku hosted by Crypton Future Media, Inc. It was held in August 30, 2014 and the total number of attendees is about 11,000.

Figure 11 presents the appearance of the system in each event. Many people enjoyed using the Interactive History Player and watched videos nostalgically. At first many people were passively browsing contents using the system, and then they seek their own memorial contents or period using the category filtering and the time warp function. Some users talked about good old contents with friends while using our system.

7. RELATED WORK

Songrium, at its core, is a music browsing assistance service. Most previous research into interactive music browsing has emphasized visualization to explore musical collections. Given the multiple dimensions associated with

[4] https://youtu.be/GOano9x9cBY
[5] https://youtu.be/71o8Jit1c4I
[6] http://www.chokaigi.jp/2014/abroadEnglish.html
[7] http://www.chokaigi.jp/2015/abroadEnglish.html
[8] http://magicalmirai.com/2014/index_en.html

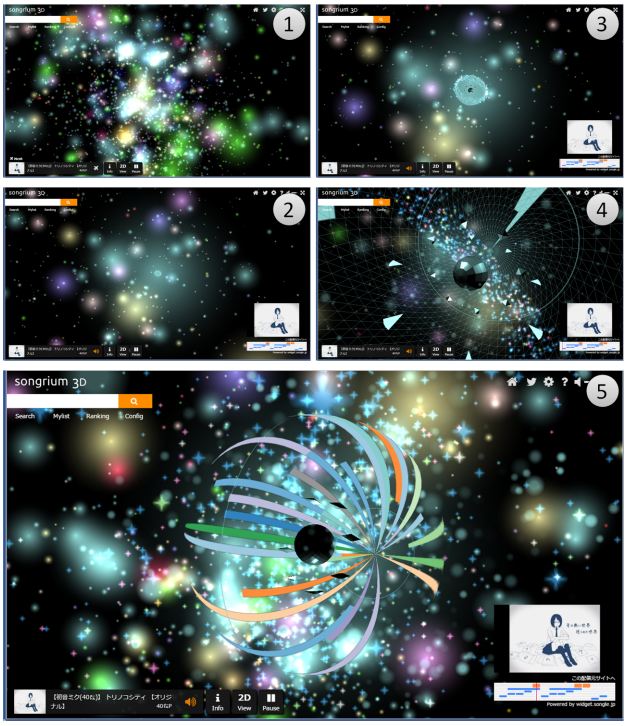


Figure 7. Transition from top page to a user-specified song in Songrium3D. All original VOCALOID songs are visualized in a three-dimensional space. The colors of these stars correspond to VOCALOID characters, with brightness indicating popularity (number of plays).

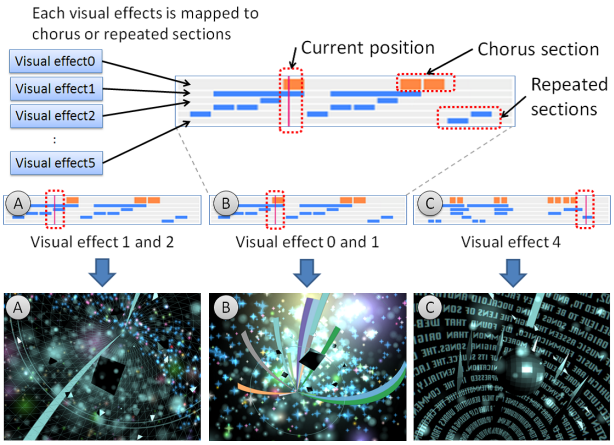


Figure 8. Examples of pair of visual effects and repeated sections. Each visual effects is mapped to chorus section or one repeated section. In the left one, the current time is in repeated section 1 and 2. Then users can see a mix of visual effects 1 and 2. Each visual effects will be started at the beginning of the repeated section.



Figure 10. Songrium3D on the live stage of Hatsune Miku. Animation generated by Songrium3D is shown in the gate-shaped LED display.

music data, a particular visualization technique that is often attempted is visualization of a music collection in a two-dimensional plane [7–9] and three-dimensional plane [10–13]. Our Music Star Map (see Section 3.3) and Songrium3D (see Section 4) are particular examples of this. Interactive interfaces are also important for user experiences; [14] assists a user in discovering songs. [15] assists a user in finding artists. In contrast to the advances in interactive music browsing described above, Songrium visualizes not only original songs, but also their derivate works and respective histories, facilitating the effortless browsing of web-native musical content. Furthermore, we apply our visualization methods to huge and dynamic music content and release them as a web application.

Music recommendation [16–18] is an automated method to give users the opportunity to encounter unfamiliar but potentially interesting songs. Similarly, automatic playlist generation [19–22] can provide such opportunities for users. Songrium also assists such user activities by visualizing massive user-generated music content. However, Celma reports the collaborative filtering approach which is a typical recommendation method that is prone to popularity bias [23]. It means that a tendency by which "The rich get richer" is reinforced by music recommendations. It is unsuitable for a browsing assistance of massive user-generated music content, but Kamalzadeh reports 50% of active listeners would like to choose songs one after another [24]. Furthermore, just 9% use online recommendation and 10% use shuffle when listening to a collection. This result indicates that active listeners enjoy not only listening to songs but also choosing songs. Regarding this point, visualizing massive user-generated music content can provide an excellent experience for active listeners, having a complementary relation with music recommendation.

8. CONCLUSIONS

As described in this paper, we proposed two new interfaces of a music browsing assistance service called *Songrium* that visualizes *VOCALOID* music including original songs and their derivative works on the video sharing site *Nicovideo*. Our target music content was *web-native music*: music content that was born, listened to, and distributed on the web. Songrium provides various visualization tools

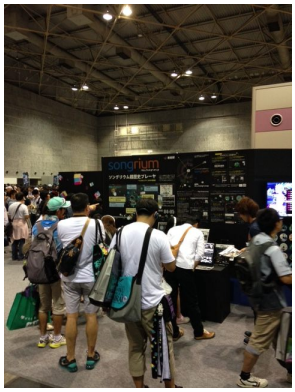



Figure 11. Demonstration of Interactive History Player in public event. The left photograph shows that a child used our system with touch panel display at public event for Niconico. The right photograph shows the booth of our demonstration at public event for Hatsune Miku.

to assist users in grasping the relations among web-native music. In particular, this paper features *Songrium3D* that shows whole of web-native music content in a community and *Interactive History Player* that presents a history of web-native music content. [9]

For future work, we will continue to run the Songrium service and improve it based on user feedback. Herein, we described only VOCALOID music, but web-native music is available from many other sources, which we hope to exploit in the near future.

Acknowledgments

We thank Keisuke Ishida for the web service implementation of Songrium. We also thank anonymous users of Songrium for editing social annotations. This work was supported in part by CREST, JST.

9. REFERENCES

[1] M. Hamasaki, M. Goto, and T. Nakano, "Songrium: A music browsing assistance service with interactive visualization and exploration of a web of music," in *Proc. WWW 2014 Companion*, 2014, pp. 523–528.

[2] M. Hamasaki and M. Goto, "Songrium: A music browsing assistance service based on visualization of massive open collaboration within music content creation community," in *Proc. of WikiSym/OpenSym 2013*, 2013, pp. 4:1–4:10.

[3] H. Kenmochi and H. Ohshita, "Vocaloid – commercial singing synthesizer based on sample concatenation," in *Proc. of Interspeech 2007*, 2007, pp. 4011–4010.

[4] 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 *Proc. of uxTV 2008*, 2008, pp. 165–168.

[5] M. Hamasaki, H. Takeda, T. Hope, and T. Nishimura, "Network analysis of an emergent massively collaborative creation community: How can people create videos collaboratively without collaboration?" in *Proc. of ICWSM 2009*, 2009, pp. 222–225.

[9] A demonstration video of the Songrium3D and Interactive History Player with English captions is available at https://staff.aist.go.jp/masahiro.hamasaki/SMC2015/ for SMC reviewers.

[6] M. Goto, "A chorus-section detection method for musical audio signals and its application to a music listening station," *IEEE Transaction on ASLP*, vol. 14, no. 5, pp. 1783–1794, 2006.

[7] E. Pampalk and S. Dixon, "Exploring music collections by browsing different views," *Computer Music Journal*, vol. 28, no. 2, pp. 49–62, 2004.

[8] M. Schedl, C. Hoglinger, and P. Knees, "Large-scale music exploration in hierarchically organized landscapes using prototypicality information," in *Proc. of ICMR 2011*, 2011, pp. 17–20.

[9] Z. Juhasz, "Low dimensional visualization of folk music systems using the self organizing cloud," in *Proc. of ISMIR 2011*, 2011, pp. 299–304.

[10] P. Knees, M. Schedl, T. Pohle, and G. Widmer, "An innovative three-dimensional user interface for exploring music collections enriched," in *Proc. of Multimedia 2006*, 2006, pp. 17–24.

[11] P. Lamere and D. Eck, "Using 3D visualizations to explore and discover music," in *Proc. of ISMIR 2007*, 2007, pp. 173–174.

[12] S. Leitich and M. Topf, "Globe of music - music library visualization using geosom," in *Proc. of ISMIR 2007*, 2007, pp. 167–170.

[13] A. Azcarraga and S. Manalili, "Design of a structured 3d som as a music archive," in *Proc. of WSOM 2011*, 2011, pp. 188–197.

[14] M. Goto and T. Goto, "Musicream: Integrated music-listening interface for active, flexible, and unexpected encounters with musical pieces," *IPSJ Journal*, vol. 50, no. 12, pp. 2923–2936, 2009.

[15] E. Pampalk and M. Goto, "Musicsun: A new approach to artist recommendation," in *Proc. of ISMIR 2007*, 2007, pp. 101–104.

[16] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 16, no. 2, pp. 435–447, 2008.

[17] Y. Saito and T. Itoh, "Musicube: a visual music recommendation system featuring interactive evolutionary computing," in *Proc. of VINCI 2011*, 2011, pp. 5:1–5:6.

[18] F. Ricci, "Context-aware music recommender systems: Workshop keynote abstract," in *Proc. of WWW 2012 Companion*, 2012, pp. 865–866.

[19] T. Pohle, E. Pampalk, and G. Widmer, "Generating similarity-based playlists using traveling salesman algorithms," in *Proc. of DAFx 2005*, 2005, pp. 220–225.

[20] C. Baccigalupo and E. Plaza, "Case-based sequential ordering of songs for playlist recommendation," in *Proc. of ECCBR 2006*, 2006, pp. 286–300.

[21] F. Maillet, D. Eck, G. Desjardins, and P. Lamere, "Steerable playlist generation by learning song similarity from radio station playlists," in *Proc. of ISMIR 2009*, 2009, pp. 345–350.

[22] J. L. Moore, S. Chen, T. Joachims, and D. Turnbull, "Learning to embed songs and tags for playlist prediction," in *Proc. of ISMIR 2012*, 2012, pp. 349–354.

[23] O. Celma and P. Cano, "From hits to niches?: or how popular artists can bias music recommendation and discovery," in *Proc. of the 2nd KDD Workshop on LargeScale Recommender Systems and the Netfrix Prize Competition*, 2008.

[24] M. Kamalzadeh, D. Baur, and T. Moller, "A survey on music listening and management behaviors," in *Proc. of ISMIR 2012*, 2012, pp. 373–378.

ARCHAEOLOGY AND VIRTUAL ACOUSTICS. A PAN FLUTE FROM ANCIENT EGYPT

Federico Avanzini, Sergio Canazza, Giovanni De Poli, Carlo Fantozzi, Niccolò Pretto, Antonio Rodà
Dept. of Information Engineering
University of Padova
name.surname@unipd.it

Ivana Angelini, Cinzia Bettineschi, Giulia Deotto, Emanuela Faresin, Alessandra Menegazzi, Gianmario Molin, Giuseppe Salemi, Paola Zanovello
Dept. of Cultural Heritage
University of Padova
name.surname@unipd.it
paola.zanovello.1@unipd.it

ABSTRACT

This paper presents the early developments of a recently started research project, aimed at studying from a multidisciplinary perspective an exceptionally well preserved ancient pan flute. A brief discussion of the history and iconography of pan flutes is provided, with a focus on Classical Greece. Then a set of non-invasive analyses are presented, which are based on 3D scanning and materials chemistry, and are the starting point to inspect the geometry, construction, age and geographical origin of the instrument. Based on the available measurements, a preliminary analysis of the instrument tuning is provided, which is also informed with elements of theory of ancient Greek music. Finally, the paper presents current work aimed at realizing an interactive museum installation that recreates a virtual flute and allows intuitive access to all these research facets.

1. INTRODUCTION

Sound and music computing (SMC) is a research field with an intrinsic vocation to multidisciplinarity, well exemplified in the project presented here, which combines a team of researchers in such fields as archaeology, 3D scanning and modeling, materials chemistry – as well as SMC – around a unique artistic artifact: an exceptionally well preserved ancient pan flute, probably of greek origins, recovered in Egypt in the 1930's and now exhibited in the Museum of Archaeological Sciences and Art (MSA), University of Padova. Presenting this musical instrument to the general public is a complex task, because of its multi-faceted nature. It is necessary to effectively communicate aspects related to history, iconography, acoustics, musicology, etc., as well as the research carried out during the project.

Starting from this case study, the project aims at defining a novel approach and methodology to "active preservation" of archeological artifacts, and specifically musical instruments. Preservation of documents is usually categorized into passive preservation, meant to protect the original documents from external agents without alterations, and active preservation, which involves data transfer from the analogue to the digital domain [1]. The traditional "preserve the original" paradigm has progressively shifted to the "distribution is preservation" idea of digitizing the content and making it available in digital libraries [2].

We aim at transposing these categories to the field of physical artifacts and musical instruments: passive preservation is meant to preserve the original instruments from external agents without altering the components, while active preservation involves a redesign of the instruments with new components or a virtual simulation, thus allowing access to them on a wide scale. These concepts may be summarized in a single "mission statement": we want to bring back to light archeological remains, but also to bring them back to life, with the aid of technology.

The final goal is to develop an installation that re-creates the instrument, allowing museum visitors to interact with it and its history. Achieving this goal requires truly multidisciplinary methodologies as it entails (i) studying the history and iconography of pan flutes, with a focus on Classical Greece; (ii) analyzing the geometry, construction, age and geographical origin of this artifact through non-invasive techniques such as 3D scanning and materials chemistry; (iii) studying its acoustics, timbre, and tuning, also by combining physics with elements of ancient Greek music theory; (iv) designing interactive installations that recreate a virtual flute allowing intuitive access to all these facets.

The remaining sections touch upon all of these points, with the main goal of illustrating the research methodologies and their potential, while only preliminary results obtained in the early months of the project will be discussed.

2. PAN FLUTES

2.1 A unique artifact

Amongst the archaeological items recovered during the recent reassessment of the MSA in Padova, there is an exceptional musical instrument, an ancient pan flute, probably of greek origins, consisting of 14 reeds of different lengths held together by ropes and a natural binder, and originally coated with a resin layer (now partially missing).

The artifact is one of several objects arrived in Padova thanks to archaeological researches of Carlo Anti, who directed the Italian Archaeological Mission in Egypt since

Copyright: ©2015 Federico Avanzini 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. the pan flute in the box for photographic plates, before restoration (photo by Team EgittoVeneto).



Figure 2. The restored flute (photo by Nicola Restauri).

1928, and led excavations in the ancient village of Tebtynis in the Fayum oasis, from 1930 to 1936, assisted by the italian-english archaeologist Gilbert Bagnani. The flute was stored in a box, originally made for photographic plates (see Fig. 1), which probably belonged to Bagnani, as documented by a short note in the interior. The box cover instead reports a sentence in French in the tiny handwriting of Bagnani's wife, which sets the original finding in Saqqara, from the area of the Mastaba n. XV, thus near Pepi II's tomb. A further information is found in Anti's archive and in a letter written by Evaristo Breccia (Director of the Archaeological Museum of Alexandria), in which he asked about this instrument which he saw in Tebtynis.

This origin is supported by the presence in Padova of other antiquities from Bagnani's campaigns, stored in small boxes like the one of the flute, and unlike other archaeological materials. Except for a few exceptions, the findings recovered at the MSA are from 1935, therefore this is probably the year of the discovery of this pan flute too.

The flute was first exhibited at the exhibit "Egypt in Veneto" (April-June 2013), in the section hosted at the MSA and devoted to "The excavations of Carlo Anti in Egypt". On this occasion, it underwent a major restoration programme for consolidation and preservation [3], as shown in Fig 2. This allowed not only to save the artifact but also to obtain the first analytical data useful to set the continuation of the research. In particular, infrared (IR) investigations found no evidence of earlier decorations, ultraviolet (UV) X and-ray investigations assessed the status of conservation, and chemical analysis tested the related techniques of construction. The flute is currently exhibited at MSA, in a dedicated show-case with air-tight and continuous monitoring of environmental conditions.

2.2 Related literature and iconography

Although its excellent preservation makes this artifact a unique archaeological item, literary and iconographic references to pan flutes are abundant in Greek-Roman world.

The syrinx (*syrìzo*: whistling, playing the bagpipes) appears in the most ancient Greek sources: in Homer's Iliad it is mentioned as an instrument related to the pastoral field (XVIII, 526) and festivals (X, 13), while in the Homeric Hymns it is connected to divine figures such as Hermes (IV) and Pan (XIX). In the Roman world both these aspects are recalled by several authors. In the Metamorphoses (I, 689-712) Ovid tells the story of the god Pan, when he saw the nymph Syrinx, devoted to Diana and so similar to the goddess that the two could not be distinguished. The nymph, at the sight of the monstrous body of the god, fled through inaccessible places, but had to stop on the swampy banks of the Ladon river, her father, where she prayed her sisters to disguise her in order not to be taken. When Pan reached her, all he found was a bundle of reeds. He sighed and the wind on the reeds produced a faint sound, a lament; the god, hit by its sweetness, said: "This conversation between you and me will last forever" and so "welded with wax some unequal reeds and the name of the girl lived forever." Thus, poetically, the invention of a simple and universal instrument is told.

Ovid in the Tristia (V, 10, 25) mentions pitch as another type of binder for the pipes. In his Onomastikon (IV, 69) Julius Pollux, who lived in Egypt during the 2nd century A.D., describes the syrinx as a structure "of many pipes" or "many sounds" formed by a series of reeds put together from the largest to the smallest and joined with flax and wax, leveled at one end and with a wing-like form. It is usually played by bringing it to the mouth and its musical potential is amazing: it is possible to play the flute, accompany with the flute, and stun with the flute. Pollux (IV, 77) also recalls that a "flute of many notes, discovery of Osiris" was in use among the ancient Egyptians.

In the Archaic period iconographic sources become even richer, both in Greek context, as in the François Vase depicted in Fig. 3(a), and in the Italic one, as in the contemporary (6th century B.C.) Certosa Situla in Bologna depicted in Fig. 3(b). Starting from the Hellenistic-Roman era, the representation of the pan flute spreads enormously, particularly in the Pompeian area: see Fig. 3(c). On the basis of the sources, it can be stated that until the classical period the instrument was quadrangular and made by pipes of equal external length (as in the François vase), while during the Hellenistic era the instrument was wing-shaped with unequal canes. The number of the elements is generally in the range $3-9$ during the Archaic period, $4-10$ in the classical period, and $4-18$ in the Hellenistic period: some Greek sources cite flutes with nine "voices", while the number seven is preferred in Latin authors [4, 5].

3. NON-INVASIVE ANALYSIS

3.1 3D laser scanning

A 3D model of the flute was acquired using non invasive and non-contact techniques. In order to inspect the sur-



(a) (b) (c)

Figure 3. Iconographic sources: (a) the François Vase; (b) the Certosa Situla in Bologna; (c) a fresco from the Villa of the Mysteries in Pompeii.

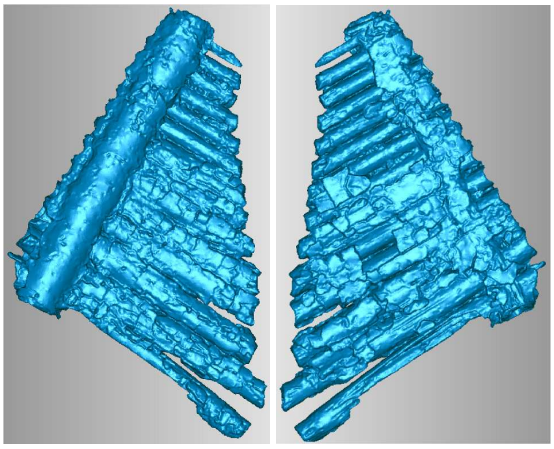


Figure 4. Very high resolution model of the flute.

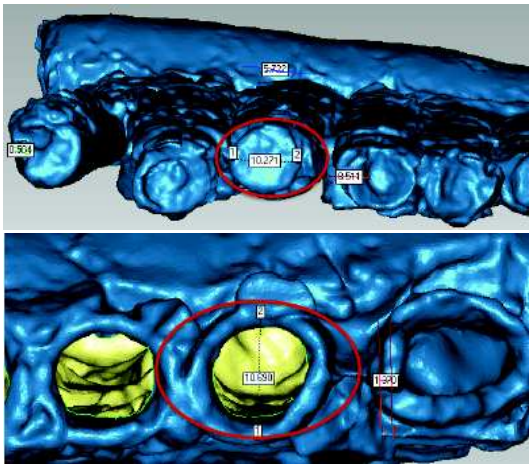


Figure 5. Examples of measurements of pipe diameters: d_x at the bottom end of the third pipe (upper panel), d_y at the top end of the second pipe (lower pipe).

faces (front and recto) and the border too, a ScanArm V3 from Faro was used. This is a seven-axis measurement system with a fully integrated laser scanner with a scan rate up to 19200 points/s and an accuracy of ± 35 μm. The field depth is 85 mm, and up to 640 points/row can be acquired. We followed a rather standard processing pipeline, which started with raw data acquisition (more than 4.5 million points for each side). At decimation of triangle meshes, more than 470000 triangles were obtained for each side.

In the alignment phase, various scans from different views were mosaicked to obtain the fused model that can be studied in a virtual space performing also metric measurements. In the post-processing phase, additional tools (specifically, Mesh Doctor in the Geomagic software environment) were used to fill holes and to automatically detect and correct errors in the polygonal mesh. As a result a very high resolution model composed by 920152 triangles was obtained, which is shown in Fig. 4.

Metric measurements were performed on this model in order to extract the main relevant parameters for subsequent analysis of the flute acoustics and tuning. Specifically, for each pipe the external length l and the diameter d were estimated. Additionally, in order to obtain a more reliable estimate for the diameter, for each pipe it was estimated along the x-axis (d_x) and the y-axis (d_y), both at the top and at the bottom ends of each pipe. Figure 5 shows two example measures of d_x and d_y.

Table 1 reports the estimated external lengths and diameters for all the 14 pipes.

3.2 Analysis of the pipe coating

Non-destructive mineralogical investigations were carried out on the two surfaces of a fragment of the coating, by X-ray Diffraction (XRD) coupled with Scanning Electron Microscopy (SEM) and Energy Dispersive Spectroscopy (EDS). The external surface (see Fig. 6) shows contaminations by soil sediments (quartz, calcite, anhydrite, kaolinite, albite) and the presence of evaporitic minerals like gypsum and halite (commonly known as rock salt), which are of particular interest since they correlate strongly to the depositional context: the presence of halite and gypsum suggests a depositional context rich in water and with a high evaporation rate like the Fayum oasis. The internal surface also shows quartz, calcite, albite, and wedellite, a common authigenic calcium oxalate related to the reaction between soil and organic matter. Microchemical investigations through SEM-EDS highlighted that halite is distributed all around the sample as shown by the white plaques (see Fig. 7). The sample has high concentrations

Pipe	l	d_x (bottom)	d_y (bottom)	d_x (top)	d_y (top)
1	145.56	11.983	11.404	17.981	11.292
2	144.563	12.943	10.089	10.635	10.69
3	127.976	10.271	10.481	10.798	10.366
4	117.315	10.742	9.559	10.671	10.748
5	110.685	11.787	9.243	10.037	9.706
6	96.512	7.394	8.417	9.097	8.98
7	86.397	9.535	6.981	7.787	9.672
8	81.327	8.704	6.546	8.446	8.663
9	71.795	7.251	7.129	6.86	8.109
10	64.341	6.566	6.093	7.63	7.629
11	58.862	6.889	6.222	7.561	6.879
12	51.42	5.616	5.795	5.989	6.562
13	49.554	5.838	5.307	5.423	6.042
14	43.655			4.659	4.731

Table 1. Main flute parameters extracted from the model. All lengths are expressed in mm.

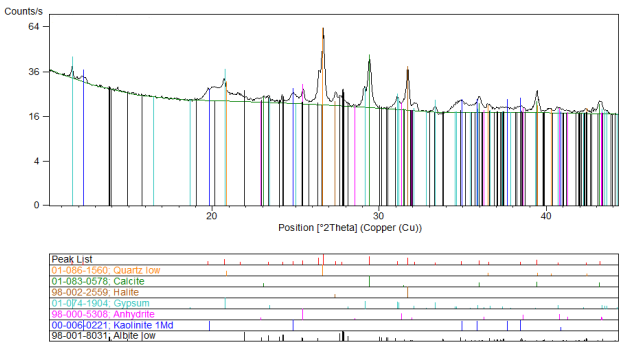


Figure 6. Diffraction pattern of the investigated sample, external surface

of carbon and oxygen, and this provides evidence that the coating is an organic compound.

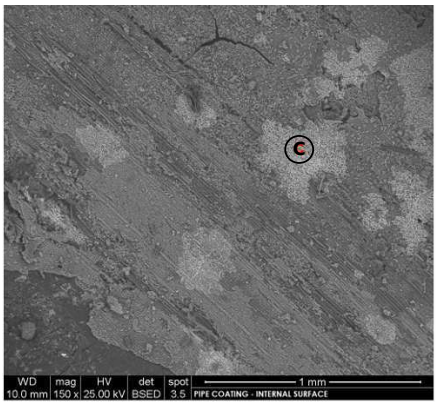
The sample is mainly constituted by an homogenous, lustrous, brittle, brown matrix of organic nature. It is likely to be resin, as confirmed by the elemental composition, but it may also be pitch, tar or bitumen. The investigation on the chemical composition of the coating is still in progress using IR and Raman analysis, and possibly Gas Chromatography-Mass Spectrometry (GC-MS) for the identification of the organic compound. The response will be of high interest to understand the production techniques used for this type of ancient musical instruments, also in comparison to what is known from classical literary sources.

The sample also exhibits a thin vegetal layer detached from the external surface of the pipes, therefore botanic investigations may determine the botanic species of the pipes. Once the nature of the coating matrix will be identified, it may be possible to carry out absolute ^{14}C carbon dating, if its vegetal derivation will be confirmed.

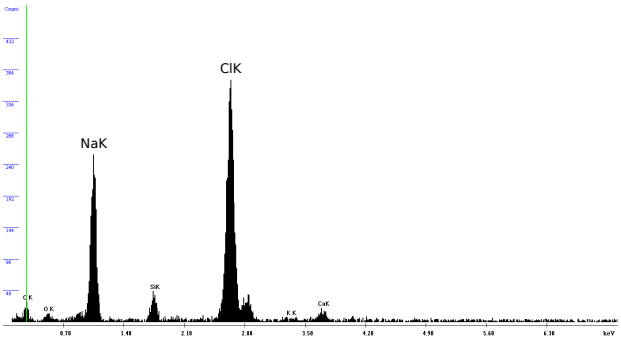
4. THE VIRTUAL FLUTE

4.1 Acoustics and tuning

The measurements discussed in the previous section are the starting point for an analysis of the tuning of the flute.



(a)



(b)

Figure 7. Microstructural and chemical analysis of the investigated sample: (a) SEM image showing white plaques of NaCl; (b) EDS spectrum of the white plaques in the area marked with a "C" showing high Na and Cl peaks.

While most flutes (the transverse flute, the recorder, etc.) are made of jet-excited unstopped pipes (i.e. open at both ends), the pan flute is peculiar in that it is a stopped-pipe wind instrument, thus requiring ad-hoc examination of its aerodynamics [6]. For the sake of our analysis, the first important consequence is that the fundamental frequency f is half that of an unstopped pipe of the same length:

$$f = \frac{4c}{l_{\text{int}} + \Delta l} \quad \text{Hz}, \tag{1}$$

where c is the sound velocity, l_{int} is the internal pipe length, and $\Delta l \sim 0.305 d_{\text{int}}$ is the length correction at the open end, proportional to the internal pipe diameter d_{int} [7, Ch. 8].

Unfortunately currently available metric measurements do not allow to infer reliable estimates of the internal lengths l_{int}. In fact, it is known that these were reduced by carefully increasing the thickness of the closures through addition of wax or propolis or other organic materials [8], thus achieving fine tuning. Computed axial tomography (CAT scan) may show the internal thickness of the occluding organic material, but this analysis has not been performed yet.

Given the limitations of currently available data, our preliminary estimation of tone frequencies uses Eq. (1) where l_{int} is estimated from Table 1 and a 5 mm-thick closure is assumed at the bottom of the pipes. Internal pipe diameters d_{int} are also estimated from Table 1, by averaging the four measures $d_{x,y}$ at the bottom/top ends, and then subtracting the wall thickness (estimated on average in 1.5 mm).

Pipe	2	3	4	5	6
f Hz	588.25	666.10	726.30	770.72	895.84
Pipe	7	8	9	10	11
f Hz	1001.14	1068.32	1222.43	1373.71	1507.45

Table 2. Tone frequencies estimated from current metric measurements.

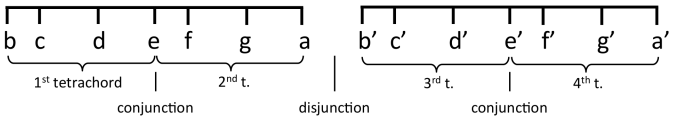


Figure 8. The Greater Complete System.

Table 2 reports the estimated tone frequencies for pipes from 2 to 11 only, since the first pipe is broken and since the approximation in current metric measurements introduces large errors in shorter pipes (from 12 to 14). The estimated frequency for pipe 2 corresponds approximately to a D5 in modern terms. The first obvious observation is that the tuning is consistent with a heptatonic scale, as the pipe pairs $2-9, 3-10, 4-11$ are all in a slightly sharp octave ratio (more precisely, the relation $f(\text{pipe}+7)/f(\text{pipe}) \sim 2.07$ holds for pipe$= 2:4$). This slight and uniform detuning may be due to our use of constant 5 mm-thick closures, which overestimate the frequencies of shorter pipes.

However we believe that, even in presence of more accurate measurements, a thourough analysis of the tuning of the flute cannot be based solely on acoustics, but needs to be informed with elements of theory of ancient Greek music. Melodic structures of this music are known [5, 9] to be based on the *tetrachord*, a series of four notes with the extremes tuned at a perfect fourth, i.e with a $4:3$ pitch ratio. While this interval is fixed, internal intervals can vary, giving rise to different *genera* of tetrachord. Each has variants, called *chroai* (shades), whose tunings comprise a large set of intervals, including 1/4-, 1/3-, and 7/6-tones. Two tetrachords can be linked together either by *synaphē* (conjunction), when the highest note of the first tetrachord coincides with the lowest of the second, or *diazeuxis* (disjunction), when there is a 1-tone interval between the two tetrachords. More complex structures (or *systems*) are derived by using more tetrachords. For example, a sequence of four tetrachords with the first-second and third-fourth in conjunction, and the second-third in disjunction, produces the *Greater Complete System* (see Fig. 8, where the relation between letters and tones is purely conventional and does not correspond to modern music notation system.).

More details about ancient Greek music are beyond the scope of this paper, but what is relevant here is that a particular system is characterized by a limited number of fixed pitches, corresponding to the outer notes of the tetrachords that compose the system, and tuned to a perfect fourth. Moreover, instruments were often tuned through a process called *lēpsis dia symphonias* (acquiring by concord) [9], i.e. by tuning in perfect fifths $(3:2)$ and fourths $(4:3)$,

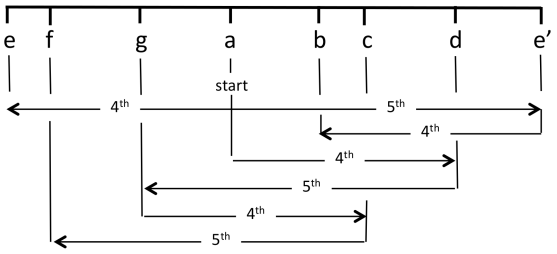


Figure 9. The "acquiring by concord" tuning method.

whereas tones were obtained by subtracting a fourth from a fifth, which correspond to a ratio of $9:8$ (see Fig. 9. Once more accurate estimates of pipe frequencies will be available, it will be possible to follow this method and search for $4:3$ and $3:2$ frequency ratios in order to reconstruct the tetrachord structure of the flute tuning.

4.2 An interactive multimedia installation

An important project outcome, from both the scientific and the dissemination viewpoints, is the realization of interactive applications that allow museum visitors to manipulate a virtual model of the flute, as well as to access historical and archaeological documentation about the instrument, such as photos, videos, and contextual information. A system composed of two applications is being designed.

The first application is an interactive multimedia installation allowing visitors to explore the artifact through the 3D model, since the original instrument is only weakly exposed to light for conservation purposes. The visitor controls the model with his hands (finger movements are tracked by an infrared sensor) while observing it on a monitor and, at the same time, playing the flute by blowing into a microphone. Furthermore, the visitor can choose different versions of the flute: the first based on the current state of the flute, the others based on virtual restaurations of the instrument that integrate the knowledge acquired during musicological and historical studies with the original model. By combining multimedia information about history of the ancient instrument with the musical practice, the installation is expected to provide an innovative and meaningful solution for informal learning.

In addition a mobile application will be freely downloadable by visitors on their own devices. Smartphones and tablets offer unprecedented multimedia and multisensory capabilities, being endowed with a wide range of sensors and input devices, and non-negligible computing power. Consequently mobile devices are finding significant applications in the virtual reconstruction of environments [10] and physical objects [11]. Apps for musical cultural heritage are a particularly interesting domain [12]. A mobile application with a skeuomorphic interface (i.e., one that leverages on the appearance and behavior of the physical artifact) is being designed: after the visit at the flute, visitors can study the history of the instrument and the music of its time from the installation described above, try to play some tunes handling the virtual model, and learn more complicated tunes studying on their own device, using it

as a pan flute thanks to a *sensor fusion* approach that integrates data from the built-in camera, accelerometer and gyroscope to track movements and select the correct virtual pipe in front of the mouth. The microphone detects the attack envelope and the intensity of the breath.

5. CONCLUSIONS

The work presented in this paper is in its early stages. The main foreseen developments in the short term are a more refined analysis of the instrument tuning, which uses additional non-invasive measures (particularly CAT scan to estimate internal pipe lengths), and exploits elements of music theory of ancient Greek music as discussed in Sec. 4.1. The applications discussed in Sec. 4.2 are also under development and will be ready for visitors by summer 2016.

In the mid term we expect that the available data and results will fuel several further developments. One is sound synthesis of the pan flute by means of physical modeling approaches, to be integrated in the installations (which currently employ wavetable synthesis) in order to increase their interactivity. This is an interesting research topic *per se*, since to our knowledge there is only one previous study on sound synthesis of the pan flute in the literature [13].

A very high resolution 3D model may also be exploited for computationally intensive (i.e., based on finite differences or finite elements) approaches to acoustic simulations and sound synthesis, which have started to unveil their potential in recent years [14]. One further current research trend is 3D printing of musical instruments [15]: a "digitally restored" 3D model of the pan flute can be 3D printed and sensorized, thus becoming a tangible interface that recreates the physicality of the original instrument. A similar approach has been recently adopted by some of the authors for the case of electrophone instruments [16].

It is also worth mentioning that, since little is known about ancient greek music (and almost exclusively from treatises), such a well preserved instrument may serve as an important testbed for currently accepted theories. Even most importantly, we believe that, being the pan flute a primeval instrument which is widespread in different cultures worldwide, the impact of this research goes beyond this particular exemplary. The proposed "active preservation" approach developed for the project can be applied to other ancient and prehistoric musical instruments.

Acknowledgments

This work was supported by the research projects *Archaeology & Virtual Acoustics*, Univ. of Padova, under grant no. CPDA133925, and *BiD-Algo*, Univ. of Padova, under grant no. CPDA121378.

6. REFERENCES

[1] F. Bressan and S. Canazza, "A systemic approach to the preservation of audio documents: Methodology and software tools," *Journal of Electrical and Computer Engineering*, vol. 2013, p. 21 pages, 2013.

[2] E. Cohen, "Preservation of audio in folk heritage collections in crisis," *Proceedings of Council on Library and Information Resources*, pp. 65–82, 2001.

[3] A. Menegazzi, M. Cesaretto, E. M. Ciampini, and P. Zanovello, "La scatola misteriosa," in *Egitto in Veneto*, P. Zanovello and E. M. Ciampini, Eds., Padova, 2013, pp. 91–104.

[4] C. Sachs, *The history of musical instruments*, 1st ed. W W Norton & Co Inc (Np), 1940.

[5] M. L. West, *Ancient Greek Music*. Oxford: Oxford University Press, 1992.

[6] N. H. Fletcher, "Stopped-pipe wind instruments: Acoustics of the panpipes," *J. Acoust. Soc. Am.*, vol. 117, no. 1, pp. 370–374, Jan. 2005.

[7] N. H. Fletcher and T. D. Rossing, *The physics of musical instruments*. New York: Springer-Verlag, 1991.

[8] E. Civallero, *Introducción a las flautas de pan*, 1st ed., Madrid, 2013, Creative Commons.

[9] J. G. Landels, *Music in ancient Greece and Rome*. Routledge, 1999.

[10] J. Thomas, R. Bashyal, S. Goldstein, and E. Suma, "Muvr: A multi-user virtual reality platform," in *Virtual Reality (VR), 2014 iEEE*, March 2014, pp. 115–116.

[11] M. Figueiredo, P. Cardoso, C. Goncalves, and J. Rodrigues, "Augmented reality and holograms for the visualization of mechanical engineering parts," in *Information Visualisation (IV), 2014 18th International Conference on*, July 2014, pp. 368–373.

[12] S. Canazza, C. Fantozzi, and N. Pretto, "Accessing tape music documents on mobile devices," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. Accepted for publication, 2015.

[13] A. Czyżewski, J. Jaroszuk, and B. Kostek, "Digital waveguide models of the panpipes," *Archives of Acoustics*, vol. 27, no. 4, pp. 303–317, 2002.

[14] S. Bilbao, B. Hamilton, A. Torin, C. Webb, P. Graham, A. Gray, K. Kavoussanakis, and J. Perry, "Large scale physical modeling sound synthesis," in *Proc. Stockholm Musical Acoustics Conf. (SMAC 2013)*, Stockholm, Aug. 2013, pp. 593–600.

[15] A. Zoran, "The 3d printed flute: Digital fabrication and design of musical instruments," *J. New Music Res.*, vol. 40, no. 4, pp. 379–387, 2011.

[16] F. Avanzini and S. Canazza, "Virtual analogue instruments: an approach to active preservation of the studio di fonologia musicale," in *The Studio di Fonologia - A Musical Journey*, M. Novati and J. Dack, Eds. Milano: Ricordi (MGB Hal Leonard), June 2012, pp. 89–108.

DEVELOPING MIXER-STYLE CONTROLLERS BASED ON ARDUINO / TEENSY MICROCONTROLLERS

Dr. Constantin Popp
constantin@knobtronix.co.uk

MSc Eng. Rosalia Soria-Luz
rosalia@knobtronix.co.uk

ABSTRACT

Low-cost MIDI mixer-style controllers may not lend themselves to the performance practice of electroacoustic music. This is due to the limited bit depth in which values of controls are transmitted and potentially the size and layouts of control elements, providing only coarse control of sound processes running on a computer. As professional controllers with higher resolution and higher quality controls are more costly and possibly rely on proprietary protocols, the paper investigates the development process of custom DIY controllers based on the Arduino and Teensy 3.1 micro controllers, and Open Source software. In particular, the paper discusses the challenges of building higher resolution controllers on a restricted budget with regard to component selection, printed circuit board and enclosure design. The solutions, compromises and outcomes are presented and analysed in fader-based and knob-based prototypes.

1. INTRODUCTION

In their performance practise the authors use mixer-style controllers to diffuse and improvise electroacoustic music, in particular the Korg nanoKONTROL [1] and a Behringer BCF2000 [2]. The two controllers are readily available and immediately compatible with computer music software, as they are relying on the MIDI protocol for data transfer. Although both provide plenty controls to adjust sound processes running in the computer, they transmit the controls in 7 bit and therefore may not lend themselves to nuanced control. Furthermore, the nanoKONTROL also compromises tactility for compactness with it's short 45 mm faders and small hard-touch knobs (Figure 1).

Other controllers, such as the Mackie Control Universal Pro XT [3] may be more touch friendly and would offer higher resolution but they are in comparison expensive (starting around 700 pounds) and the used protocols are closed source / proprietary.

While others have investigated the departure from a mixer-style controller using, among others, accelerometers [4], capacitive touch [5] or optical sensors [6], the authors sought for an incremental improvement, focusing mainly on improving the resolution and layout of the controller with respect to the nanoKONTROL and the BCF2000. To solve these issues, the authors decided to develop their own mixer-style controllers.

Copyright: ©2015 Dr. Constantin Popp 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. Size comparison of a Korg nanoKONTROL with a Behringer BCF2000 and 10 British pence.

2. GENERAL DESIGN PHILOSOPHY

2.1 Choosing a platform

Because of the accessibility and wide-spread use of the Arduino platform, the Teensy 3.1 from PJRC [7] and the Sparkfun Pro Micro [8] were chosen. Both are only around 35.56 mm by 17.78 mm small and can be programmed using the Arduino IDE. The community around the Arduino developed already Open Source software libraries for USB-MIDI [9], OSC [10] and network communication [11], solving the need to deal with communication protocols manually.

2.2 Hardware and software considerations

The authors sought for a scalable hard- and software solution accommodating a variable number of knobs, faders and switches. That suggested the use of sixteen channel analog multiplexers (Texas Instruments CD74HC4067E) for reading the voltages of analog potentiometers and eight channel digital multiplexers (Texas Instruments SN74HC138N and SN74HC151N) for reading and setting the state of switches and LEDs, reducing the need of a high number of input and output pins on the microcontroller. The microcontroller then connects to the computer via USB and draws the necessary power from it. Figure 2 shows a schematic diagram

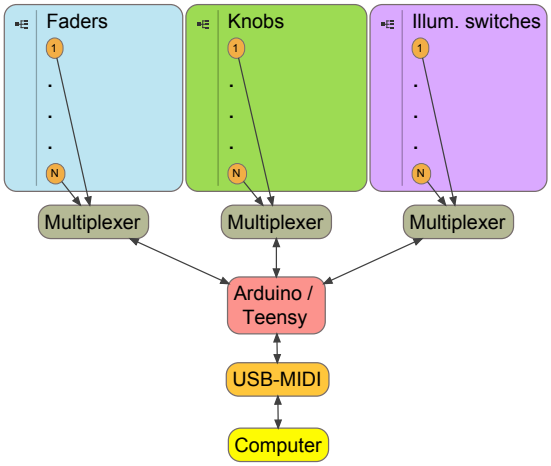


Figure 2. Schematic overview of the hardware design.

of the hardware implementation. The software side deals with signal conditioning, I/O management (controlling the multiplexers, sending and receiving data via USB-MIDI, ADC conversion) and controller configuration.

Capturing and transmitting values of the knobs and faders with more than 7 bit affects the hard- and software design. The Arduino offers analog to digital conversion in 10 bit resolution, whereas the Teensy 3.1 up to 13 bit. The authors decided to transmit the captured values to the computer using the MIDI protocol. Two consecutive control change messages transmit the 10 bit (11 bit in case of the Teensy 3.1) values with the higher bits arriving first, the lower bits last. That way, the convenience of MIDI could be used, relieving the need to use OSC just out of consideration of bit depth. Alternative ways to transmit the data could be implemented as the firmware is Open Source.

The development of initial prototypes – a knobbox and a faderbox – was straight forward (Figure 3). However, making the prototypes more usable and reliable opened up new challenges.

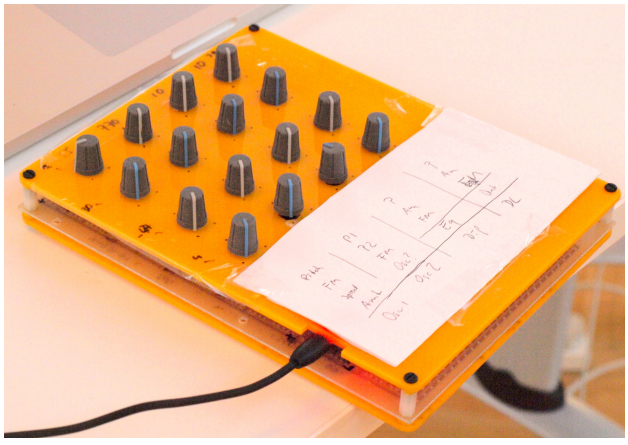


Figure 3. Initial prototypes of the knobbox (top) and faderbox (bottom).

3. CHALLENGES

With regard to the hardware and enclosure, several aspects made the development complicated.

3.1 Hardware

Firstly, we relied on online shopping for finding and purchasing parts. To determine if a part suits the requirements it needs to be bought and tested in case it's datasheet sounds promising. This was especially problematic for finding the desired combination of knobs and potentiometers as not every knob fits every potentiometer and not every knobs feels touch friendly. In the end 9 mm ALPS potentiometer with a 6 mm D shaft were chosen and paired with Multicomp soft touch knobs (CR-BA-7C6-180D) (Figure 4).

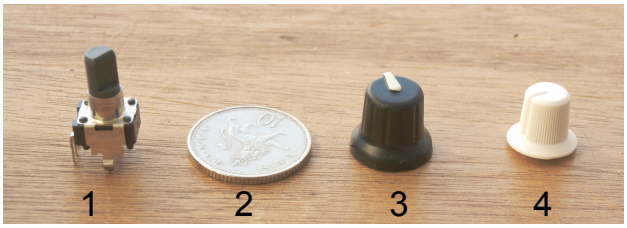


Figure 4. Size comparison of the potentiometer (1), 10 British pence (2), the CR-BA-7C6-180D (3) and the knob of a nanoKONTROL (4).

Secondly, finding a suitable printed circuit board (pcb) design software proved to be challenging. As the authors decided to use a custom made pcb to reduce noise in reading the potentiometers while facilitating construction and improving reliability, a pcb design software was needed. Due to licensing restrictions and costs, as well as no access to a Windows PC the authors were unable to use professional pcb design software (e.g. Eagle [12]) and decided to use Fritzing [13]. As a consequence the footprints of most of the parts used had to be designed manually as they weren't available in Fritzing's libraries. This process proved to be prone to errors and time consuming.

Thirdly, the specific nature of components create additional challenges as each design change means a redesign and re-manufacture of the pcb. For example, the micro USB-connector broke off from the cheap Arduino clone used in one of the prototypes after a few weeks, causing the authors to look for a different microcontroller. As the pin mapping differs from the various microcontroller types the board has to be redesigned [1]. Something similar happened with the knobs. One prototype was designed prior deciding on a suitable knob for the potentiometers. After many weeks of online searching and shopping we settled on a knob which required a different orientation of the potentiometer on the pcb, forcing us to redesign the board again. As the validity of the hardware ultimately needed to be tested in real life, each hardware iteration had to be manufactured again, costing the authors money and time.

[1] Compare the pinout mapping of the Teensy 3.1. [1] with the Pro Micro [14].

3.2 Enclosure

Since the authors prioritised a comfortable layout of potentiometers over standardised dimensions for enclosures, ready-made industrial enclosures were not available. Further, customisation of said enclosures, i.e. drilling holes or engraving lattices for the potentiometers, through specialised companies would be prohibitive due to budget restrictions. Instead the authors opted for the DIY route here, too, using the tools and knowledge available to us, borrowing ideas found by other projects [15] [16]. This led the authors to base the design on a combination of slices of laser cut plywood and acrylic, as well as specially designed pcbs (Figure 5).



Figure 5. Corner view of the most recent version of the knobbox. The pcb serves as the face plate with the silkscreen and copper acting as labelling and markup. The frosted acrylic sheets envelope the plywood sheets to improve stiffness and design.

3.3 Budget

However, it is worth noting that some of these challenges could have been solved if a research budget would have been available for the development. A budget would have allowed the authors to hire knowledge and leverage access to professional manufacturing processes, including using CNC machines and laser cutters to process metals or injection moulds custom parts. Luckily, through help through our local Hackspace the authors found an affordable pcb manufacturing service freeing us from having to etch our own boards [17].

4. COMPROMISES AND OUTCOMES

The challenges in the design process and the specifics of the components lead to two controller types which differ in the way they solve the design requirements. One controller type uses a modular approach. Components are grouped on specialised, chainable boards which can be interconnected via cables (Figure 6). In the moment the authors created six different boards, each of them housing either the microcontroller and additional components, four 100 mm faders, 16 knobs, a pair of jack connectors for foot switches, an ethernet controller or 8 illuminated tactile switches. The other controller type uses a single pcb for all components.

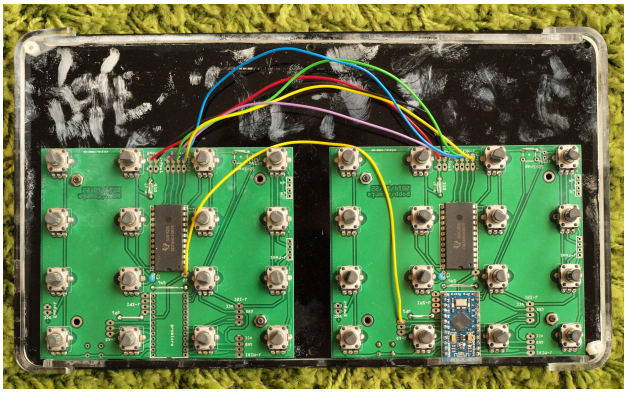


Figure 6. Example of two chained knob based boards.

The modular approach suits the 100 mm faders well, facilitating the creation of fader boxes. As the pcb manufacture service the authors use offers the production of boards in quantities of five or ten, with ten boards costly almost the same as 5 boards, spreading repeating controls over several copies of the same board proved to be cost effective. For example, 16 faders of a faderbox could be spread over four boards each housing four faders. These four faders fit well on a board of 14 by 11 cm and the ten boards cost approximately 43 pounds with leaving 6 boards in spare. In comparison, attempting to fit the 16 faders on a single board would have either exceeded the service's maximum pcb dimensions or cost ineffective (ten boards of 14 by 30 cm would cost approximately 71 pounds, leaving 9 in spare). Chaining smaller boards together offers the side benefit that a controller can be customised to the number of controls the clients wishes. In the two prototypes built using this approach the authors chose a combination of faders and knobs which allow the hands to rest on the knobs. The layout of the faders is made to accommodate different playing techniques (Figure 7). However, due to the size of the faders and spacing, the 16 channel fader box exceeds the typical backpack size (Figure 8).



Figure 7. Illustration of the fader layout.

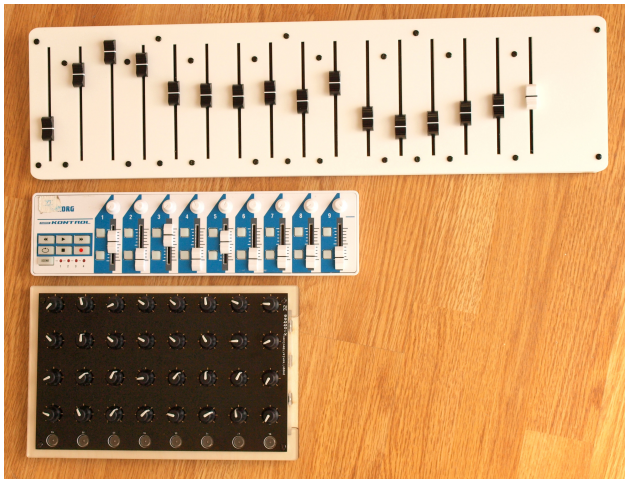


Figure 8. Size comparison between the 16 channel faderbox, nanoKONTROL and knobbox.

The single board solution suited the knobs well, leading to the creation of a knob box. As 9 mm knobs require a lot less space than 100 mm faders, a higher number of controls can be fitted on a small enough pcb, removing the need for chaining boards. However, in a single board situation the number of controls is not customisable per client anymore. In the moment, after a modular version, the authors settled on a pcb in the dimensions of approx. 24 cm by 15 cm, housing 32 knobs, 8 illuminated tactile switches, two jack sockets (expression pedal and MIDI out), as well as the microcontroller and additional circuitry (Figure 9, Figure 10). This made it possible to offer a high density of controls in a relatively compact way without having to sacrifice the layout and tactile feel of the controls while creating a device that is still backpack compatible. The layout of the potentiometers allows for adjusting adjacent potentiometers, prioritising the vertical over the horizontal (19 mm horizontally, 23 mm vertically). The knobs employed feature soft touch and round edges to improve tactile feel.

As may have become apparent, designing a controller also means finding a balance between competing criteria such as costs, type and number of controls, layout and portability. Each criteria may have different priority depending on the use scenario. In a sound diffusion context, large faders might be more useful as opposed to knobs, as around eight faders can be moved at once (4 per hand), instead of merely two knobs (1 per hand). A fader box for a diffusion system then would feature as many faders as possible and portability and costs (in terms of number of controls per space) might be less of an issue, compared with regard to the multiple reels of cables, stands and loudspeakers required to build a diffusion system. However in a electroacoustic improv context, portability and costs would be more important. Here knobs might be more useful, as they use less space and tend to be cheaper than large 100 mm faders, albeit smaller or fewer faders than knobs could be used.

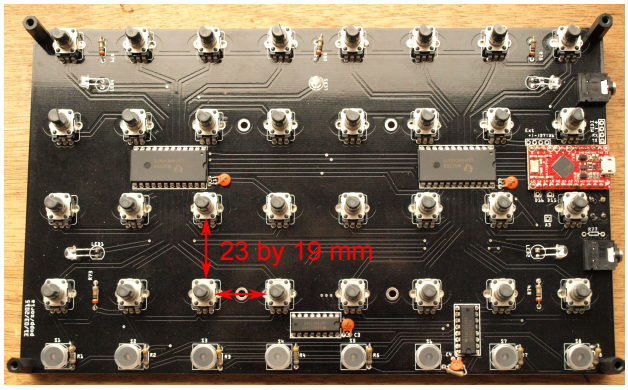


Figure 9. Populated most-recent knobbox pcb.



Figure 10. Isometric view of the knobbox.

Also, if a controller uses MIDI or OSC as a protocol equally depends on the priorities of the criteria or the use scenario. OSC via ethernet cables would require the use of a RJ-45 socket which in turn would require a lot of space, making a controller bulkier, especially when used in a combination with a readymade breakout board such as the Wiznet Wiz820io. Figure 11 illustrates how the RJ-45 socket would exceed the dimensions of the fader box. However, connecting a controller via ethernet would allow for scenarios where the computer is not in controller's proximity. Replacing ethernet with WIFI could solve the size issue, but WIFI hasn't been implemented, yet. In that sense, the authors decided to stay with MIDI purely for convenience, although the modular controller approach would support OSC if required.

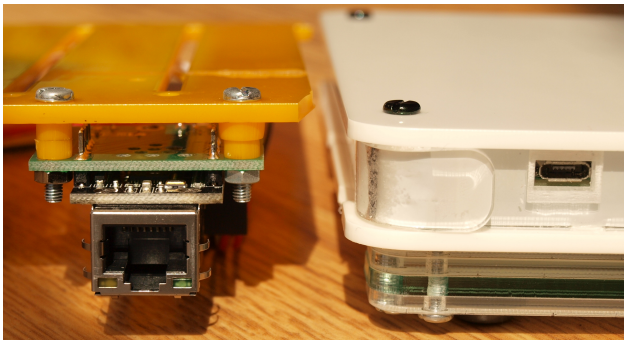


Figure 11. Size comparison between the ethernet module (left) and the faderbox (right).

5. CONCLUSION

The paper discussed the challenges and solutions of a DIY approach in building mixer-style controllers which suit the performance of electroacoustic music better. The DIY route posed own challenges, such as budget restrictions limiting access to components, materials and production processes, as well as knowledge. As the Arduino platform provides Open Source libraries for communication and I/O management, the controller's software implementation proved to be less challenging than the hardware design. The authors made their controller firmware Open Source and opened unused microcontroller I/O's on the pcb, inviting customisation and extension by the user.

The paper also presented two types of controllers which resulted from different compromises in solving the design challenges – a fader box and a knob box. Both controllers aim at different aspect of the electroacoustic performance practise with one mainly meant for sound diffusion, the other one for the improvisation of electroacoustic music via knobs.

The authors currently focus on the development on the knob box. In a future revision, it will also allow boards to be chained while replacing the Arduino Pro Micro with a Teensy LC [18], improving customisation through more processing power and higher I/O count compared to the Arduino. Furthermore, an extension pcb featuring eight 80 mm faders will be offered, to augment the knob based approach through faders.

Acknowledgments

The authors would like to thank Hackspace Manchester and Fab Lab Manchester for their kind support.

6. REFERENCES

[1] Korg, "nanoKONTROL2 SLIM-LINE USB CONTROLLER | MIDI Controllers | KORG," 2015. [Online]. Available: http://www.korg.com/us/products/controllers/nanokontrol2/

[2] Behringer, "Behringer: B-CONTROL FADER BCF2000," 2015. [Online]. Available: http://www.behringer.com/EN/Products/BCF2000.aspx

[3] LOUD Technologies Inc, "Mackie - Mackie Control Universal Pro," 2015. [Online]. Available: http://www.mackie.com/products/mcupro/

[4] J. E. Cobb, "An accelerometer based gestural capture system for performer based music composition," 2011. [Online]. Available: http://etheses.whiterose.ac.uk/2252/

[5] A. R. Jensenius, R. Koehly, and M. M. Wanderley, "Building low-cost music controllers," in *Computer Music Modeling and Retrieval*. Springer, 2006, pp. 123–129. [Online]. Available: http://link.springer.com/chapter/10.1007/11751069_11

[6] R. Graham and B. Bridges, "Gesture and Embodied Metaphor in Spatial Music Performance Systems Design," in *Proceedings of the Internation Conference on New Interfaces for Musical Expression (2014)*. Goldsmiths University of London, 2014, pp. 581–584. [Online]. Available: http://nime2014.org/proceedings/papers/526_paper.pdf

[7] PJRC, "Teensy 3.1: New Features," 2014. [Online]. Available: https://www.pjrc.com/teensy/teensy31.html#specs

[8] Sparkfun Electronics, "Pro Micro - 5V/16MHz," 2014. [Online]. Available: https://www.sparkfun.com/products/12640

[9] PJRC, "Teensyduino: Using USB Mouse with Teensy on the Arduino ID," 2012. [Online]. Available: https://www.pjrc.com/teensy/td_midi.html

[10] CNMAT, "OSC for Arduino and Embedded Processors," 2015. [Online]. Available: http://cnmat.berkeley.edu/oscuino

[11] Arduino LLC, "Arduino Playground - LibraryList," 2015. [Online]. Available: http://playground.arduino.cc/Main/LibraryList#Comm

[12] CadSoft, "Freeware | CadSoft EAGLE," 2011. [Online]. Available: http://www.cadsoftusa.com/download-eagle/freeware/

[13] IXDS, "Fritzing," 2010. [Online]. Available: http://fritzing.org/.

[14] Jimb0, "Pro Micro & Fio v3 Hookup Guide," 2014. [Online]. Available: https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide#hardware-overview-pro-micro

[15] SliceCase, "SliceCase," 2015. [Online]. Available: http://www.slicecase.com/about

[16] electrodacus, "Make a cool case out of PCB only," 2015. [Online]. Available: http://www.eevblog.com/forum/projects/make-a-cool-case-out-of-pcb-only/

[17] Elecrow, "10pcs- 2 layer PCB [SPP01010PP] - $9.90: Elecrow bazaar, Make your making more easy," 2015. [Online]. Available: http://www.elecrow.com/10pcs-2-layer-pcb-p-1175.html

[18] PJRC, "Teensy LC (Low Cost)," 2015. [Online]. Available: http://www.pjrc.com/teensy/teensyLC.html

Pop Music Visualization Based on Acoustic Features and Chord Progression Patterns Applying Dual Scatterplots

Misa Uehara
Ochanomizu University
misa@itolab.is.ocha.ac.jp

Takayuki Itoh
Ochanomizu University
itot@is.ocha.ac.jp

ABSTRACT

Visualization is an extremely useful tool to understand similarity among large number of tunes, or relationships of individual characteristics among artists, effectively in a short time. We expect chord progressions are beneficial in addition to acoustic features to understand the relationships among tunes; however, there have been few studies on visualization of music collections with the chord progression data. In this paper, we present a technique for integrated visualization of chord progression, meta information and acoustic features in collections of large number of tunes. This technique firstly calculates the acoustic feature values of the given set of tunes. At the same time, the technique collates typical chord progression patterns from the chord progressions of the tunes given as sequences of characters, and records which patterns are used in the tunes. Our implementation visualizes the above information applying the dual scatterplots, where one of the scatterplots arranges tunes based on their acoustic features, and the other figures co-occurrences among chord progression and meta information. In this paper, we introduce the experiment with tunes of 20 Japanese pop musicians using our visualization technique.

1. INTRODUCTION

Thanks to the enlargement of storage of music players, now we can bring a large number of tunes in our daily life. Also, recent on-line music delivery services enabled us easier to find favorite tunes without visiting off-line music stores. On the other hand, it is not always easy to quickly understand which tunes are preferable for users. Though various music recommendation techniques have been developed, it is not still easy for users to understand how the recommended tunes are estimated as preferable for them. We expect visualization is an effective and intuitive approach to make users overview and understand the relevancy or similarity of particular tunes or artists easily.

There are many factors which affect user's preference or impression of pop music. Acoustic features and chord progression are typical musical factors, while visual factors such as fashions and verbal factors such as lyrics are also important for the preferences and impressions of the tunes. This paper presents our visualization technique for overview and exploration of relevancy among acoustic features, chord progression patterns, and meta information of pop tunes. The presented visualization technique displays the following two scatterplots side-by-side, as illustrated in Figure 1:

Copyright: ©2015 Misa Uehara 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.

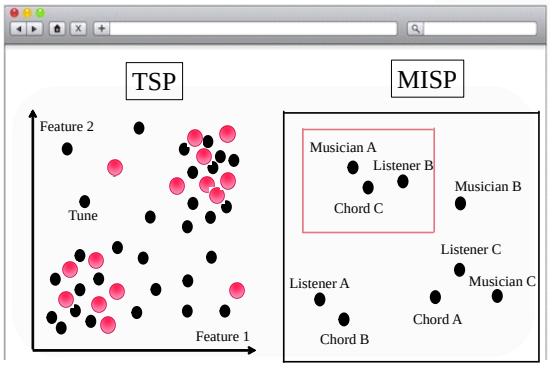


Figure 1. Illustration of the structure of the presented visualization technique. It features the tune scatterplot (TSP) in the left side of the window, and the meta information scatterplot (MISP) in the right side. When a user interactively selects a set of meta information in MISP as enclosed by a pink rectangle, dots corresponding to the selected meta information are colored in TSP.

Tune scatterplot (TSP): The scatterplot visualizing a set of tunes with their acoustic feature values. Tunes correspond to dots one-by-one in this scatterplot.

Meta Information Scatterplot (MISP): The scatterplot visualizing the co-occurrence of meta information including artist names, preferred tunes of particular listeners, and chord progression patterns. Meta information corresponds to dots one-by-one in this scatterplot.

Two of the acoustic features are automatically or manually assigned to the two axes of TSP, and the dots corresponding to the tunes are displayed in TSP. Meanwhile, a dimension reduction scheme is applied to the set of meta information including chord progression patterns to place them as dots in MISP.

We suppose the following operation scenario. Users firstly look at MISP, and then interactively select a set of closely

displayed dots corresponding to well-correlated meta information (e.g. artist name) and chord progression patterns. The technique assigns independent colors to each of the selected dots in MISP At the same time, tunes in TSP corresponding to the manually selected dots in MISP are also colored. Interactively selecting the acoustic features which are to be assigned to the axes of TSP, we can discover relationships among meta information, chord progression patterns, and acoustic features.

We expect this visualization technique can be used for various purposes. Students majoring in pop music analysis can study the trends related to acoustic features and chord progression patterns. Consumers can interactively explore their favorite tunes. Marketing experts can discuss how to optimize the music recommendation services by observing the visualization results.

2. RELATED WORK ON MUSIC VISUALIZATION

Visualization is a useful tool to briefly understand the contents of music, and actually several survey or tutorial presentations have been presented [2] [6]. Most of music visualization techniques can be divided into the following two categories: visualizing the detail of one tune [12] [5], and visualizing collections of large number of tunes. This paper discusses the latter category of visualization techniques.

There have been a lot of techniques for visualizing collections of large number of tunes focusing on similarity of their acoustic features. Pampalk [9] represented acoustic similarity of sets of tunes by applying self organizing map (SOM) and a graphical metaphor of islands. Goto et al. [3] represented the sets of tunes as moving objects so that users can interactively catch and play the interested tunes. Leitich et al. [8] applied GeoSOM to display a set of tunes onto a sphere based on the similarity of spectrum descriptor of the tunes. Kusama et al. [7] applied an abstract image generation technique and a zooming user interface to intuitively explore the hierarchically structured tunes. Acoustic similarity is also applied to visualize the similarity of artists [10] in addition to above studies to represent the similarity of tunes. These techniques apply acoustic analysis to organize the sets of tunes; however, it is difficult to directly read the acoustic features from their representation. It is easier to directly read the acoustic features if users can select important features and assign them to axes in the display spaces [11] [13]. However, these studies did not apply the knowledge related to chord progressions.

There have been several studies on visualization of music structure of a single tune with chord information [1] [4]. On the other hand, there have been few studies on visualization of large number of tunes applying combination of acoustic features and chord information.

3. PRESENTED VISUALIZATION TECHNIQUE

This section describes the processing flow of the presented visualization technique. The technique consists of four technical components: acoustic feature calculation, chord progression pattern matching, scatterplot construction, and interaction.

3.1 Acoustic Feature

We suppose to calculate the acoustic feature values of the given set of tunes as a preprocessing. Currently we apply MIRtoolbox [14] to calculate the following feature values.
RMS energy is the root-mean-square of the acoustic energy. This value tends to be higher while applying of recent pop, rock, or electric music, because their acoustic power is controlled as nearly constant by electric effects such as compressor and limiter. On the other hand, this value tends to be lower while applying to ballads, classical music, and other non-electric music, because their acoustic power varies along their developments. Consequently, this value is useful to divide the tunes according to their genres or instruments.
Tempo can be calculated from the cyclic patterns of power peak or harmony change. We believe tempo is important information to estimate the preference of music listeners.
Brightness is the ratio of acoustic energy of 1500Hz or higher frequency, which is mainly brought from overtones of instruments. This value is useful to divide tunes according to orchestration or recording settings: it tends to be higher if instruments which sound rich overtones (e.g. violin, saxophone, and cymbal) are effectively used by the arrangements of the tunes.
Mode is the ratio of time occupied by major or minor harmonies. This value is useful to divide enjoyable and sad sounds of the music.
Spectral irregularity is the degree of variation of the successive peaks of the spectrum. This value is useful to measure the dynamics of music.
Inharmonicity is the amount of energy outside the ideal harmonic series. This value is useful to divide traditional and modern music, because inharmonic tones are relatively often used by modern classical music, jazz, and contemporary pop music.

3.2 Chord Progression Pattern Matching

We suppose that chord progression information is provided as sequences of characters for each of given tunes. Currently we use the chord progression database for Japanese pop music on the Web [15]. Our implementation then transposes chord progressions of all the tunes to C-major or A-minor as a preprocessing.

The technique also supposes that several typical chord progression patterns are provided. Table 1 shows examples of the typical chord progression patterns used in many Japanese pop songs. Our implementation collates the prepared typical patterns with the chord progression of the tunes, and records which patterns are used in the tunes.

3.3 Scatterplot construction

The presented visualization technique supposes the set of tunes as $T = \{t_1, t_2, ..., t_m\}$, where t_i is the i-th tune, and m is the total number of tunes. A tune t_i has the values $t_i = \{f_{i1}, f_{i2}, ..., f_{in_F}, c_{i1}, c_{i2}, ..., c_{in_C}\}$, where f_{ij} is the

Table 1. Examples of typical chord progression patterns.

1	C F G
2	F G7 Em Am
3	Am F G C
4	Am Dm G Am
5	C Am F G7
6	F G Am Am
7	C Am Dm G7
8	Am Em F G7
9	C G Am Em F C F G

j-th acoustic feature value, c_{ij} is the j-th meta information value, n_F is the number of acoustic feature values, and n_C is the number of meta information values. Meta information value is a boolean variable regarding various attributes such as chord progression pattern, artist name, and preference of a listener. The corresponding value will be true if the tune contains the specific chord progression pattern, or if the specific listener prefers the tune.

TSP displays the set of tunes as m dots. The positions of the dots are calculated when the two acoustic features are assigned to the horizontal and vertical axes.

MISP represents the relevancy among meta information values as n_C dots. The technique calculates the distances between arbitrary pairs of meta information values, and applies a dimension reduction scheme to calculate the positions of the dots corresponding to the meta information values. Our current implementation calculates the distance between the u-th and v-th meta information values as $d_{uv} = 1-m_{uv}/m$, where m_{uv} denotes the number of tunes which both u-th and v-th values are true. Then, it simply applies multidimensional scaling (MDS) to calculate the positions. This scatterplot can represent various trends of meta information, including co-occurrence of multiple chord progression patterns, and preference of chord progression patterns of artists or listeners.

3.4 Interaction

Our implementation features TSP in the left side, and MISP in the right side of the window. It also features the following interaction mechanisms.

Selection of meta information. This implementation provides an interaction to drag the pointing devices in MISP to select dots corresponding to the meta information. It assigns independent colors to the selected dots in MISP, and to the dots corresponding to the tunes in TSP.

Suppose two dots in MISP corresponding to the u-th and v-th meta information are selected by the drag operation, and blue and red are assigned to the two dots respectively. This implementation then assigns blue or red to the dots in TSP corresponding to tunes whose c_{iu} and/or c_{iv} values are true. If both c_{iu} and c_{iv} values of a particular tune are true, TSP displays the dot corresponding to this tune as combination to two hemi-circles painted in blue and red.

Our current implementation limits the number of selected dots 5 or smaller.

Selection of acoustic features. Our implementation features GUI widget buttons to select two acoustic features to be assigned to horizontal and vertical axes of TSP. Users can freely and interactively observe the relationships between meta information selected in MISP and arbitrary pairs of acoustic features.

The visualization technique also features a method for automatic selection of the acoustic features for the axes of TSP. When a user selects an arbitrary set of meta information by the drag operation in MISP, the technique evaluates the visualization results of TSP for each pair of acoustic features, and automatically applies the pair of acoustic features which brings the best visualization result. Here, we suppose it is more meaningful if colored dots concentrate at the particular portions in TSP, because such visualization results bring clearer knowledge on relationships between meta information and acoustic features. Based on this supposition, the technique calculates the entropy of the colored dots in the display space. The technique divides the display space into l_D subspaces, and count the number of the dots which a specific combination of colors are assigned. It then calculate the entropy of the dots $H = -\sum_i^{l_D} p_i$, where p_i denotes the ratio of the number of the dots in the i-th subspace. Here, we need to calculate this entropy H for each combination of the assigned colors. If MISP assigns blue and red to dots, we calculate the entropy for blue, red, and blue+red dots. In other words, we calculate the entropy for $2^{l_C}-1$ times, if the number of assigned colors is l_C. The technique calculates the sum of the entropy $sumH = \sum^{2^{l_C}-1} H$ for each pair of acoustic features, and finally applies the pair of acoustic features which bring the smallest $sumH$ value.

4. EXAMPLES

This section introduces our experiment using the presented visualization technique. We applied 100 Japanese pop tunes including 5 tunes for each of 20 musicians.

Figure 2 shows a snapshot of our implementation. MISP displays 29 dots corresponding to 20 musicians and 9 chord progression patterns in this example. When a user drags the cursor on MISP, the implementation assigns colors to the dots which are close to the trajectory of the drag operation. Simultaneously, TSP colors the dots corresponding to the meta information (musicians or chord progression patterns) dragged on TISP. Thanks to this mechanism, users can interactively select the set of interested (and well correlated) meta information, and visually observe the relationship between the selected meta information and acoustic features.

Figure 3 shows a close up view of MISP displaying the 29 dots. Here we could observe several reasonable correlations indicated as (a), (b), and (c). Figure 3(a) depicts that the musician Tetsuya Komuro often uses the chord progression patterns “3” and “6”. Actually the chord progression pattern “3” is famously called “Komuro chord progression” by Japanese pop fans. Figure 3(b) depicts that the musicians Yumi Matsutoya, Kazumasa Oda, and Aiko commonly used similar chord progression patterns. We later found that these musicians actually used many of the patterns shown in Table 1. Figure 3(c) depicts that the mu-

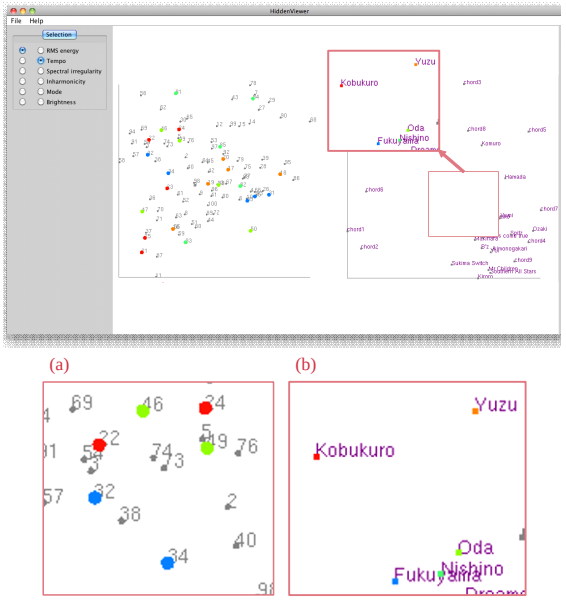


Figure 2. Snapshot of out implementation. When a user drags the cursor on MISP, the dots close to the trajectory of the drag operation are colored. Also, the dots in TSP corresponding to the meta information colored in MISP are also colored. (a) shows the colored dots in TSP. (b) shows the interactively selected dots in MISP.

sicians Mr. Children and Southern All Stars use similar chord progression patterns. We later found that they often used patterns "1" and "2".

Figure 4 shows an interesting trend discovered during the interactive operations. Here, a user selected two dots corresponding to chord progression patterns "1" and "2" by a drag operation in MISP. These dots are colored in red and orange respectively, as indicated in Figure 4(b). At the same time, the dots corresponding to the tunes using the patterns "1" and "2" are also colored in red or orange in TSP. While dots colored in either red or orange are well scattered, dots colored in both red and orange are concentrated in the upper-left region in TSP, as indicated in Figure 4(a). We visually discovered an association rule that the tunes including both chord progression patterns "1" and "2" tend to have smaller RMS energy and larger Tempo values.

Figure 5 shows another trend discovered during the interactive operations. Here, a user selected four dots corresponding to a chord progression pattern "6", and three artist name (Kobukuro, Masaharu Fukuyama, and Sukima Switch) during the drag operation in MISP. These dots are colored in red, orange, bright green, and cyan, respectively, as indicated in Figure 5(b). At the same time, the dots corresponding to the tunes played by one of the above artists and used the chord progression pattern "6" are also colored in red or orange in TSP. While dots colored in red are well scattered, dots colored in both red and one of other colors are concentrated in the particular region in TSP, as indicated in Figure 5(a). We found that these three artists used the same chord progression pattern to their tunes which

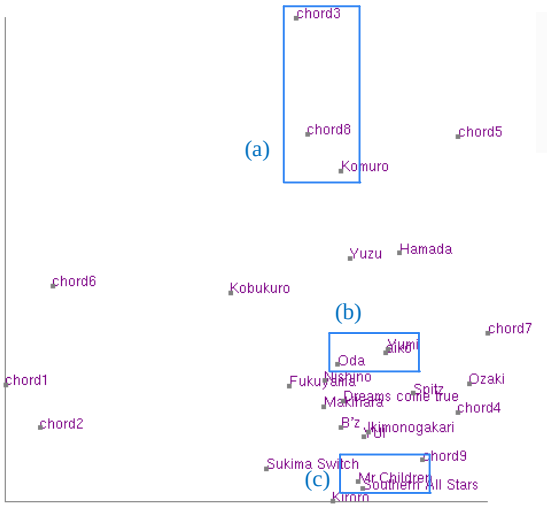


Figure 3. Close up view of MISP. We could observe several reasonable correlations among musicians or chord progression patterns.

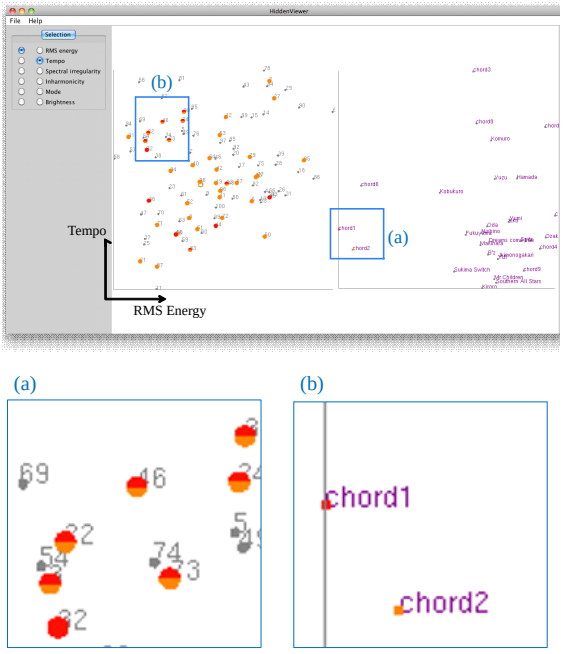


Figure 4. Example of an interesting trend. (a) Tunes which contain both the selected patterns concentrated in the upper-left region in TSP. (b) A user interactively selected two dots corresponding to chord progression patterns "1" and "2".

have similar acoustic features.

We would like to observe more associations between meta information and acoustic features using this visualization technique, and discuss what kinds of chord progressions and acoustic features are coupled while composing and arranging pop music.

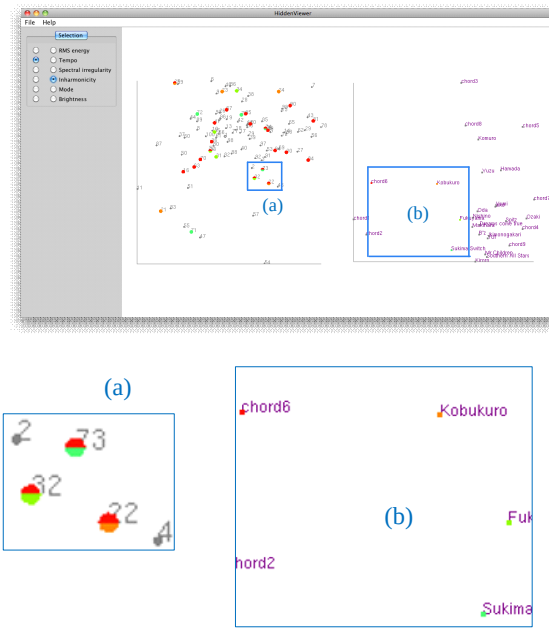


Figure 5. Example of an interesting trend. (a) Tunes which contain two of the selected meta information concentrated in the particular region in TSP. (b) A user interactively selected four dots corresponding to a chord progression pattern and three artist names.

5. CONCLUSIONS

This paper presented a visualization technique featuring dual scatterplots. One of the scatterplots (called TSP) represents the distribution of tunes with their acoustic features, while the other (called MISP) visualizes the correlations among meta information of the tunes. The technique provides an interaction mechanism to select interested set of meta information in MISP, and represent the distribution of the selected meta information in TSP. The paper introduced several examples of visualization results with Japanese pop songs to demonstrate the effectiveness of the presented technique.

The following are our potential future issues.

Listener preferences as meta information. Our current dataset only contains artist names and chord progression patterns as meta information. On the other hand, relationships between listeners' preferences and other meta information or acoustic features are also interesting and worth to be visualized. We would like to hear favorite tunes from experimental users of this technique, add the information to our current dataset, and observe the visualization result again.

Extension of chord progression pattern extraction. Our current implementation on chord progression pattern matching is too naive. There are many chord progressions which are seemingly different but theoretically similar; however, our current implementation does not recognize such patterns. We would like to extend the implementation to extract patterns more flexibly. Also, we would like to extract usage of tensions. It is often observed that specific tensions are used by specific composers or specific genre of tunes. It is also an important factor to discover the characteristics of tunes or artists.

Improvement of visual representation. Our current implementation of scatterplots just assigns colors to meta information in the order of the drag operations. In other words, the coloring mechanism does not have particular semantics. We would like to improve the mechanism so that users can understand the relationships between the selected meta information more intuitively. Also, we would like to test with other dimensionality reduction schemes to MISP. Our current implementation just applies a classical MDS. We observed inconsistent layout results, where dots corresponding to correlated meta information are distantly placed while dots corresponding to less correlated meta information are closely placed. We expect other dimensionality reduction schemes will improve the results.

Scalability test. Our current dataset is too small and therefore the examples shown in this paper does not demonstrate the scalability of the presented technique. Also, correlations among meta information in the examples are not reliable because our dataset only contains 5 tunes for each artist. We would like to extend the dataset and test the scalability of the visualization technique.

Acknowledgments

We appreciate J-Total Music for their approval of our usage of the chord progression published on the Web for our academic research purpose.

6. REFERENCES

[1] P. Ciuha, B. Klemenc, D. Solina, Visualization of Concurrent Tones in Music with Colours, ACM International Conference on Multimedia, pp. 1677-1680, 2010.

[2] J. Donaldson, P. Lamere, Using Visualization for Music Discovery, International Symposium on Music Information Retrieval, 2009.

[3] M. Goto, T. Goto, Musicream: New Music Playback Interface for Streaming, Sticking, Sorting, and Recalling Musical Pieces, 6th International Society for Music Information Retrieval, pp. 404-411, 2005.

[4] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, T. Nakano, Songle: A Web Service for Active Music Listening Improved by User Contributions, 12th International Society for Music Information Retrieval, pp. 311-316, 2011.

[5] A. Hayashi, T. Itoh, M. Matsubara, Colorscore - Visualization and Condensation of Structure of Classical Music, Knowledge Visualization Currents: from Text to Art to Culture, Springer Edit Volume, ISBN-978-1-4471-4302-4, 2012.

[6] J. Holm, Visualizing Music Collections Based on Metadata: Concepts, User Studies and Design Implications, Tampere University of Technology, 2012.

[7] K. Kusama, T. Itoh, Abstract Picture Generation and Zooming User Interface for Intuitive Music Browsing, Springer Multimedia Tools and Applications, Vol. 73, No. 2, pp. 995-1010, 2014.

[8] S. Leitich, M. Topf, Globe of Music - Music Library Visualization Using GeoSOM, International Conference on Music Information Retrieval, pp. 167-170, 2007.

[9] E. Pampalk, Islands of Music: Analysis, Organization, and Visualization of Music Archives, Master Thesis, Vienna University of Technology, 2001.

[10] E, Pampalk, M. Goto, Musicrainbow: A New User Interface to Discover Artists Using Audio-based Similarity and Web-based Labeling, International Conference on Music Information Retrieval, pp. 367-770, 2006.

[11] Y. Saito, T. Itoh, MusiCube: A Visual Music Recommendation System featuring Interactive Evolutionary Computing, Visual Information Communication and Interaction Symposium (VINCI'11), 2011.

[12] H.-H. Wu, J. P. Bello, Audio-Based Music Visualization for Music Structure Analysis, Sount and Music Computing, 2010.

[13] J. Zhu, Perceptual Visualization of a Music Collection, IEEE International Conference on Multimedia and Expo, 1058-1061, 2005.

[14] O. Lartillot, MIRtoolbox, available from http://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox

[15] J-Total music, http://music.j-total.net/index.html

BEAN: A DIGITAL MUSICAL INSTRUMENT FOR USE IN MUSIC THERAPY

Nicholas J. Kirwan
Aalborg University Copenhagen
nkirwa13@student.aau.dk

Dan Overholt
Aalborg University Copenhagen
dano@create.aau.dk

Cumhur Erkut
Aalborg University Copenhagen
cer@create.aau.dk

ABSTRACT

The use of interactive technology in music therapy is rapidly growing. The flexibility afforded by the use of these technologies in music therapy is substantial. We present steps in development of Bean, a Digital Musical Instrument wrapped around a commercial game console controller and designed for use in a music therapy setting. Bean is controlled by gestures, and has both physical and virtual segments. The physical user interaction is minimalistic, consisting of the spatial movement of the instrument, along with two push buttons. Also, some visual aspects have been integrated in Bean. Sound synthesis currently consists of amplitude and frequency modulation and effects, with a clear separation of melody and harmony. Bean is being co-developed with clients and therapists, in order to assess the current state of development, and provide clues for optimal improvement going forward.

1. INTRODUCTION

A basic working definition of music therapy is the use of music as a tool, in a therapeutic setting. Tailored to the individual needs of the client, this tool can be used to achieve therapeutic goals such as enabling communication or improving motor skills [1]. The flexible nature of a Digital Musical Instrument's (DMI) sonic output and control possibilities could be a powerful tool to add to the arsenal of a music therapist. Indeed it has been shown that the use of electronic musical technologies has an impact on outcomes relating to communication and expression [2], while also enabling a sense of achievement and empowerment [3]. As mentioned, communication is a common goal in this form of therapy. Facilitating performance and ancillary gestures through tangible interaction, could therefore lead to expressive communication when combined with music [4].

For some clients the "up to date technology" itself can be a positive and engaging factor in music therapy, in addition to the possibility for new and interesting sounds or "new sound worlds" [5]. The use of novel technologies in music therapy can however pose some practical, as well as design problems. For instance, can clients easily understand the musical contribution is of their making? Is the control of these contributions intuitive and understandable? Is the experience of using these technologies engaging, with enough variance to hold interest? These issues are not directly related to music therapy, but are in fact universal factors associated with DMI design, for example the "ubiquitous mapping problem" [6]. While the term music therapy is too general, and may cover physical, cognitive, learning, and rehabilitation goals, as well as different target groups, effective utilization of these factors could possibly be tried out in participatory design settings. Participation in design is of greater importance when the user has complex needs.

In this paper, we present the iterative development of Bean, a novel *visual, aural, and tangible* DMI. Bean was created to investigate problems like the above-mentioned, and to help provide some answers. After outlining the background research relevant to the design, the design and construction process of Bean is elaborated on. Next an initial participatory design session is described, followed by a discussion. We conclude with the future plans for the development of Bean.

2. BACKGROUND & RELATED WORK

The use of technology in music therapy has the potential for many positive applications, but the therapist must have the required knowledge to effectively use these technologies in a therapeutic setting [7]. Previous research has investigated technology use in music therapy [7][8]. While these studies cannot be directly used as design requirements for new musical instruments, they provide some starting points. For example, they make clear that distance sensing is the most frequently used sensing mode.

Tangible interface use is not as widespread in music therapy; a percentage of clients would have physical disabilities that could hinder such interaction. Despite a lack of total inclusivity, there is still a need for the option of tangible interaction for those clients with this ability, to ideally enable an embodied musical experience. Tangible DMIs could reveal the conceptual metaphors of the clients, address their tactile/kinesthetic hyposensitivity, and act as diagnostic and performance tools to gauge their capabilities.

Copyright: © 2015 First author et al. 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.

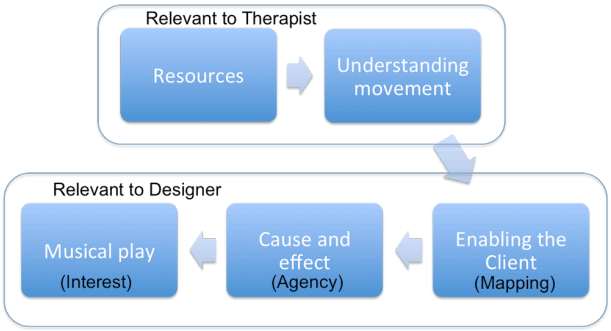


Figure 1. Framework for technology use in music therapy, adapted from [2].

A framework has been previously suggested through an investigation of music therapists' experience with technology use [2] (see Figure 1). The data gathered here can in part, be used to effectively design technologies that suit this setting. The first two points are aimed more at informing the therapist, and have little relevance to the design of instruments. The three last points can be intrinsically linked to the functionality and design of DMIs such as Bean. These elements in the context of DMI design would however be more intuitive in the following order: Cause/effect and a sense of agency is a primary element. After this, comes enabling the client through effective mapping, which should lead to musical play that holds the interest of the user.

2.1 Cause and effect: agency

Cause and effect are interlinked with agency. Paine & Drummond [9] categorize agency into two approaches in relation to DMI design: 1) the control of predetermined sequences of sounds such as triggering sounds in sample based software, and 2) the *creation* of sound through real-time manipulation of software synthesis variables. Furthermore, when the *creation* paradigm is designed for, it is suggested immediate agency should be facilitated accounting for primary causality in the use of the DMI. Immediate agency and corresponding feedback could be seen as modeling the cause and effect cycle.

2.2 Enabling the client: mapping

Magee & Burlan [2] take a practical view to enabling the client, with mention of switch or sensor placement in relation to the client's difficulties that is very similar to the aforementioned understanding movement element of the process. The focus is mostly on physical impairments. Mapping is nonetheless also rudimentarily mentioned.

The importance of mapping has been investigated [10]. It largely defines the user interaction and experience [6]. An effective mapping strategy would enable the client to effectively interact with the musical content. A client with complex needs might benefit also from a *transparent mapping* strategy, which could be complemented by cross-modal feedback such as visual cues similar to those discussed in [11, p52]. The use of transparency in this context can be defined as an easily understandable connection from action to audible change.

2.3 Musical Play: sustained interest

Playing music is inherent in music therapy, but the quality aesthetically, is secondary to the effectiveness of the use of music as a tool to achieve a goal. Sound design and the aural feedback framework are, along with mapping, central to this topic. The effectiveness of the sound design and amount of control over these sounds can have an influence on the amount of time a client is willing to spend playing the instrument. Effective integration between these aspects could lead to sustained play. It is not necessarily the quality of musical content, but rather the sustained interest in the content, which in turn provides a tool to possibly facilitate communication and expression in a therapeutic setting.

2.4 Related Commercial Applications

According to a survey investigating current technology use in music therapy, including over 600 therapists [8], Soundbeam[1] is the most popular system of interactive technology in use in music therapy, followed by MIDIcreator.[2] Both of these systems are directed more towards physical impairments, and the first one lacks an option for tangible, embodied interaction. As regards tangible interfaces for musical novices, notable commercial examples include the Skoog[3], which was produced with an aim towards inclusion of those with special needs, and the open-source, Teensy-based Kyub[4].

3. BEAN

Bean is a gesturally controlled digital musical instrument. It is ellipsoidal in shape, which innately fits well between two hands. The user interaction is minimalistic, consisting of the rotational movement of the instrument, along with two push buttons. The instrument is played by a combination of these two modes of interaction. Some combinations happen naturally through gestures. This could be described as an extra mapping layer [10]. The block diagram and various stages of the Bean's design are illustrated on Fig 2 top and bottom, respectively.

3.1 Musical Interaction Design

The simplicity of Bean is an intentional design feature, to provide a safe and durable entry point for two-handed interaction and transfer back and forth to other clients or therapists. Although primarily a musical instrument, there are also some visual aspects integrated in Bean. All aspects can be easily extended, augmented, or redesigned within or after participatory sessions.

[1] http://soundbeam.co.uk
[2] http://www.midicreator-resources.co.uk/
[3] http://www.skoogmusic.com
[4] http://kyubmusic.com/

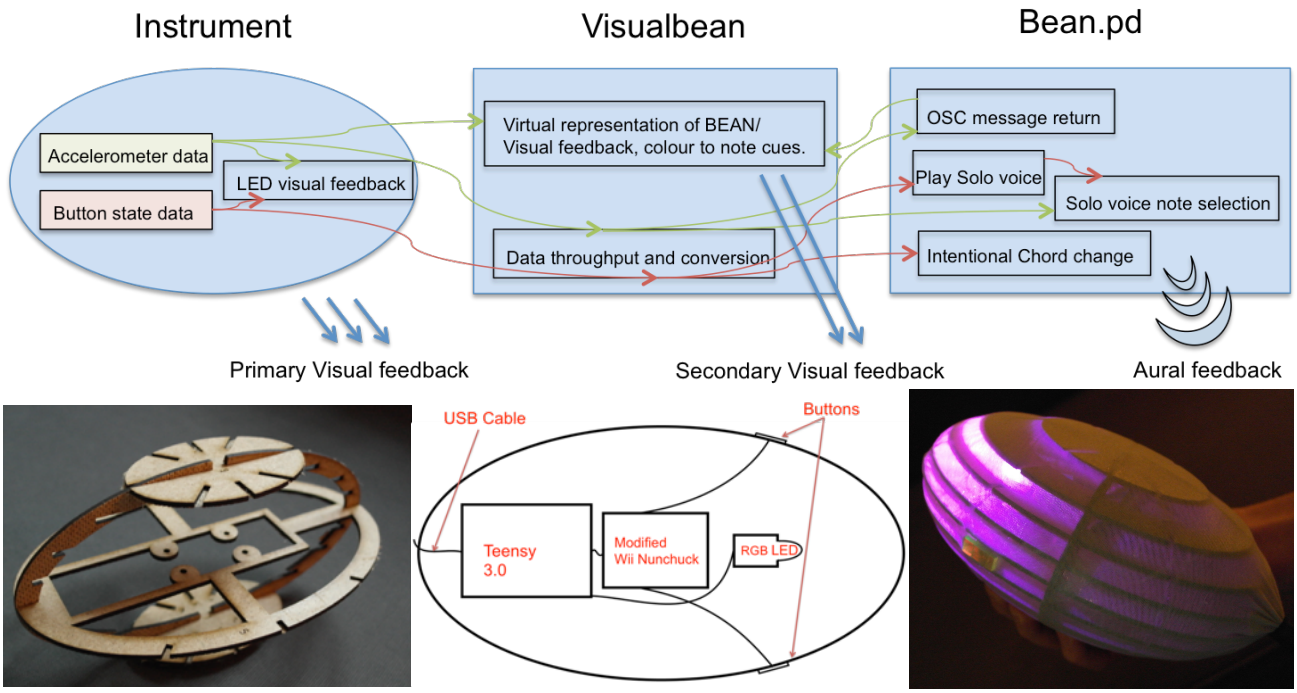


Figure 2. (Top) A data flow diagram showing the sensor data and control paths. (Bottom) Various stages of design.

3.1.1 Sonic feedback

The concept behind the current implementation of aural feedback is that of harmonic backing chords, which shift autonomously. This harmony provides a musical setting, a starting point. Over this the client has the opportunity to improvise using a solo voice, which is governed by certain rules to enable the client to easily find notes that fit with these chords

The harmonic content of the chords is noncomplex in nature. The four chords are Cmaj9, Dmin9, Emin7 and Fmaj9: all the elements of the C major pentatonic scale fit with these chords. For this reason, the notes of the C pentatonic scale in two octaves are used for the solo voice element of the aural feedback. There is also another group of tones made available to the user when the instrument is shaken briefly. These notes constitute an A blues scale. This new state lasts for 30 seconds, providing and option for tonal variance and possible dissonance in the solo, before the pentatonic tone mode is re-engaged.

Aural feedback was implemented using Pure Data. Bean.pd is the main hub where the sensor data is received and formatted. Open Sound Protocol (OSC) is used to transmit the sensor data into this patch. Formatting, in this context, can be understood in this way; the accelerometer data and the current state of both buttons are transformed into data usable by the synthesizers and control elements, e.g. accelerometer roll data is received as numbers between 70-170 then scaled to a number between 0-1. There are also OSC control messages broadcast from Bean.pd. These messages are composed using the sub-patch OSCreturn.pd, and have the purpose of controlling certain aspects of the visual feedback.

The method of sound creation is a combination of additive synthesis and frequency modulation synthesis. The additive synthesis comprises of a fundamental and three partials. These partials are individually adjusted in amplitude to provide an element of timbre change. Frequency modulation is used to add complexity to the aural content of the users' solo. An ADSR envelope and a phaser complements the solo instrument.

Another sub-patch creates the accompanying harmonies with a bank of five additive synthesizers, one for each note in the harmony. Each of these synthesizers in turn composes a tone, constructed of a fundamental and three partials. The four chords change randomly over time with equally weighted probability for each. In the current implementation there is also an additional option for the user to intentionally change the accompanying chord.

3.1.2 Mapping

The mapping strategy for Bean is generally one-to-one mapping. The selection of note in the solo voice is the most discernable change aurally. This change is mapped to the pitch angle of the instrument (Fig. 3). When the instrument is swiveled downwards on the X-axis, the pitches fall, and conversely when the instrument is swiveled upwards on this axis, the pitches rise. Measurable movement range is divided into ten to facilitate the available notes.

Change in roll is mapped to the aforementioned tonal variation. Rotational movement on the Y-axis to the left effectively gives a more bass rich sound. This movement is mapped to a reduction in amplitude of the upper partials of the additive synthesis component in the solo voice. The opposite gesture, rotation to the right, produces the effect of a strong higher frequency element to the sound. This is achieved by increasing the amplitude of the three upper partials. This increase is staggered from low to high in order to give a smooth timbral alteration. It is envisaged that these movements, swivel up/down and rotate

left/right, will become elements of dynamic gestures by the user. The *jolt* gesture in the Z-axis is currently functioning as a trigger, which when activated, switches the scale from C pentatonic to the A blues scale. This change of scale automatically resets after 30 seconds if not re-triggered.

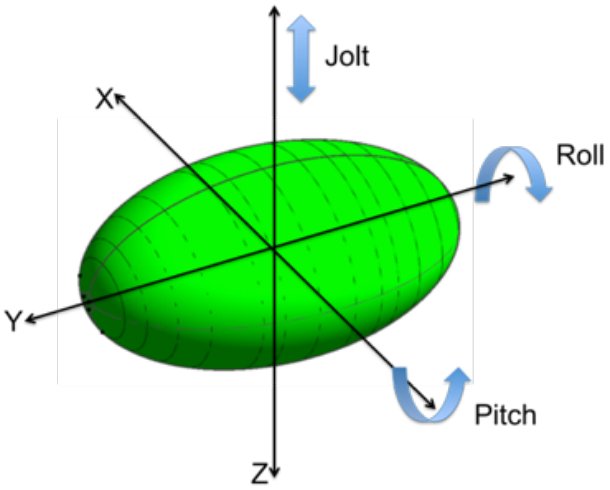


Figure 3. The accelerometer data used in mapping.

The two buttons also have the possibility to have a major effect on the aural feedback. The button situated on the right of the instrument, is assigned as a *play* button. When this button is pressed, the attack, decay and sustain part of the envelope is engaged and the solo voice plays. When the button is released the release part of the envelope engages to taper off the amplitude. This is an intrinsic element in every instrument, the initiation of sound. Rather than a higher level continuous *control* model where the user would interact with a pre existing sound framework, Bean is designed with the *creative* model, as discussed in [9].

3.2 Physical Design and Implementation

Bean's ellipsoidal has been modeled in 3D, segmented and laser cut from a press-fit format. The internal hardware was securely attached of. Several iterations where cut, during a fine-tuning process for both fit and size. The material used to manufacture the press-fit skeleton was 3mm hardboard. Corel Draw and the laser cutter were also used to cut the button tops from 3mm acrylic sheet material. These additions were needed to increase the surface of the pressable area on each of the buttons. Finally, the outer surface covering consists of layers of PVC foil, covered by a double layer of nylon from a pair of stockings. This covering has a dual purpose. The first restricts access to the internal hardware by enclosing the skeletal frame. The second is partly cosmetic, to diffuse the internal light source and make Bean pleasing to the eye.

3.2.1 Hardware

Embedded computing is at the heart of Bean (Fig. 2). Teensy 3.0[5], a compact Arduino[6] compatible USB microcontroller, is the "brain" of the physical segment of the instrument i.e. the ellipsoid. The Teensy board powers up and initiates communication with the Wii Nunchuck[7] board. It then receives all the sensor data, turns the relevant data into direct visual feedback, and also transmits all the data further over serial communication to the computer. For ease of connection the Teensy was mounted on a custom made circuit board, which allowed for effective connection and disconnection with both the Nunchuck board and the LED.

The sensor unit is in fact a modified Wii Nunchuck, to enable the original buttons to be extended away from the body of the Nunchuck, and to be placed on the outer shell of the instrument. The main sensor is an on-board accelerometer from the Nunchuck. This sensor enables movement tracking in both the pitch (X-axis) and roll (Y-axis), and *jolt* detection vertically (Z-axis). The two buttons allow extra access to control parameters.

3.2.2 Sensor input

The first step was to program the Teensy microcontroller. An Arduino sketch was created that enables the Teensy to initialize the Wii Nunchuck, by using the I2C[8] communication protocol, and begin receiving the sensor data. The LED is also initialized with this sketch, and is communicated with, by the use of the SPI- communication protocol. The sketch also directly maps certain sensor data to different colours produced by the LED. The final step is the formatting and transmission of the sensor data over the serial bus to the laptop.

3.2.3 Visual feedback

The LED installed inside the physical element of the instrument provides primary visual feedback. This feedback is mirrored in a secondary visual feedback, which is a 3D virtual representation of Bean. Colour to musical note mapping was implemented to provide a form of visual cueing. To facilitate this virtual representation, the application *VisualBean* was created on Processing[10], but will be not discussed here. However, the colour to tone mapping will be discussed only on the physical part of the instrument here. The equal temperament frequencies of the selected notes were transposed and superimposed from the audible range to the visual frequency range; chromesthesia [13] could have been an alternative way.

4. PARTICIPATORY DESIGN AND EVALUATION

Participatory design and evaluation has been done in two sessions over two days.

[5] https://www.pjrc.com/teensy/index.html
[6] http://arduino.cc/
[7] The Wii Nunchuck is a controller for the Wii game console.
[8] http://www.i2c-bus.org/
[10] http://www.processing.org

4.1 Session 1: Clients

Two service users (Participant A and B) of an adult training center, along with a member of staff, agreed to participate in an informal evaluation and participatory design session. Both clients are male, were in their early twenties and have mild/borderline intellectual disabilities. They both had no formal music training, but both have had music therapy sessions in the past. The setting was informal, not therapeutic in nature. Nevertheless, this was a valuable opportunity to initially assess the instrument with a prospective target group, with a view to gathering information for further development.

The session took between 30-35 minutes. Both participants were in the room simultaneously. The format of the meeting took the following form: The first 20 minutes were spent with the two participants taking turns in free play with the instrument, without any instruction. After this, there was a short discussion about the device, to gauge the participants' impressions, and level of understanding. The session then continued, with the participants and the staff member engaged in more free play turn taking. The prototype used in the session was an earlier, less developed iteration. There was no outer covering on the prototype and there was also no internal LED.

4.1.1 Free play

Participant A was initially hesitant in using Bean. His interaction was exploratory, starting with just moving the instrument in space, registering that the representation on screen was mirroring the physical movements. Shortly after, the buttons were pressed, with resulting surprise when the solo voice engaged.

Participant B was more direct in use, engaging the play button immediately. This was to be expected, as he could see the first participant's use of the device. His gestures were slow and deliberate at the start, but quickly changed to moving the device more aggressively.

4.1.2 User impressions

An open discussion followed the free play. Semi-structured questions included: What are your first impressions? Did you understand the control functionality? Was it interesting to use? How would you change/improve it? First impressions of Bean were that it was different, but fun. Whether this fun factor was because the technology is new, or the fact that making music was facilitated in a new way, was unclear. They were both nevertheless eager to try the interface again.

Both participants understood that movement affected the sound, and that the *play* button had to be pressed to solo. The *change chord* button however was a mystery. Participant B triggered the *jolt* that controlled A blues scale; the participants did not realize the change in scale.

Both participants found Bean interesting to use. When they were asked in connection to interest, if they could see themselves using the instrument for a sustained time, they both answered yes. As with the first question it is unclear if the opportunity to play music, or the opportunity to play with new technology was the deciding factor. As for the improvement, both participants agreed that a cover for the surface of the device would be a good idea. Participant A also felt that the device could be used for other purposes, relating to computer control. The member of staff was also of the opinion that the device was very flexible and could be used for other purposes.

4.1.3 Free play continued

After the discussion, the participants got more play time. During both of these free play sessions contrasting styles of use could be observed. Participant A continued with a more methodical style, actively searching certain notes and evaluating the sound changes. In contrast participant B was more interested in moving the device as fast as possible, not as selective with which notes he played, but rather getting fast runs up and down the scales. The movement of the virtual representation was possibly of more interest than the sound of the instrument for this participant. The member of staff helping in the evaluation also played the instrument at this time. He put forward the opinion that the device could be very beneficial in a group music therapy setting.

4.2 Session 2: Therapists

There was also an opportunity to talk to a practicing music therapist and an art therapist. The music therapist is an experienced musician and uses an improvisational approach to music therapy. He has some experience with the use of Soundbeam, but aside from that limited experience of technology use in therapy. The art therapist also had limited experience with tangible technologies in therapy.

Both of the therapists played the Bean. The music therapist was the first to use the interface, and immediately wanted more methods of control. On the top of Bean where his thumbs naturally rested in use, could be an optional placement for more buttons, he suggested. Also he felt that the aural feedback lacked a rhythmic element or a "beat". After considering the device's current state, he felt that the prototype could be easily destroyed by some of his users. If they for instance became frustrated the gaps in the outer structure were finger sized, providing a grip to pull the device apart.

The art therapist was positive about the applications a device like Bean could have in an art therapeutic setting, if the visual feedback was more flexible, to perhaps enable drawing. In effect translating the visual cue based feedback currently implemented, into a more visually creative virtual canvas.

5. DISCUSSION

Valuable information was gathered in the sessions. Observations of the two participants' free play sessions suggested two potential paths of development: *refine the musical control* and *promote the kinetic aspects* of the instrument.

The implementation of extra control options, also mentioned by the music therapist, would have both

positive and negative consequences: The balance of control options and usability must be carefully maintained. Users with complex needs could possibly have trouble conceptually managing more control options. With the minimalistic style of Bean comes the risk of a lack of control content to maintain interest. This was not evident in Session 1 (Sec. 4.1). Both participants seemed to be engaged while using the device. A larger scale, formal evaluation would be needed to give more conclusive results to this problem, the initial results are nonetheless promising.

During the sessions, the fact that Bean was a new device using up to date technologies, was clearly a positive influence. The participants were interested, and one could even say motivated by that fact alone, before interaction even took place. This adds weight to a claim that more technology use in music therapy could have positive effects, at least relating to a young male demographic, similar to our participants, and possibly not exclusively to this demographic.

The rhythmic element suggestion mentioned by the therapist is an interesting one. In some forms of contemporary music the "Beat" could be seen as being of more importance than harmonic content. This suggestion is certainly food for thought going forward, and outlines a possible deficiency in the current musical content of Bean. Visual interactivity changes as proposed by the art therapist, were interesting and undoubtedly an avenue of development for a broader base of therapy options.

6. CONCLUSIONS AND FUTURE WORK

This paper has outlined the design and development of a digital musical instrument, Bean, which is primarily being designed for use as a novel tool in the arsenal of the music therapist. Research pertaining to the fields of music therapy practice, DMI/NIME design and human computer interaction has guided the process. An initial informal evaluation of a functioning prototype by a possible target group and professionals in the field has proved to be informative for the further development of Bean.

Much work is still needed on some aspects of the system, but there is a firm foundation to work further from here. The developments carried out since this evaluation have improved the device structurally, and the hope is that the instrument now has better playability after visual cueing has been introduced. Some aspects of the mapping strategy will also be reviewed, such as the *change chord* option. This could possibly be changed to an option, which would allow extended range, similar to some small MIDI keyboard controllers offer.

To provide more flexibility in sound choice, and a familiar protocol the music therapists, MIDI messaging could be implemented. The proliferation of MIDI device use in music therapy would suggest that it would be preferable to have some MIDI functionality integrated in the system. The Bean.pd patch could be developed further to facilitate flexibility with regards MIDI communication.

There are plans to replicate the Bean system, in order to enable musically collaborative therapeutic group work. A larger scale, formal evaluation would however be a next step, to possibly get empirical data, informing on how Bean would perform in a therapeutic setting, We could, for instance, implement two different mappings (the current one plus a more percussive-like mapping - using the accelerometer to trigger notes with varying velocities similar to the Kyub), and use both empirical data and user experience feedback to compare and contrast the different modes/playing styles.

7. ACKNOWLEDGMENTS

Many thanks go to the service users and staff members from Cope foundation for facilitating and participating in this evaluation. Also, thanks to both therapists Eoin Nash and Ed Kuczaj, who generously offered their professional opinions on Bean.

8. REFERENCES

[1] K. E. Bruscia, *Defining Music Therapy*. Barcelona Publishers, 1998.

[2] W. L. Magee and K. Burland, "An Exploratory Study of the Use of Electronic Music Technologies in Clinical Music Therapy," *Nord. J. Music Ther.*, vol. 17, no. 2, pp. 124–141, Jul. 2008.

[3] K. Burland and W. Magee, "Developing identities using music technology in therapeutic settings," *Psychol. Music*, vol. 42, no. 2, pp. 177–189, Nov. 2014.

[4] M. Wanderley and B. Vines, "The musical significance of clarinetists' ancillary gestures: an exploration of the field," *J. New Music Research*, 2005.

[5] A. Hunt, R. Kirk, and M. Neighbour, "Multiple media interfaces for music therapy," *IEEE Multimedia* 11, 3 (July, 2004), 50–58.

[6] J.Malloch and M. Wanderley, "The T-Stick: From musical interface to musical instrument,". *Proc. NIME07*, New York City, USA, 2007.

[7] B. Farrimond, D. Gillard, D. Bott, and D. Lonie, "Engagement with Technology in Special Educational & Disabled Music Settings," *Youth Music*, 2011.

[8] N. D. Hahna, S. Hadley, V. H. Miller, and M. Bonaventura, "Music technology usage in music therapy: A survey of practice," *Arts Psychother.*, 39, 5, (Nov. 2012), pp. 456–464.

[9] G. Paine and J. Drummond, "Developing an Ontology of New Interfaces for Realtime Electronic Music Performance," *Electroacoust. Music Stud.*, 2009

[10] A. Hunt, M. M. Wanderley, and M. Paradis, "The Importance of Parameter Mapping in Electronic Instrument Design," *J. New Music Res.*, 32, 4, (Dec. 2003), pp. 429–440.

[11] S. Fels and M. Lyons, "NIME 2011 Tutorial: NIME Primer."

[12] P. Wyeth, "Agency, tangible technology and young children," *IDC '07 Proc. Intl. Conf. Interact. Des. Child.*, pp. 101–104, 2007.

[13] G. Rogers, "Four cases of pitch-specific chromesthesia in trained musicians with absolute pitch," *Psychol. Music*, 1987.

Music Synthesis based on Impression and Emotion of Input Narratives

Saya Kanno
Ochanomizu University
saya@itolab.is.ocha.ac.jp

Takayuki Itoh
Ochanomizu University
itot@is.ocha.ac.jp

Hiroya Takamura
Tokyo Institute of Technology
takamura@pi.titech.ac.jp

ABSTRACT

This paper presents a technique to synthesize the music based on the impression and emotion of the input narratives. The technique prepares a dictionary which records the sensibility polarity values of arbitrary words. The technique also supposes that users listen to the sample chords and rhythms, and input the fitness values to the pre-defined impression word pairs, to learn the relations between features of chords/rhythms and these impression. After these processes, the technique interactively synthesizes the music for input narratives. It estimates the fitness values of an input narrative to the impression word pairs using the dictionary, and then selects the chord and rhythm progressions those impressions and emotions are the closest to the narrative. Finally, the technique synthesizes the output tune by combining the chord and rhythm. We suppose this technique encourages to express impression and emotion of the input narratives by generating music.

1. INTRODUCTION

We may want to express the impression and emotion of narratives by creating another media. We sometimes write our sentiments as review documents, or illustrate the scenes of the novels. Here, we may provide too much information to the readers while writing reviews even though if they do not want to know the contents and details before reading the narratives. Or, we may provide unnecessary or inadequate impression from the illustrations. Our motivation of this study is to express the impression and emotion of narratives applying music.

This paper presents a technique to express the impression and emotion of narratives by synthesizing music. Music has effects to derive imagination, behavior, and emotion to the listeners. These are effective to every people, whether the listeners are deeply interested in the music or not [15]. We expect such effects are helpful to feel the impression and emotion of input narratives, without knowing too much information. Moreover, we have been already familiar with multi-modal arts which combines music and other expression, such as Opera and Ballet. Therefore, we expect that people may feel new types of sensation and inspiration by listening to the music created for the narratives while the people are reading them.

Copyright: ©2015 Saya Kanno 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.

We had a questionnaire "Would you like to listen to the music based on the impression and emotion of the narratives after you read that narratives?" performed on the Web, and received answers from 57 participants. As a result, 24 participants answered "Strongly want to listen", and 26 of others answered "Rather want to listen". This result suggests that 88% of the participants are interested in listening to the music based on the impression of the narratives. We expect the music generated based on the impression of the input narratives is effective to stimulate interest of users. We can imagine the abstract impression of the narrative before reading by listening to the music generated by the presented system. Or, we can play the background music that matches the impression of the narratives which we are reading them. Moreover, we expect this technique is useful to publish and intercommunicate the impression of literature by uploading the music. We also expect the uploaded tunes bring interest and imagination to the narratives.

2. RELATED WORK

There has been a long history of studies on automatic music generation with different media including images and narratives. Many of the studies generate music based on high-level semantics such as structures of stories or scenes, while many other studies are based on abstract impression.

Background music generation for slideshows, movies, computer animations, or real spaces is a recent active research issue. Some of the studies are based on high-level semantics [3] [11] while some of others are based on abstract impression [1] [12] [14]. The study presented in this paper is closer to the latter type of studies.

There have been smaller number of studies on music synthesis adapting to narratives applying natural language processing techniques. Endo et al. [4] presented a technique to generate music based on arguments and grammatical structures of input narratives. This technique is not based on the impression and emotion of the narratives. Kitahara et al. [10] presented a technique to automatically compose music by a note sequence generation from the impression and emotion of the input narratives. We subjectively suppose this approach is not always suitable to generate user-preferable music, because it just automatically generates sequences of notes without learning the impression of listeners and utilizing the user-preferred music patterns. Cruz et al. [2] also presented a technique to generate music based on the emotion of input narratives. Again, this technique does not adopt preferences or impression of users to the generation of music. On the other hand, our technique

is based on the mash up of user-prepared chord and rhythm progressions, and learning of the impression and emotion of the chords and rhythms. This approach can take into account the impression of the listeners to the music, and utilize the user-prepared musical patterns.

Ishizuka et al. [9] presented a theme music arrangement system based on impression of story scenes. Its architecture is close to our study since it arranges input theme tunes based on the numeric impression values. However, it does not deeply discuss how to calculate impression values and learn users' own impressions. Also, this technique does not apply user-prepared musical patterns.

3. PRESENTED TECHNIQUE

This section describes the processing flow and implementation detail of the presented technique. The technique consists of the following three technical components:

1. **Preliminary data construction:** Selection of impression words and musical features, and dictionary construction applying a semantic orientation calculation technique [16]. These steps are applied once by the system developer.

2. **Learning:** Calculation of coherency between the musical features and these impressions. This step is applied once for each user as a preprocessing.

3. **Interactive process:** Music synthesis for input narratives.

Here, we prepared the chords and rhythms by reference to the commonly used patterns in pops, rock, jazz, and classical music. Our current implementation supposes BPM (Beat per Minute) and number of measures of all the chords and rhythms are equal.

Our current implementation generates music by just synthesizing chords and rhythms. It is our on-going work to implement the procedure to synthesize melodies to generate the music, since the presented mechanism to select chords and rhythms can be similarly applied to melodies.

The below subsections describe the processing flow and implementation detail of the each technical component.

3.1 Preliminary data construction

Our preliminary data construction phase includes the following two processes: 1) selection of impression word pairs and musical features, and 2) dictionary construction. Our current implementation constructs Japanese dictionary; however, the presented mechanism is not limited to specific natural languages.

3.1.1 Selection of impression word pairs and musical features

This step firstly selects impression word pairs used for semantic orientation calculation, and musical features used for the selection of chords and rhythms. We listed the impression word pairs and musical features shown in Table 1 as the candidates. We learned impression word pairs from Ikezoe et al. [8], and musical features from Hasegawa et al. [6]. Also, we subjectively added frequency of several chords, ratio of inharmonic tones, and frequency of tones for each of drums, because they are often effective to express the particular mood or emotion.

Table 1. Candidates of impression word pairs and musical features.

Impression word pairs for chord progression
Bright - Dark
Light - Heavy
Enjoyable - Wistful
Brassy - Simple
Tripping - Quiet
Energetic - Calm
Musical features for chord progression
Average of tones
Distribution of tones
Number of simultaneous tones
Ratio of inharmonic tones
Frequency of major, minor, seventh, major seventh, and minor seventh chords
Impression word pairs for rhythm progression
Fast - Slow
Light - Heavy
Quiet - Loud
Brassy - Simple
Energetic - Calm
Musical features for rhythm progression
Frequency of tones for each of drums
Total number of tones
Frequency of 16-, 8-, and 4-beat notes
Frequency of triplets

Then, we conducted a questionnaire for the selection of impression word pairs and musical features. We asked participants to listen to the sample chord and rhythm progressions, and answer the subjective fitness between the samples and impression word pairs in the five-point Likert scale. We calculated the correlativity between each of the fitness values and each of the musical feature values. If a fitness value was not well correlated with any of the musical feature values, we removed the impression word pair. At the same time, we removed a musical feature value from the candidate, if it was not well correlated with any of the fitness values. As a result, we listed the impression word pairs and musical features shown in Table 2 in our implementation.

3.1.2 Dictionary construction

We constructed a Japanese dictionary containing noun, verb, adjective, and adverb, with normalized values representing the fitness to all the impression word pairs. Here, we defined the fitness value corresponding to one of the impression words as "1", and the value corresponding to the other word as "-1". We calculated the fitness values of each word in the dictionary for each of the impression word pairs, applying a semantic orientation calculation technique [16]. As a result, the j-th word in the dictionary has a M_w di-

Table 2. Finally selected impression word pairs and musical features.

Impression word pairs for chord progression
Bright - Dark
Enjoyable - Wistful
Tripping - Quiet
Energetic - Calm
Musical features for chord progression
Average of tones
Distribution of tones
Number of simultaneous tones
Ratio of inharmonic tones
Frequency of major, major seventh, and minor seventh chords
Impression word pairs for rhythm progression
Fast - Slow
Light - Heavy
Brassy - Simple
Musical features for rhythm progression
Frequency of tones for Toms, Snare drums, Bass drums, cymbals, and High-hats
Total number of tones
Frequency of 16-beat notes
Frequency of triplets

mensional vector $g_j = \{g_{j1}, g_{j2}, ..., g_{jM_w}\}$, where M_w is the number of impression word pairs, and g_{ji} is the fitness value of the j-th word to the i-th impression word pair.

3.2 Learning of impression and emotion

This step learns the relationships between the musical features and these impressions. This step is applied once for each user as a preprocessing. Our current implementation supposes to ask a user to listen to the sample chord and rhythm progressions, and answer their fitness to the impression word pairs. This process then calculates the relationships between the musical features of the sample chords/rhythms and the fitness values answered by the user. Here, our current implementation applies a linear multi-regression analysis to solve the relationships, because we suppose the relationships can be approximated as linear functions. This study applies the following equation to express the relationships between the musical features and the fitness values:

$$f_i = \sum_{j=1}^{M} a_{ij} m_j \quad (1)$$

Here, f_i is the fitness value for the i-th impression word pair, M is the number of musical features, m_j is the j-th musical feature, and a_{ij} is the coefficient for the i-th impression word pair and the j-th musical feature. The linear multi-regression analysis solves the values of the coefficients, given the set of values of f_i and m_j. Our implementation applies this process independently to chords and rhythms. We can estimate the impression of later provided chord and rhythm progressions by calculating the fitness values using the above equation, after solving the coefficients of the equation.

3.3 Interactive process

This step synthesizes music adopting to the impression and emotion of input narratives. Our implementation interactively provides the synthesized music when a narrative is given. This process is divided into the following components: document analysis, selection of chord and rhythm, and music synthesis.

3.3.1 Document analysis

This step firstly applies a morphological analysis to the input narratives. Our current implementation applies an open source Japanese morphological analysis software MeCab [17] to the narratives. It then extracts noun, verb, adjective, and adverb from the result, and then calculates the average of fitness values recorded in the dictionary generated by the preliminary data construction process, for each of the impression word pairs. The technique treats the average values as the estimated impression and emotion of the narrative.

Here, long narratives or novels contain progression and variation of impression and emotion. Therefore, it is not always adequate to calculate the average impression/emotion of the whole narrative. Our implementation calculates the averages of the fitness values scene-by-scene. Currently we suppose that input narratives are manually divided to multiple scenes. We would like to apply automatic scene recognition techniques to divide the input narratives as a future work.

As a result, we express the impression of a scene of the input narrative as a M_w dimensional vector $h = \{h_1, h_2, ..., h_{M_w}\}$. We calculate the fitness value to the i-th impression word pair as $h_i = \frac{1}{M_s} \sum_j^{M_s} g_{ji}$, where M_s is the total number of words appeared in the scene. g_{ji} is the fitness value of the j-th word to the i-th impression word pair, extracted from the preliminary constructed dictionary.

3.3.2 Selection of chord and rhythm

Next, this step selects chord and rhythm progressions. The technique calculates the musical features m_j for all the prepared chords and rhythms, and then estimates the fitness values f_i for the chords and rhythms by using the equation (1). Our current implementation uses the four-dimensional fitness values for the following impression word pairs, [Bright - Dark], [Enjoyable - Wistful], [Tripping - Quiet], and [Energetic - Calm] for the chord progressions. For the rhythm progression, it uses the three-dimensional fitness values for the following impression word pairs, [Fast - Slow], [Light - Heavy], and [Brassy - Simple].

Our technique generates trajectories of the fitness values consisting of N_{scene} vertices and $N_{scene} - 1$ segments, where N_{scene} is the number of scenes. It then selects the chords and rhythms so that the trajectories of the fitness values of the selected chords or rhythms looks similar to the trajectory generated from the input narrative. Our current implementation selects sets of chords or rhythms of the scenes which minimizes the following formula:

$$min\left(\alpha \sum_{i}^{N_{scene}} \|\mathbf{f}_i - \mathbf{g}_i\| + (1-\alpha) \sum_{i}^{N_{scene}-1} \left(1 - \frac{\mathbf{f'}_{i,i+1} \cdot \mathbf{g'}_{i,i+1}}{|\mathbf{f'}_{i,i+1}||\mathbf{g'}_{i,i+1}|}\right)\right) \quad (2)$$

Here, α is a user-defined constant real value satisfying $0 \leq \alpha \leq 1$, $\mathbf{f}_i$ is a fitness value vector of a chord or rhythm for the i-th scene, $\mathbf{g}_i$ is a fitness value vector of the i-th scene calculated from the input narrative. $\mathbf{f'}_{i,i+1}$ and $\mathbf{g'}_{i,i+1}$ are defined as follows:

$$\mathbf{f'}_{i,i+1} = \mathbf{f}_{i+1} - \mathbf{f}_i, \mathbf{g'}_{i,i+1} = \mathbf{g}_{i+1} - \mathbf{g}_i. \quad (3)$$

The first term of equation (2) attempts to minimize the sum of distances between the fitness value vectors of the chords/rhythms and narratives, while the second term attempts to minimize the geometric differences of the trajectory between the chords/rhythms and narratives. We expect this definition realizes the selection of chords and rhythms taking into account the total variation of the generated music in addition to the similarity of impressions.

3.3.3 Music synthesis

Finally, this step synthesizes the selected chord and rhythm. Our current implementation supposes that the chords and rhythms are stored as independent MIDI files which contain a single track for the chord or rhythm. It simply copies the tracks of selected chord and rhythm into another MIDI file consisting of the two tracks. This implementation will be extended to incorporate melodies as a future work.

4. EXPERIMENT AND DISCUSSION

This section introduces our experiments with the presented technique. We had the following steps for the experiments:

Step 1: We asked participants to listen to the sample chords and rhythms. Then, we asked them to answer the fitness of the impression word pairs in six-point Likert scale, to learn their sensibility for the music.

Step 2: We generated tunes for input narratives applying the learning results of each of the participants. Then, we asked them to evaluate the degree of coincidence of the impression between the narrative and tune in five-point Likert scale. We also asked them to freely comment their impressions for the output tunes.

Here, all chords recorded in the sample MIDI files were played as half notes, and their timbre specified by the program number was piano, in this experiments. Lengths of all sample MIDI files of chords and rhythms were eight measures, and their tempo was 120 BPM (Beat Per Minute).

Table 3. The contents of the input narrative.

Scene 1	your profile of when you sing with smile was very beautiful.
Scene 2	I fell in love with you by looking that profile.
Scene 3	But, I was disappointed in love.

Table 4. The fitness for impression word pairs in each scene of the input narrative.

	Scene 1	Scene 2	Scene 3
Bright - Dark	-0.034480	0.013741	-0.176004
Enjoyable - Wistful	-0.115318	0.171532	0.122126
Tripping - Quiet	-0.149857	0.617872	0.580159
Energetic - Calm	0.325041	0.528052	0.531251
Fast - Slow	-0.122990	0.236133	0.220244
Light - Heavy	0.564965	0.436106	-0.091217
Brassy - Simple	0.521458	0.468449	-0.241370

Table 5. Results and evaluation of participants A and B.

Participant A			
	Chord	Rhythm	Evaluation
Scene 1	chord 12	rhythm 2	4
Scene 2	chord 9	rhythm 3	5
Scene 3	chord16	rhythm 11	1
Total evaluation	2		
Participant B			
	Chord	Rhythm	Evaluation
Scene 1	chord 9	rhythm 19	2
Scene 2	chord 11	rhythm 19	3
Scene 3	chord 16	rhythm 16	4
Total evaluation	4		

In the Step 1, we provided the same sample chords and rhythms for both the participants. After asking participants to listen to the generated tunes in the Step 2, we showed participants the contents of input narratives and explained how the scenes are split. We did not explain how the narrative was split in order, because we wanted participants to be unconscious of contents of the input narrative while they were evaluating the output music. We prepared a dramatic short narrative divided to three scenes in this experiment. Participants evaluated three tunes generated for the scenes of the narrative. Table 3 describes the scenes, and Table 4 shows the fitness of impression word pairs.

The participants of this experiment were two female students in the master's course, who had experiences and expert skills of musical instruments and vocals, and a certain level of musical knowledge. The length of the MIDI files were unified in 16 seconds in this experiment. As a result, lengths of all the tunes synthesized by the proposed technique were 48 seconds. We prepared 23 pieces of chords and rhythms for this experiment.

Table 5 shows the chord/rhythm selection results for two participants, and the evaluations by the participants. The result denotes the synthesized tunes are different despite completely same narrative was provided to the two participants. This suggests our technique can generate different tunes according to the sensibility of each user.

Let us discuss on the results of participant A. Table 6 and Table 7 show the fitness values for chords and rhythms respectively.

Participant A mentioned that "I had an impression that wistful event was happened but I thought it was not a serious scene" while listening to the tune for Scene 1. The fit-

Table 6. The fitness values of participant A for the chords in each scene.

	Scene 1	Scene 2	Scene 3
Bright - Dark	-0.166850	0.888388	0.014798
Enjoyable - Wistful	-0.270462	0.165336	-0.190081
Tripping - Quiet	0.090049	0.443500	0.382450
Energetic - Calm	0.042111	0.653096	-0.068819

Table 7. The fitness values of participant A for the rhythms in each scene.

	Scene 1	Scene 2	Scene 3
Fast - Slow	0.075839	0.496695	0.033937
Light - Heavy	0.145658	0.437712	0.752391
Brassy - Simple	0.075839	0.996108	0.485212

ness value for [Enjoyable - Wistful] of the chord for Scene 1 was negative, close to "Wistful". The above comment for Scene 1 is consistent to the fitness value for the chord. Participant A also mentioned "This scene seems happy, and more exciting than the previous scene" for Scene 2. Here, all fitness values for the chord and rhythm in Scene 2 are higher than the fitness values in Scene 1, as shown in Table 6 and Table 7. This variation of fitness values conforms to the transition of the fitness values of input narrative shown in Table 4. Again, these results denote the above comment for Scene 2 is consistent to the variation of fitness values. Actually, evaluations of the participant A for Scenes 1 and 2 were 4 and 5, relatively high, as shown in Table 5. On the other hand, participant A mentioned that "I had an impression that a good event was happened and end up from music, however, the narrative of this scene was sad." for Scene 3. This comment denotes impression of participant A for music and input document was really opposite. Many of the fitness values of Scene 3 in the input narrative are positive as shown in Table 4, which suggest nimble and energetic impression. These fitness values are inconsistent to the comment that participant A felt sad impression after reading Scene 3. To solve this problem, we would like to extend our implementation of document analysis component customizable to users' sensibility.

Next, let us discuss on the results of participant B. Table 8 and Table 9 show the fitness values for chords and rhythms respectively.

Participant B mentioned that "Although I received a cheerful impression like exercising from the music, it was actually a quiet scene with no movements" for Scene 1. However, the fitness value for [Tripping - Quiet] and [Energetic - Calm] of the chord for Scene 1 were negative, as shown in Table 8, where these values denote the quiet impression. This result suggests our experiment might not successfully learn the sensibility of participant B. In addition, participant B mentioned that "I felt that there was no change between Scene 1 and 2, because the rhythm of Scenes 1 and 2 were the same". This comment suggests rhythm was an important musical factor for participant B, and we may need to analyze which musical factor participants remark. Participant B mentioned that "I felt the sudden change to have a serious feeling" while listening to the tune for Scene 3. Actually, many fitness values of chords and rhythms for Scene 3 got smaller than those for Scene 2, as shown in Table 8 and Table 9. This result demonstrates our technique could represent the significant changes of the impression between the scenes. Participant B mentioned that "Entire flow of music substantially coincides with the flow of story I imagined from the music", and finally rated the total evaluation as 4.

Table 8. The fitness values of participant B for the chords B in each scene.

	Scene 1	Scene 2	Scene 3
Bright - Dark	0.059933	0.896135	0.043383
Enjoyable - Wistful	-0.141256	0.108170	0.216329
Tripping - Quiet	-0.235864	0.969589	-0.123636
Energetic - Calm	-0.041046	0.015811	-0.111593

Table 9. The fitness values of participant B for the rhythms in each scene.

	Scene 1	Scene 2	Scene 3
Fast - Slow	0.507989	0.507989	0.322039
Light - Heavy	0.18577	0.18577	0.363906
Brassy - Simple	0.642532	0.642532	-0.111264

Our experiment assigned the same chord progression for Scene 3 for participants A and B, as shown in Table 6 and Table 8. However, participant A mentioned "I got the impression like happy ending", while participant B answered "It is a serious scene". These comments are actually opposite. This result suggests that the same chord and rhythm progression may give different impression to different users.

The results introduced in this section suggest that the impression of input narratives was close to the impression of generated music in many cases. We would like to customize the document analysis based on users' sensibility so that we can improve the degree of coincidence of the impression between the music and narratives.

5. CONCLUSION

We proposed a technique to synthesize music based on emotion and impression of the input narratives. This technique selects the chords and rhythms to the scenes of the narratives using by estimating the fitness values for the impression word pairs from the musical features. In the preliminary data construction step, we selected sets of impression word pairs as a result of survey on the correlations between musical feature values and pairs of sensibility words. This paper introduced experiments which generated different impression of music in response to the sensibility of the participants. The result suggests that we could generate tunes which roughly match to the fitness values specified by the sensibility polarity dictionary, while we still have issues on document analysis.

Our future issues include the following:

- Review of the association of impression word pairs

and musical features.

- Re-design of interface to input the sensibility of users.
- Implementation of automated recognition of scene breaks in the narratives.
- Association of temporal deployment of music and narrative.
- Embedding melodies while the synthesis of music.

The presented study listed candidates of music features of chords and rhythms excluding several important ones such as velocity of the notes. We suppose participants might have bias in representation of impressions and feelings because such important features were constant in our sample chords and rhythms. We would like to review the music features again, and then extend our implementation by adding important features.

We received several comments from the participants of the presented experiment regarding the methodology of the learning step. One of the typical comments was that "I could not keep the constant criteria in mind, while listening to the sample chords or rhythms, and answering the fitness values." Reflecting these comments, we would like to test other interfaces to input the sensibility of users. Hevner presented a music evaluation method [7] which asks participants to choose one of the word groups which matches to the listened tunes. Several studies on music evaluation applied tournament methods which ask participants to comparatively select one of the tunes, and finally specify the best tune. We would like to apply such methods to appropriately learn the preferences of participants.

Our current implementation learns users' sensibility just for automatic selection of chords and rhythms. We expect our technique can be improved by re-defining fitness values of characteristic words in the preliminary constructed dictionary according to the semantics of the narratives or users' sensibility. We would like to implement a mechanism to customize the dictionary based on this discussion.

Finally, we would like to develop the melody selection process, in addition to larger database construction for chords and rhythms, to realize the generation of more impressive and emotional music. Our on-going work applies an automatic music composition method featuring a genetic algorithm [13] and maximum likelihood estimation [5].

6. REFERENCES

[1] P. Casella, A. Paiva, MAgentA: An Architecture for Real Time Automatic Composition of Background Music *International Workshop on Intelligent Virtual Agents*, pp. 224-232, 2001.

[2] R. Cruz, A. Brisson, A. Paiva, E. Lopes, I-Sounds - Emotion-Based Music Generation for Virtual Environments, *Affective Computing and Intelligent Interaction*, pp. 769-770, 2007.

[3] P. Dunker, P. Popp, R. Cook, Content-Aware Auto-Soundtracks for Personal Photo Music Slideshows, *IEEE International Conference on Multimedia and Expo*, pp. 1-5, 2011.

[4] J. Endo, T. Kitadate, T. Ogata, A Music Generation/Expression Mechanism in the Narrative Generation System: A Consideration from the Application Systems, *The 29th Annual Meeting of the Japanese Congnitive Science Society*, pp. 3-29, 2012.

[5] S. Fukayama, K. Nakatsuma, S. Sako, Y. Yonebayashi, T.-H. Kim, S.-W. Qin, T. Nakano, T. Nishimoto, S. Sagayama, Orpheus: Automatic Composition System Considering Prosody of Japanese Lyrics, *Entertainment Computing - ICEC 2009*, Springer Berlin Heidelberg, pp. 309-310, 2009.

[6] T. Hasegawa, T. Nishimoto, N. Ono, S. Sagayama, Composer Identification from MIDI Data by Combination Features of Pitch and Duration based on Musical Knowledge, *IPSJ SIG Technical Reports[MUS] Vol. 2009-13*, pp. 47-52, 2009.

[7] K. Hevner, Experimental studies of the elements of expression in music, *American Journal of Psychology*, Vol. 48, pp. 246-268, 1936.

[8] T. Ikezoe, Y. Kajikawa, Y. Nomura, Music Database Retrieval System with Sensitivity Words Using Music Sensitivity Space, *Journal of Information Processing Society in Japan*, Vol. 42, No. 12, pp. 3201-3212, 2001.

[9] K. Ishizuka, T. Onisawa, Generation of Variations on Theme Music Based on Impressions of Story Scenes Considering Human's Feeling of Music and Stories, *International Journal of Computer Games Technology*, 2008.

[10] K. Kitahara, H. Watanabe, D. Ando, Method for Automatic Generation of Music Reflecting the Web Activity, *Japanese Society for Sonic Arts*, Vol. 1, No. 1, pp. 8-11, 2012.

[11] H.-C. Lee, I.-K. Lee, Automatic Synchronization of Background Music and Motion in Computer Animation, *Computer Graphics Forum*, Vol. 24, No. 3, pp. 353-361, 2005.

[12] C.-T. Li, M.-K. Shan, Emotion-based Impressionism Slideshow with Automatic Music Accompaniment, *ACM International Conference on Multimedia*, pp. 839-842, 2007.

[13] Y. Maeda, Y. Kajihara, Automatic Generation of Musical Tone Row and Rhythm Based on the Twelve-Tone Technique Using Genetic Algorithm, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 14, No. 3, pp. 288-299, 2010.

[14] J. Nakamura, T. Kaku, K. Hyun, T. Noma, S. Yoshida, Automatic Background Music Generation based on Actors Mood and Motion, *The Journal of Visualization and Computer Animation*, Vol. 5, No. 4, pp. 247-264, 1994.

[15] H. Saito, *Kokoro wo ugokasu oto no shinnrigaku (Psychology of sound that moves the sensibility)*, YAMAHA Music Media, 2011.

[16] H. Takamura, T. Inui, M. Okumura, Extracting Semantic Orientations Using Spin Model, *Annual Meeting on Association for Computational Linguistics*, pp. 133-140, 2005.

[17] Joint research unit project of Kyoto University Graduate School of Informatics and NTT Communication Science Laboratories, *MeCab*. http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html, 2015.2.24

CrossSong Puzzle: Generating and Unscrambling Music Mashups with Real-time Interactivity

Jordan Smith, Graham Percival, Jun Kato, Masataka Goto, Satoru Fukayama
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{ jordan.smith, graham-percival, jun.kato, m.goto, s.fukayama } @aist.go.jp

ABSTRACT

There is considerable interest in music-based games, as the popularity of Rock Band and others can attest, as well as puzzle games. However, these have rarely been combined. Most music-based games fall into the category of rhythm games, and in those games where music is incorporated into a puzzle-like challenge, music usually serves as either an accompaniment or reward. We set out to design a puzzle game where musical knowledge and analysis would be essential to making deductions and solving the puzzle.

The result is the CrossSong Puzzle, a novel type of music-based logic puzzle that truly integrates musical and logical reasoning. The game presents a player with a grid of tiles, each representing a mashup of measures from two different songs. The goal is to rearrange the tiles so that each row and column plays a continuous musical excerpt.

Automatically identifying a set of song fragments to fill a grid such that each tile contains an acceptable mash-up is our primary technical hurdle. We propose an algorithm that analyses a corpus of music, searches the space of possible fragments, and selects an arrangement that maximizes the "mashability" of the resulting grid. This algorithm and the interaction design of the system are the main contributions.

1. INTRODUCTION

Why is listening to music enjoyable? One hypothesis is that a listener's pleasure derives from their ability to detect patterns in the music, thereby "compressing" it in their mind [1]. There is some evidence that, compared to other works, compositions widely regarded as musical masterpieces may be more compressible, despite having a more complex surface representation [2]. Whether or not this is the only explanation, music shares an important trait with puzzles: pattern identification is central to the enjoyment of both. In the case of logic puzzles, such as sudoku, discovering patterns helps the solver to make deductions about how to complete the puzzle.

While being enjoyable for arguably similar reasons, there are few activities that target those with an interest in both music and puzzles. Devising a satisfying combination of active listening and puzzle-solving is a difficult task. Puzzles, including jigsaws and crosswords, are usually solved at a leisurely pace—interruptions are no hindrance—while music is defined by its happening in time, and interruptions or sudden changes in rhythm or playback can be disturbing. We have embraced the challenge of wedding these two forms together and have made a real-time puzzle game in which the solver listens to the audio "clues" without interruptions.

Copyright: ©2015 Jordan Smith 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.

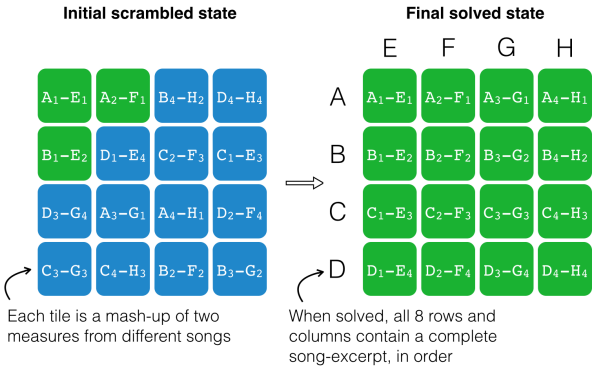


Figure 1: CrossSong puzzle overview. Green tiles indicate correct placement. The solver cannot see the labels and must deduce the correct order by listening to the tiles. Gameplay video:
https://www.youtube.com/watch?v=lOQH3bdIgyo

The result is the CrossSong Puzzle (Figure 1). The puzzle consists of a 4x4 grid of tiles, where each row and column represents a four-measure excerpt of a song. Each tile thus represents one measure-long mashup of two songs. The solver is presented with a scrambled grid, and the object of the puzzle is to discover the correct arrangement of tiles by listening to them. Each excerpt has been time stretched to the same duration so that all beats match. Gameplay is continuous, with each tile playing one after the other with a constant tempo, to prevent the player from being distracted by the interruptions.

The puzzle resembles a musical version of a 4x4 sliding-tile puzzle, in which the goal is to reconstruct an image given similar constraints. However, it more strongly resembles a crossword puzzle in its construction. A crossword setter must find suitable words to fill a grid such that wherever two words cross, the same letter is used. Likewise, to make a pleasing CrossSong puzzle, we must find suitable song excerpts such that wherever two songs cross, a pleasing mashup is made. Discovering a set of excerpts

where this is possible is a formidable but necessary challenge: if the mashups are dissonant or poorly matched rhythmically, the resulting discord will make gameplay tedious. The algorithm we developed for doing this, based on the work of [3], is one of our main contributions.

The other main contributions are the design of the puzzle itself and the interface used to solve it. Both were refined and tested iteratively, and the result is a puzzle that is challenging but accessible.

The rest of the paper is structured as follows. The next section reviews existing combinations of music and puzzles, as well as previous work in mashability estimation. Section 3 gives a formal overview of the proposed CrossSong Puzzle design, including gameplay and implementation. Section 4 describes the algorithm that answers the technical challenge stated above. Section 5 discusses the iterative testing, design principles, and possible improvements to the system. We give concluding remarks in Section 6.

2. RELATED WORK

In this section, we first review prior effort on using music as part of interactive play, including examples in both pre and postcomputer age. Second, we review existing software for creating and estimating the quality of mashups.

2.1 Music as Part of Interactive Play

If we do not limit the scope to computer-aided puzzles, there is one tradition of musical puzzles that dates at least to the 15th century: puzzle canons. The puzzle consists of a single monophonic melody, and the solver (usually a student of composition or other expert) must discover how to realize it as a canon. An early example of turning music-making into a game is the musical dice game of Western Europe, dating to the 1700s [4], in which random rolls of the dice were used to choose a selection of score fragments which were then performed for the amusement of the assembled.

There are many web- and smartphone-based games today which are based on music; however, a partial survey [5] suggests that the market is dominated by sound banks, multimedia players, instrument emulators, and music-creation apps like synthesizers and sequencers. Among the music-related puzzles we discovered, the link between the music and the puzzle mechanics were not very strong; in most cases, the logical reasoning is separate from the music, which serves more as a progress indicator or as a reward generated by the correct solution to the puzzle (e.g., Auditorium [1], Chime [2], Lumines [3]). Even when the music is deeply integrated into the puzzle structure, such as with FRACT OSC [4], musical insight is not required to solve the challenges. Other related music-based games include the popular genre of rhythm games (e.g., Guitar Hero [5], Hatsune Miku: Project DIVA [6], Idolmaster [7])—but these are better described as physical challenges than as logic puzzles.

We would like to see a puzzle where the music is the *source* of information needed by the solver, and where careful listening is required. To our knowledge, the only predecessor with this feature is the puzzle game developed by Hansen et al. [6], who developed a musical analogue of a jigsaw puzzle. A 15-second excerpt of music is divided into pieces and the solver's goal is to arrange the pieces from left to right in order to reconstruct the original excerpt, much like jigsaw pieces must be arranged in order to reconstruct an image. As an added challenge, the audio of several pieces has been randomly transposed; the solver must detect and undo these transpositions in order to complete the puzzle.

Their design has a certain limitation, which ours aims to overcome. First, each musical excerpt is divided into pieces at arbitrary timepoints, so the resulting pieces do not sound like coherent fragments. Thus, when the pieces are in incorrect order, the result will sound not only incorrect but also unmusical. It would be preferable to divide the fragments only at beat or downbeat positions. In fact, some music psychology experiments support the view that rearranging parts of a piece of music at a sensible timescale does not necessarily disrupt one's enjoyment of the music [7].

2.2 Automatic Level Creation for Music Games

Creating levels for music games could be done with manual effort, but is cumbersome and makes it difficult to customize the gaming experience based on the users' needs. For instance, matching the audio clips to beat boundaries could be done with manual editing of the audio files, but a better approach is to generate levels based on rhythmic information extracted from the audio automatically. In this way, users can create levels based on their music libraries. Automatic methods of level creation have already been developed for music rhythm games such as Guitar Hero, Beat the Beat [8] and AudioSurf [8].

For the CrossSong puzzle, we require an algorithm that can do two things: first, automatically align the beat of two pieces with beat-tracking; and second, estimate the quality of the resulting mashup at multiple shifts in pitch. Many tools are capable of estimating beat locations to facilitate the creation of mashups, such as the Echo Nest Remix API [9]. Beat-Sync-Mash-Coder [9] computes this beat information and uses it to automatically synchronize the playback of two clips, but the portion of each song to use must be manually selected, and the system does not attempt to match the pitch of the clips. The commercial system Mixed In Key [10] estimates the mutual harmonic compatibility of all songs in a collection, and can recommend source material for users to create mashups on their

[1] http://www.cipherprime.com/games/auditorium/
[2] http://www.chimegame.com/
[3] http://lumines.jp/
[4] http://fractgame.com/
[5] http://www.guitarhero.com/
[6] http://miku.sega.jp/arcade/en/
[7] http://idolmaster.jp/
[8] http://www.audio-surf.com/
[9] http://echonest.github.io/remix/
[10] http://mashup.mixedinkey.com/HowTo

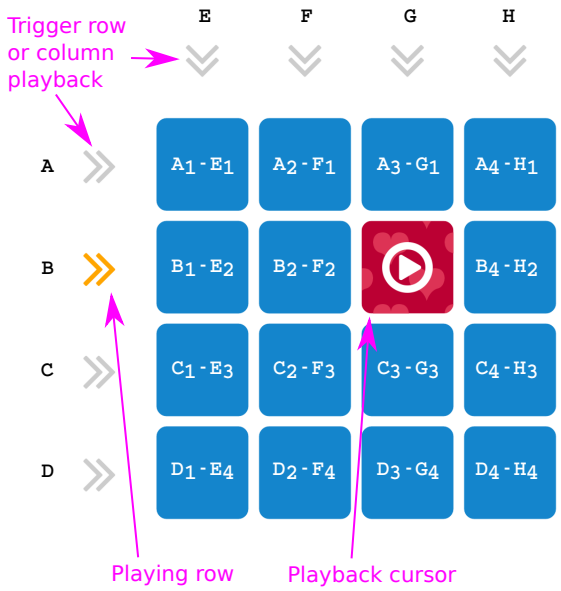
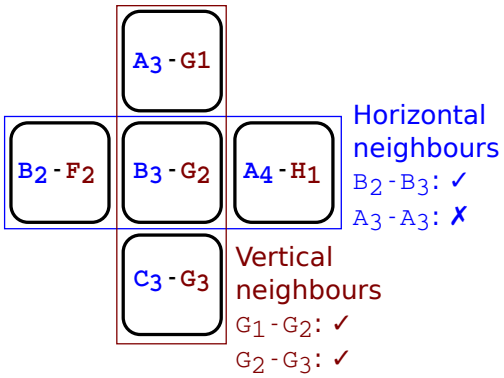
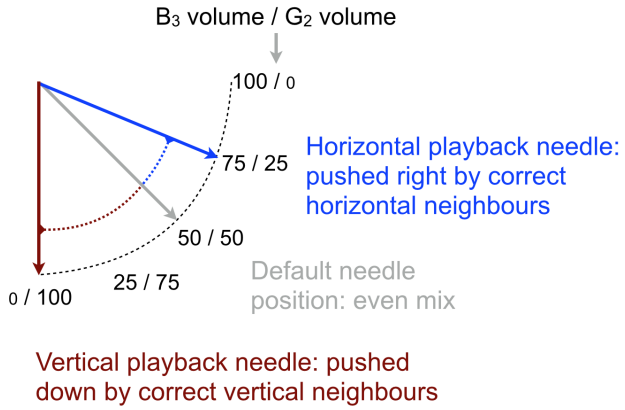


Figure 2: CrossSong Puzzle in its solved state, with labels added to each tile to illustrate the arrangement of music clips. Each tile contains a mashup of two clips; clip label X_i indicates the ith measure of of song X. Solvers never see the tile labels, and begin with the tiles in random order.



(a) Illustration of relative cell correctness.



(b) Illustration of how clips are mixed depending on correctness.

Figure 3: Diagrams for how neighbour correctness is calculated for a given tile, B_3-G_2, and the resulting balance when played as part of a row or column.

own. However, the compatibility estimate is on a song-to-song basis with no timing information; this is too coarse for our purpose, since the compatibility of two excerpts can be greatly affected by the phase of the excerpts.

Among existing systems, AutoMashUpper [3] fulfills our requirements best. First, it performs beat, downbeat, and phrase-level boundary detection, since mashups between phrases that are intact and aligned downbeat-to-downbeat are understood to sound better. Second, it estimates the harmonic, rhythmic and spectral compatibility of two phrases at all possible shifts in pitch and time. The harmonic compatibility of two segments is taken as the correlation between chromagrams estimated from the audio. Rhythmic compatibility is estimated in the same way, using a rhythmic feature derived from the pattern of estimated kick and snare onsets. Finally, the coarse spectra from each segment are compared; the flatter their sum, the more the two excerpts are deemed to have complementary spectra, and the greater their mashability. Details of this algorithm can be found in [3]. In Section 4 we describe how the algorithm was adapted for our needs.

3. CROSSSONG PUZZLE

The CrossSong Puzzle was described briefly in the introduction. In this section, we explain the design and construction of the puzzle in more detail. In the Section 5, we explain how our design evolved over a series of user tests.

In its solved state, the puzzle contains excerpts from 8 different songs, labelled A–H, one for each row and column of the grid. (See Figure 2.) Each excerpt X is 4 measures long; each of these measures, X_1–X_4, is associated with a different tile, and each tile is a mashup of measures from two songs. The solver begins the puzzle with the tiles arranged randomly and their task is to determine the correct order by listening to the tiles. Audio playback is continuous: the tiles are sounded in order from left to right, top to bottom, and the tile currently being played is highlighted. When the last column has finished playing, playback continues at the first row. All the tiles have the same duration and tempo, so even in the initial random configuration of tiles, the music has rhythmic coherence.

During gameplay, the solver can click on any two tiles to swap their position. They may also click on arrows outside the grid to choose which row or column to begin playing after the current one has ended. A link to a gameplay video is given in Figure 1. Solving a single puzzle takes roughly 10 minutes.

Normally, the two clips in each tile are played with equal loudness. However, as a reward for partial progress, the balance between the clips changes if the tile is positioned correctly with respect to its neighbours. The more correct neighbours, the more the mixing is reduced. The concept of "relative cell correctness" is illustrated in Figure 3a. In this example, the tile B_3-G_2 has one correct horizontal neighbour, since the tile B_2-F_2 belongs to its left in the solved puzzle. The impact of this arrangement is seen in Figure 3b. When B_3-G_2 is played as part of the current row ("horizontal playback"), instead of the mix being 50/50, it will be 75% B_3 and 25% G_2. When played as part

of the current column ("vertical playback"), since both vertical neighbours are correct, the mix will be 100% G_2. It does not matter if B_3-G_2 is in the correct place in the 4x4 grid; this audio clue is based only on relative correctness.

3.1 Platform

We chose to implement the game as a web-based application. This has the advantage of making it instantly cross-platform: we have played it successfully on a desktop with a mouse, on a smartphone with a touchscreen, and even on a large-format touchscreen with multiple users (as pictured in Figure 6).

Once a puzzle has been generated (discussed in Section 4), it is presented to the player in a JavaScript interface. We used the Web Audio API, allowing us to leverage the increasing capabilities of modern web browsers for interactive audio applications [10]. This allows the solving portion of the puzzle (as opposed to the generation phase) to scale to many users, as the server need only provide the html, css, javascript, and audio files to the user. The actual gameplay logic, as well as the audio mixing and scheduling, is performed on the local client computer.

Using a central server to generate and serve the audio has advantages and disadvantages. The main advantage is that we can perform audio analysis and generate puzzles using any language of our choice, rather than being restricted to javascript. Two disadvantages are that users are restricted to audio which is available on the server (i.e. they cannot use their own personal music collection), and if many users were attempting to create puzzles at the same time, the server could easily become overloaded. The latter problem is mitigated by caching all generated puzzles, so re-using an old puzzle has virtually no cost. Given that javascript audio-processing libraries are relatively new, we chose to use a central server.

4. PUZZLE CREATION ALGORITHM

As described in Section 2, AutoMashUpper estimates the mashability of two excerpts as a function of their harmonic, rhythmic and spectral compatibility, considering a range of possible transpositions. AutoMashUpper finds, for a given section of a song, the single best matching segment among a list of other songs. Our goal is different: to find a set of 8 song excerpts, each divisible into 4 equal-sized measures, such that, when arranged into a 4x4 grid, each combination of measures forms a good mashup.

The problem is similar to generating a crossword puzzle grid: for that task, letters must be found which create acceptable words in each direction. However, a strict similarity function applies for letters— they are either the same or not—but no binary measure of acceptableness is available to us. The crossword generation problem, though seemingly straightforward relative to our task, has been researched for decades. It is a complex search problem that is NP-complete [11].

Our primary obstacle is the incredibly large space of combinations to search. Each excerpt can begin on any downbeat, meaning there are roughly 100 choices of excerpt in a typical song (this is the case for a 120BPM song that lasts 3:20). For 8 songs, this gives $100^8 = 10^{16}$ possible sets of excerpts. For each set, there are $8!/2 = 20,160$ ways of arranging them in the 4x4 grid. (The factor of 2 reduction recognizes that any arrangement and its transpose are equivalent.) Finally, each excerpt may be transposed up to 3 semitones upwards or downwards, increasing the space by a power of 7, approximately.

Before explaining how we reduced this search space, here is the overall procedure for computing mashability, searching for an optimal mashup, and processing the audio.

1. Compute audio features and detect phrase boundaries according to [3]
2. Compute mashability of all phrase-initial segments at all different delays. Retain mashability of optimal transposition of each.
3. Search loop:
   (a) Select one random excerpt from each song.
   (b) Find arrangement of these excerpts into grid with maximum mashability.
4. Repeat loop for pre-determined amount of time, and keep the best solution.
5. Process audio clips:
   (a) Apply time-stretching and pitch shift to match all excerpts using Rubberband library [12]
   (b) Match perceptual loudness of all excerpts using Replay Gain method [13]

4.1 Search optimizations

We first reduced the search space by restricting ourselves to excerpts that begin at one of the section boundaries estimated by AutoMashUpper. Doing so increases the odds that each excerpt will be an intact phrase of a song.

Our next optimization is to, for a pair of excerpts, only consider the transposition that gives the optimal mashability. This reduces the search space by a power of 7, but it can lead to problems: the final grid will require that all the clips be transposed to match each other, but these optimal transpositions can easily be infeasible. For example, suppose we choose clips A, B, E and F on the basis of their optimal mashability, disregarding the required transpositions. We may then match E_1 to A_1, F_1 to A_2, and B_1 to E_2. However, this fixes the transpositions of B_2 and F_2, and the result may be dissonant.

In order to mitigate this, we compute mashability not between individual measures (such as A_2 and F_1), but between full excerpts (such as A and F with the latter offset by one measure). This creates some mutual dependence in the mashability values. In the previous example, we can expect that B_2 and F_2 will match as long as B_1 and F_1 match. Assuming all the mashability values were high, we know that B_1 matches E_2, which matches A_2, which matches F_1. Hence, to the extent that harmonic compatibility is transitive, we can use a greedy approach without worrying too much about conflicts in transpositions.

4.2 Computation time and usability

Feature processing (step 1 in the list above) requires roughly 14 seconds to analyze each song (based on an average 3-minute song). Step 2, computing the mashability, takes roughly 0.5 seconds per pair of songs, or 14 seconds overall for an 8-song puzzle. For a given choice of 8 excerpts, all possible grid arrangements can be searched in roughly 0.03 seconds (step 3b). The remaining bottleneck is incredible number of random sets of excerpts, so we simply conduct a random search within a set time limit. In our tests, acceptable solutions were found in less than a minute of searching. Finally, the audio processing using Rubberband and Replay Gain takes about 10 seconds.

If the algorithm has access to the library beforehand, steps 1 and 2 of the algorithm can be executed in advance, in which case a good puzzle can be created in around a minute. Otherwise, an additional 2 minutes of pre-processing must take place.

Lastly, it should be noted that the algorithm makes many strong assumptions about the rhythmic regularity of the piece: constant tempo, constant 4/4 meter, and for the most part, phrases that are 2^n measures long. While these assumptions clearly do not apply to all music, they are prerequisites for our purpose. The user should be aware of this constraint and avoid selecting music in different time signatures. In the future, an automatic meter-detection step could be developed to quickly warn users of incompatible songs.

5. DESIGN DEVELOPMENT

A puzzle creator has two contradictory goals: first, to confront the solver with a problem that is very difficult to solve; and second, to ensure that the solver is eventually successful [14]. We iteratively tested a number of puzzle designs in order to strike a balance between posing no challenge and posing an insurmountable one. We also kept in mind some design criteria that are supported by the popular concept of "flow" [15], which seeks to explain why certain activities are more engaging than others. Namely, that the player's goals should be clear and manageable, and that feedback should be frequent and useful. In this section, we describe the sequence of puzzle designs we developed and tested, including the pros and cons of each. Our iterations primarily affected three aspects of the puzzle: first, the balance of visual and auditory hints given; second, the way that the puzzle confirmed the progress of the solver; and third, how the listener's familiarity with the musical excerpts has handled.

Version 1: initial prototype

Our initial prototype worked as described in Section 3. All of the basic gameplay elements of this version— the swapping of tiles, the control of row and column playback, and the fading audio hint based on row correctness illustrated in Figure 3— were retained in future versions.

The puzzle was enjoyable to solve, but it was only solvable by those who knew the music beforehand. None of those who tested this version without knowing any of the music solved it; one user even spent 10 minutes without being certain of the relative position of any tiles, and was very discouraged.

Another problem is that we failed to realize that arranging the tiles in the transpose of the correct solution was logically sound, but not recognized by the system as correct.

Version 2: adding hints

Our second interface included strong visual hints to support the audio: the relative correctness of every tile was shown by displaying heart icons at the boundary with the correct neighbour (see Figure 4a). Also, to resolve the ambiguity of the solution, we added three fixed tiles in the top-left of the grid.

On the plus side, with a few fixed tiles to get started, solvers had an "in" to start the puzzle, and even solvers who were unfamiliar with the music could make progress. Unfortunately, the visual hints made progress far too rapid: once a few tiles had been placed in the correct order, the rest of the puzzle could be more easily as a visual packing problem, or simply by trial and error. Although we agreed that some visual confirmation of one's progress was needed, this version took the focus of the logic away from the audio, defeating the intent of the puzzle. The ideal visual hint would reinforce the auditory hint without adding any new information.

Version 3: refining visual hints

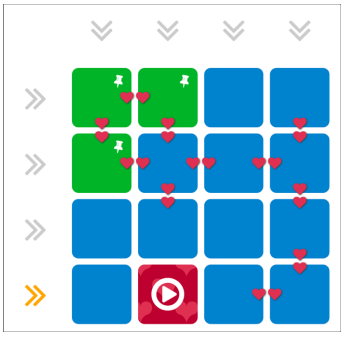
Our solution was to animate the background of the currently playing tile: we added a textured background that flows in the direction of the arrow in Figure 3b. For example, if no neighbours are correct, the background flows in a south-easterly direction; if both horizontal neighbours are correct during horizontal playback, the background flows east. Thus the solver gets a visual confirmation of the relative correctness of the tile, but without extra clues about which neighbouring tiles are correct. Also, the visual clue is only available when the solver *listens* to the tile, so trial and error is too slow to be effective.

Those testing this version reported that the puzzle was still too difficult, for two reasons. First, mentally keeping track of the tiles was taxing, and it was easy to undo one's progress: for example, one might sort several similar tiles into a single row, but then forget which row it is, or accidentally swap a tile away and lose track of it. Second, the puzzle was still very difficult for first-time listeners; many of the mashups were effective enough that it was hard to tell which parts of a tile belonged to which song!

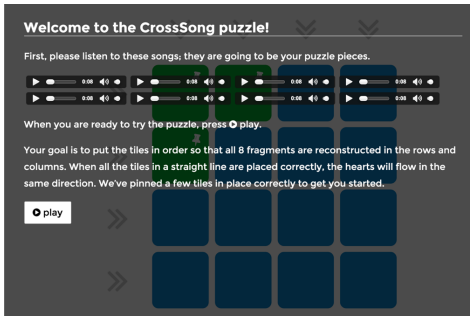
Version 4: improving usability

We added two features to make the game more user-friendly. First, following the example of [6], we added a welcome screen (see 4b) where solvers were allowed to listen to each of the 8 excerpts separately before solving the puzzle — just like jigsaw puzzle solvers can look at the picture on the box first.

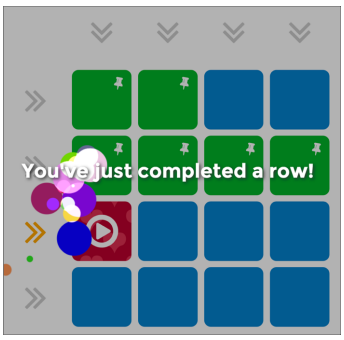
Second, we added a row-confirmation feature (Figure 4c). If all the tiles in a single row or column are placed in their



(a) Visual hints added to Version 2



(b) Welcome screen, added to Version 4



(c) Row confirmation screen, added to Version 4

Figure 4: Screenshots of development versions of CrossSong

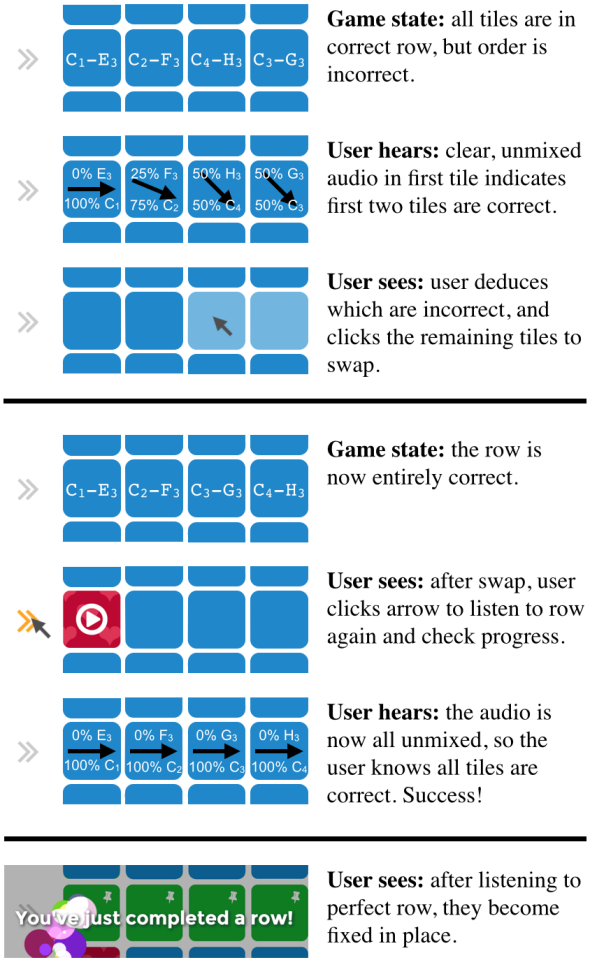


Figure 5: Depiction of a typical gameplay sequence. In the top part, the audio cues help the user identify which tiles arranged incorrect. In the middle part, the user listens to the new arrangement. The bottom part shows the visual feedback provided to the user.

correct position, a congratulatory message appears, and the tiles become fixed in place— but only after the full row (or column) is played, so that randomly shuffling tiles is still a fruitless approach. Fixing the tiles in place prevents undoing one's work but also serves as an encouraging confirmation of partial progress, which is a feature of many engaging puzzles. A typical sequence of gameplay steps leading up to this row confirmation event are depicted in Figure 5.

This final version of the puzzle has most of the qualities we sought: it combines a need for careful listening with logical deduction, and although supported by visual hints, the visual hints do not dominate the puzzle-solving experience. The puzzle sets up a series of rewards (the row and column confirmations) that are achievable whether one is playing with one's favourite songs, or someone else's. Most of all, the game is fun. The game is available to play online at `https://staff.aist.go.jp/jun.kato/CrossSong/`.

Future versions

This section has mostly discussed the development of the core game mechanics, but there are other aspects of the game that could be refined. For example, in order to sustain one's engagement in CrossSong puzzles for more than a few levels, the layout of the initially fixed tiles should change for the sake of variety. Experienced solvers may wish to be able to turn off certain aids, such as the ability to pre-audition the excerpts, or to have correct rows fixed in place. Difficulty can also be increased by creating a larger puzzle; it is trivial to modify our algorithm to generate 8x8 puzzles.

One alternative version that we have implemented is the "multiplayer" mode. Two solvers each choose 4 songs, with excerpts from one solver's songs placed in the rows, and the others in the columns. (This constraint actually reduces the search space slightly for the algorithm in Section 4, reducing the computation time of step 3(b) from 30 ms to roughly 0.8 ms per iteration.) The solvers then work on the puzzle cooperatively on a large screen device (Figure 6).

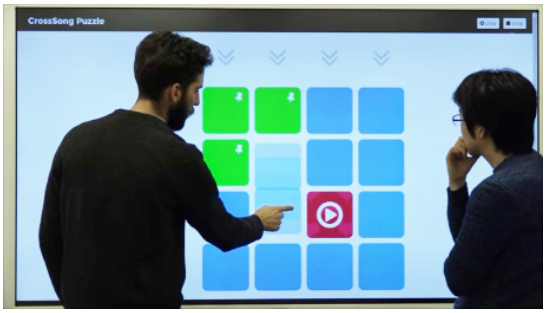


Figure 6: CrossSong Puzzle with two users working cooperatively.

6. CONCLUSION AND FUTURE WORK

We have proposed a novel type of puzzle, the CrossSong, which aims to combine the pattern-learning and pattern-seeking joys of music and puzzles. We have developed an algorithm for generating puzzles from music provided by a user, and an interface for solving them. The software allows (and solving the puzzle requires) the user to explore, in real time, a set of original mashups.

The design was iteratively refined to focus the solver on the musical rather than the visual content, and to provide them with enough confirmation to make this task feasible. We would like to test the system on a larger scale to determine what parameter settings are preferred by a larger set of people. By tracking how fast each puzzle is solved, and the strategies used to solve them, we could refine the design so that the puzzle is rarely solved too quickly or too slowly. Both are outcomes that may reduce the enjoyability of the game.

The algorithm presented in Section 4 could be improved in several ways. For example, in pop songs, most sections are repetitions of other sections; if we detected these repetitions, we could ignore redundant sections and further reduce the search space. Second, the search space could be traversed more efficiently using probabilistic methods such as simulated annealing. Mashability could also be arbitrarily increased by treating the excerpts with harmonic-percussive source separation: this way, we could attempt to pair the drums from one song with the harmonies of another, reducing the severity of any harmonic or rhythmic incompatibility. Testing the usefulness of these improvements, as well as conducting larger-scale user testing, remain our future work.

Acknowledgement

This work was supported in part by OngaCREST, CREST, JST.

7. REFERENCES

[1] J. Schmidhuber, "Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes," in *Anticipatory Behavior in Adaptive Learning Systems*. Springer, 2009.

[2] N. J. Hudson, "Musical beauty and information compression: Complex to the ear but simple to the mind?" *BMC research notes*, vol. 4, no. 1, 2011.

[3] M. E. P. Davies, P. Hamel, K. Yoshii, and M. Goto, "AutoMashUpper: Automatic creation of multi-song music mashups," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, 2014.

[4] S. A. Hedges, "Dice music in the eighteenth century," *Music & Letters*, 1978.

[5] G. Dubus, K. F. Hansen, and R. Bresin, "An overview of sound and music applications for Android available on the market," in *9th Sound and Music Computing Conference, SMC 2012*, 2012.

[6] K. F. Hansen, R. Hiraga, Z. Li, and H. Wang, "Music puzzle: An audio-based computer game that inspires to train listening abilities," in *Advances in Computer Entertainment*, ser. Lecture Notes in Comp. Sci. Springer International Publishing, 2013, vol. 8253, pp. 540–543.

[7] F. Upham and M. Farbood, "Coordination in musical tension and liking ratings of scrambled music," in *Presented at the Society for Music Perception and Cognition Conference*, 2013, p. 148.

[8] A. Jordan, D. Scheftelowitsch, J. Lahni, J. Hartwecker, M. Kuchem, M. Walter-Huber, N. Vortmeier, T. Delbrugger, U. Guler, I. Vatolkin, and M. Preuss, "BeatTheBeat: Music-based procedural content generation in a mobile game," in *Computational Intelligence and Games (CIG)*, 2012.

[9] G. Griffin, Y. E. Kim, and D. Turnbull, "Beat-sync-mash-coder: A web application for real-time creation of beat-synchronous music mashups," in *Acoustics Speech and Signal Processing (ICASSP)*, 2010.

[10] L. Wyse and S. Subramanian, "The viability of the web browser as a computer music platform," *Computer Music Journal*, vol. 37, no. 4, 2013.

[11] M. L. Ginsberg, M. Frank, M. P. Halpin, and M. C. Torrance, "Search lessons learned from crossword puzzles," in *Proc. of the Eighth National Conference on Artificial Intelligence - Volume 1*. AAAI Press, 1990.

[12] C. Cannam, "Rubber band library," http://break-fastquay.com/rubberband.

[13] D. J. Robinson, "Perceptual model for assessment of coded audio." Ph.D. dissertation, University of Essex, 2002.

[14] M. L. Gottlieb, "Secrets of the MIT Mystery Hunt: An exploration of the theory underlying the construction of a multi-puzzle contest," 1998, Bachelor's thesis.

[15] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York, NY, USA: Harper and Row, 1990.

Voice quality transformation using an extended source-filter speech model

Stefan Huber, Axel Roebel
Sound Analysis/Synthesis Team, IRCAM-CNRS-UPMC STMS, 75004 Paris, France
axel (dot) roebel (at) ircam (dot) fr

ABSTRACT

In this paper we present a flexible framework for parametric speech analysis and synthesis with high quality. It constitutes an extended source-filter model. The novelty of the proposed speech processing system lies in its extended means to use a Deterministic plus Stochastic Model (DSM) for the estimation of the unvoiced stochastic component from a speech recording. Further contributions are the efficient and robust means to extract the Vocal Tract Filter (VTF) and the modelling of energy variations. The system is evaluated in the context of two voice quality transformations on natural human speech. The voice quality of a speech phrase is altered by means of re-synthesizing the deterministic component with different pulse shapes of the glottal excitation source. A Gaussian Mixture Model (GMM) is used in one test to predict energies for the re-synthesis of the deterministic and the stochastic component. The subjective listening tests suggests that the speech processing system is able to successfully synthesize and arise to a listener the perceptual sensation of different voice quality characteristics. Additionally, improvements of the speech synthesis quality compared to a baseline method are demonstrated.

1. INTRODUCTION

In this paper we present a method to transform the deterministic and stochastic part of the glottal excitation source. The main motivation of the following paper is the presentation of an improved method for coherent modifications of the glottal pulse shape. The glottal pulse shape is generally accepted to reflect different phonation types of human voice production [1] and different voice qualities being strongly related to the vocal effort [2]. The terminology used in the following is describing the lax-tense dimension of voice quality [3] distinguishing tense (pressed), modal (normal), and relaxed (breathy) voice qualities [4].

Recent research in the speech community has notably improved the speech synthesis quality by explicitly modelling the deterministic and stochastic component of the glottal excitation source [5, 6]. Advanced source-filter decomposition strategies as in [7–9] address finer details defined by extended voice production models for human speech. These approaches analyze an extended feature set to model their transformation and synthesis. The extended feature set consists of: the VTF, the glottal pulse positions and shapes, the energies, and a random component described by spectral and temporal envelopes.

In this paper we present a novel speech analysis and synthesis system extending the source-filter model of [9]. The extension is based on using a DSM and further processing means. The deterministic part is estimated and subtracted from a speech signal to extract the stochastic part [10]. The proposed system separately models the stochastic and deterministic components. It does therefore not correspond to the classical source-filter model. The contribution of the following research and the advancements compared to the baseline method lies in the extended means to estimate the unvoiced stochastic component, to robustly extract the VTF and to handle the variations in energy and signal behaviour implied with glottal source transformations.

The paper is organized as follows. Section 2 presents the novel speech framework. Section 3 discusses the aspects of voice quality transformation. Section 4 introduces the baseline state-of-the-art speech system. Section 5 presents a subjective evaluation based on a listening test of natural human speech. Section 6 concludes with the findings studied in this paper.

Copyright: ©2015 Stefan Huber 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.

2. THE EXTENDED SOURCE-FILTER MODEL

The proposed speech analysis and synthesis system is designed for the utilization in advanced voice transformation and voice conversion applications. It is denoted **PSY** for **P**arametric **S**peech analysis, transformation and **SY**nthesis.

2.1 Voice production model

PSY operates upon the following generic interpretation of the human voice production in the time domain:

$$s(n) = u(n) + v(n) = u(n) + \sum_i g(n, P_i) * \delta(n - P_i) * c(n, P_i) \quad (1)$$

The speech signal $s(n)$ is represented by means of a stochastic (unvoiced) component $u(n)$ and a deterministic (voiced) component $v(n)$. The deterministic component contains the sequence of glottal pulses located at the time positions P_i, each representing a Glottal Closure Instant (GCI) with index i. Each glottal pulse is represented by the glottal flow derivative $g(n, P_i)$. The latter is convolved with a Dirac impulse at the GCI P_i and the VTF that is active for the related position $c(n, P_i)$. The Liljencrants-Fant (LF) model [13] is used to synthesize each $g(n, P_i)$. The LF model is parameterized by a scalar shape parameter R_d [14, 15]. Changing R_d continuously from lower to

higher values will allow changing the LF pulse shape on a continuum from tense to relaxed voice qualities.

For being able to make spectral domain manipulations the speech signal model given in equ. 1 is processed in the spectral domain using the Short-Time Fourier transform (STFT). For brevity the coverage of a few consecutive glottal pulses $g(n, P_j)$ will be denoted as $g_s(n) = \sum_j g(n, P_j) * \delta(n - P_j)$ in the following. The summation over the GCI index j is set to comprise a signal segment of a few glottal pulses being covered by the Hanning window $w_h(n)$ of the STFT. Each pulse position is related to a slightly different VTF being supposed to be minimum phase [12]. The glottal pulse shape and the VTF are assumed to not change within the window and are given approximately by the corresponding parameters in the window center.

We further assume that the filtering processes implied by each convolutional operation between the signal components of equ. 1 involves impulse responses that are shorter than the window length. The STFT of the speech signal is then given by

$$S(\omega, m) = U(\omega, m) + V(\omega, m) \quad (2)$$
$$= U(\omega, m) + G(\omega, m) \cdot H(\omega, m) \cdot C(\omega, m) \quad (3)$$

The STFT frame m is the position of the window center and ω is the frequency variable of the Discrete-Time Fourier Transform (DTFT). For brevity the dependency of all signal spectra with respect to m will be dropped in the following. $U(\omega)$ and $V(\omega)$ are the DTFT of the windowed voiced and unvoiced signals from equ. 1 assuming that g and c and the corresponding DTFT spectra $G(\omega)$ and $C(\omega)$ are quasi-stationary within the window. $H(\omega)$ is the spectral representation of the windowed Dirac impulse sequence $\delta(n - P_i)$. The radiation filter at lips and nostrils level $R(\omega)$ [11] is not explicitly present in the PSY model since it is implicitly contained in the glottal flow derivative $g(n)$ and the unvoiced component $u(n)$.

2.2 Glottal source synthesis and VTF extraction

The LF shape parameter R_d is estimated by the means proposed in [16, 17]. Each GCI is estimated by the method described in [18] and assigned the closest R_d value which is estimated on the STFT time grid. The spectral envelope sequence $\mathcal{T}_{sig}$ is estimated on the input signal $s(n)$ using the True Envelope estimator of [19]. Another spectral envelope sequence $\mathcal{T}_g$ is estimated on the synthesized glottal flow derivative sequence $g_s(n)$. The extraction of the VTF $C(\omega)$ is obtained by dividing $\mathcal{T}_{sig}$ by $\mathcal{T}_g$. The utilization of $\mathcal{T}_g$ in the full-band division is required to suppress the spectral ripples occurring for higher R_d values [15, 20].

2.3 Estimation of the unvoiced stochastic part

The separation of a speech signal $s(n)$ into the contributions of the voiced $v(n)$ and the unvoiced $u(n)$ part is based on the calculation of a residual of a sinusoidal model [21]. The following algorithmic step estimate a) the unvoiced residual $u_{res}(n)$ by deleting sinusoidal content from $s(n)$, b) $u_{HP}(n)$ by high-pass filtering $u_{res}(n)$, c) the unvoiced signal $u(n)$ by scaling $u_{HP}(n)$ in energy.

a) Re-Mixing with De-Modulation: This approach aims to simplify the sinusoidal detection by de-modulating the fundamental frequency F_0 contour and the Hilbert amplitude envelope $\mathcal{H}$ from $s(n)$. The original F_0 contour of $s(n)$ is warped to become flat by means of time varying re-sampling using as target F_0' the mean of the original F_0. The re-sampling operation changes locally and globally the time duration of all signal features. The effect will be inverted after the extraction of the residual. The varying amplitude contour of $s(n)$ is demodulated by means of dividing the signal by its smoothed Hilbert transform $\mathcal{H}(s(n))$ similar as in [5,23]. The smoothing kernel is simply the Hanning window of duration $4/F_T$. This optimally removes all envelope fluctuations that are related to the deterministic components. The resulting signal $s_{flat}(n)$ is flat in amplitude envelope and F_0 facilitating the detection of sinusoids following [21]. It avoids even for relatively high harmonic numbers the energy shift between voiced and unvoiced components [22]. The sinusoidal content is subtracted from $s_{flat}(n)$ and the demodulation steps are inverted so that the original AM-FM modulation is recreated. This generates the unvoiced residual signal $u_{res}(n)$.

b) Below F_{VU} filter: Informal tests confirm that not all sinusoidal content could be precisely estimated and deleted in the frequency band below the Voiced / Unvoiced Frequency boundary F_{VU} [24]. The F_{VU} estimation is based on the signal interpretation splitting the spectrum into two bands. The lower frequency band below the F_{VU} is determined by the voiced component $V(\omega)$. The unvoiced component $U(\omega)$ is located above the F_{VU}. A high pass filter is applied to delete remaining sinusoidal content from $U_{res}(\omega)$ below F_{VU}. The filters cut-off frequency f_c equals the estimated F_{VU} per STFT frame m. A gain of 1 is set in the filters passband equalling the stochastic frequency band $\omega > \omega_{VU}$. A linear ramp with a slope of m_{HP}=-3 dB per octave defines the high pass filtering in the filters stopband. The latter equals the deterministic frequency band $\omega < \omega_{VU}$. The experimental findings show that a heuristically defined threshold of m_{HP}=-3 dB approximates reasonably close enough the desired sinusoidal cancellation in the high pass filtered unvoiced signal $u_{HP}(n)$.

c) Scale to $\mathcal{T}_{sig}$ level: The sinusoidal detection of step a) may be erroneous for some signal segments such as fast transients. The heuristic adaptation of step b) cannot be exact for all cases. The scaling described in equ. 4 minimizes the difference between the envelope $\mathcal{T}_{unv}$ of the stochastic component $U_{HP}(\omega)$ and the envelope $\mathcal{T}_{sig}$ of the signal spectrum $S(\omega)$ above F_{VU} up to the Nyquist frequency F_{nyq}. The DFT bins found closest to the frequencies F_{nyq} and F_{VU} are denoted as k_{nyq} and respectively k_{VU}.

$$\eta = \frac{1}{k_{nyq} - k_{VU}} \sum_{k=k_{VU}}^{K=k_{nyq}} \left(\mathcal{T}_{sig}^{dB}(k) - \mathcal{T}_{unv}^{dB}(k) \right) \quad (4)$$
$$\mathcal{T}_{unv}^{w} = \mathcal{T}_{unv} \cdot (1 - k_{VU}/k_{nyq}) \cdot 10^{\eta/20}$$

η equals the mean difference in dB between $\mathcal{T}_{sig}$ and the spectral envelope $\mathcal{T}_{unv}$.The scaling of $\mathcal{T}_{unv}$ is weighted by the time-varying ratio of F_{VU} versus F_{nyq} as a regularization term to avoid a too high energy scaling. The multiplication of a white noise spectrum with $\mathcal{T}_{unv}^{w}(\omega)$ synthesizes with the STFT the unvoiced signal $u(n)$.

2.4 GMM-based F_{VU} prediction

The spectral fading synthesis presented in the following section 2.6.2 requires a transformed F'_{VU} value, with the operator $'$ indicating a transformation. F'_{VU} is predicted using a modified GMM approach detailed in [17, 25, 26]. The GMM model $\mathcal{M}$ is trained on the voice descriptor set d=[R_d,F_0,$H1$-$H2$,E_{voi},E_{unv}] and the reference value $r = F_{VU}$. $H1$-$H2$ refers to the amplitude difference in dB of the first two harmonic sinusoidal partials. E_{voi} and E_{unv} are the Root-Mean-Square (RMS) based energy measures of the voiced and unvoiced signal parts which will be introduced in the following section. The prediction function

$$F(d) = \sum_{q=1}^{Q} p_q^d d \cdot [\mu_q^r + \Sigma_q^{rd} \Sigma_q^{dd^{-1}} (d - \mu_q^d)] \tag{5}$$

is derived from $\mathcal{M}$ by the definition of equ. 5, with Q=15 being the number of utilized Gaussian mixture components. An initial F_{VU}^p value is predicted from $F(d)$. An error GMM model $\mathcal{M}_{err}$ is trained on the modelling error

$$\epsilon_M = \sqrt[2]{(F_{VU} - F_{VU}^p)^2} \tag{6}$$

serving as reference value r_ϵ=ϵ_M, and on the voice descriptor set d. The transformed descriptor counterpart d' contains the original F_0 contour but transformed values for the remaining voice descriptors: d'=[R'_d,F_0,$H'1$-$H'2$,E'_{voi}, E'_{unv}]. The GMM-based modelling to predict a F'_{VU} contour from the feature sets d and d' is described by:

$$F_{VU_\mu}^p = \mathcal{M}\left(F(d)\right) \tag{7}$$

$$F_{VU_\mu}^{'p} = \mathcal{M}\left(F(d')\right) \tag{8}$$

$$F_{VU_\sigma}^p = \mathcal{M}_{err}\left(F_{err}(d)\right) \tag{9}$$

$$F_{VU_\sigma}^{'p} = \mathcal{M}_{err}\left(F_{err}(d')\right) \tag{10}$$

$$F'_{VU} = F_{VU_\mu}^{'p} + (F_{VU} - F_{VU_\mu}^p) \cdot F_{VU_\sigma}^{'p} / F_{VU_\sigma}^p. \tag{11}$$

Each trained model pair $\mathcal{M}$ and $\mathcal{M}_{err}$ is utilized to predict via their derived prediction functions F and F_{err} the mean prediction value $F_{VU_\mu}^p$ ($F_{VU_\mu}^{'p}$) and the predicted standard deviation $F_{VU_\sigma}^p$ ($F_{VU_\sigma}^{'p}$) from descriptor set d (from the transformed set d'). The true prediction value would equal $F_{VU_\mu}^p$ if no model error occurs: ϵ_M=0. The calculation of F'_{VU} from the transformed d' and the original voice descriptor set d is defined by equ. 11. It evaluates the difference between the original F_{VU} and the predicted $F_{VU_\mu}^p$ value. The difference result is normalized by the ratio of the original and transformed standard deviations $F_{VU_\sigma}^{'p}$ and $F_{VU_\sigma}^p$ of the modelled data distribution, and corrected by the transformed predicted mean value $F_{VU_\mu}^{'p}$.

2.5 Energy modelling

2.5.1 Energy maintenance

A simple RMS measure F_{RMS} evaluates the effective energy value E on the linear amplitude spectrum A_{lin}=$|Y(\omega)|$ of any arbitrary signal spectrum $Y(\omega)$. The RMS energy measures are estimated in PSY as defined in equ. 12:

$$\begin{aligned} F_{RMS}(A_{lin}, k) &= \sqrt{1/K \cdot \Sigma_k \left(A_{lin}(k)^2\right)} \\ E_{sig} &= F_{RMS}(|S(\omega)|) \\ E_{unv} &= F_{RMS}(|U(\omega)|) \\ E_{voi} &= E_{sig} - E_{unv} \end{aligned} \tag{12}$$

E_{sig} and E_{unv} reflect the RMS energies measured on the signal $S(\omega)$ and the unvoiced component $U(\omega)$. E_{voi} is expressed as their difference to represent the RMS energy of the voiced component $V(\omega)$. A transformed $R_d^{'gci}$ contour causes an altered energy value E'_{voi} measured on the transformed voiced part $V'(\omega)$. The high (low) pass filtering applied to $U(\omega)$ ($V(\omega)$) explained in section 2.6.2 generates as well an energy change. The energy re-scaling to the original energy measures defined by equ. 13 ensures that their energy is maintained:

$$\begin{aligned} E_{voi} &= F_{RMS}(|V(\omega)|) \qquad E_{unv} = F_{RMS}(|U(\omega)|) \\ E'_{voi} &= F_{RMS}(|V'(\omega)|) \qquad E'_{unv} = F_{RMS}(|U'(\omega)|) \\ V'(\omega)\cdot &= E_{voi}/E'_{voi} \qquad U'(\omega)\cdot = E_{unv}/E'_{unv} \end{aligned} \tag{13}$$

2.5.2 GMM-based energy prediction

The original voice descriptor set D_E consists of the voice descriptors D_E=[$R_d, F_0, F_{VU}, H1$-$H2$]. The transformed voice descriptor $H'1$-$H'2$ is measured on the magnitude spectrum of $|S'(\omega)|$ in dB. The predicted F'_{VU} value is retrieved from the signal $S'(\omega)$ and the GMM model of section 2.4. The original and not transformed voice descriptor F_0 is added to the energy modelling due to its high correlation with the other voice descriptors. The manually transformed R'_d, the re-estimated $H'1$-$H'2$, the predicted F'_{VU} and the original F_0 descriptors define the transformed voice descriptor set $D'_E = [R'_d, F_0, F'_{VU}, H'1$-$H'2]$. Each energy model receives for training its corresponding reference feature R defined in equ. 11. The energy models $\mathcal{M}^{voi}$ and $\mathcal{M}^{unv}$ are used via their functions F^{voi} and F^{unv}, along with their corresponding error models $\mathcal{M}_{err}^{voi}$ and $\mathcal{M}_{err}^{unv}$ and error functions F_{err}^{voi} and F_{err}^{unv} to predict the RMS-based energy measures E_{voi}^p and E_{unv}^p.

2.6 Synthesis

2.6.1 Time domain mixing

The straight-forward mixing in the time domain adds the synthesized unvoiced stochastic waveform $u(n)$ to the synthesized voiced deterministic waveform $v(n)$. The time domain mixing operates thus full-band without any restriction on the signal bandwidth. It will be evaluated in section 5.1 together with the GMM-based prediction and scaling of the voiced and unvoiced signal energies.

2.6.2 Spectral fading synthesis

The PSY synthesis variant "Spectral fading" is designed to handle voice quality transformations by suppressing possibly occurring artefacts. A short summary discusses here the impact of R_d on the spectral slope required to understand the motivation for the spectral fading synthesis presented in this section. The spectral slope is strongly correlated with R_d. Altering R_d affects the spectral slope. References to an extensive analysis of the spectral correlates of R_d can be found in [14, 15, 20, 27, 28]. A more relaxed voice quality is reflected by higher R_d values and is related to a sinusoidal-like glottal flow derivative which generates higher spectral slopes. A more tense voice quality is parameterized by lower R_d values and relates to an impulse-like glottal flow derivative which produces lower spectral slopes. A lower (higher) spectral slope indicates that more

(less) sinusoidal content can be observed in higher frequency regions. The voice quality transformation to change an original speech recording having a modal voice quality to a more tense voice character has to extend the quasi-harmonic sequence of sinusoidals above the F_{VU}. Contrariwise, a transformation to a more relaxed voice quality needs to reduce the sinusoidal content. A modification of the glottal excitation source required for voice quality transformations implies a F_{VU} modification. The altered F'_{VU} frequency has to be naturally represented by properly joining the voiced $V(\omega)$ and unvoiced $U(\omega)$ signal components. The transformation of the original R_d^{gci} contour used to extract $C(\omega)$ introduces an energy variation in the re-synthesis of a transformed $V'(\omega)$. However, even with the energy maintenance of section 2.5 the alteration of a modal to a very tense voice quality may result into sinusoidal content being of higher energy than the noise part at F_{nyq}. This sets $F'_{VU} = F_{nyq}$ and causes audible artefacts. F'_{VU} is therefore predicted using the method described in section 2.4. Additionally, the spectral fading method employs two spectral filters to cross fade $V(\omega)$ and $U(\omega)$ around F'_{VU}. The spectral band around F_{VU} is comprised of a mix of both deterministic $V(\omega)$ and stochastic $U(\omega)$ signal content. A low pass filter P_L fades out the voiced part $V(\omega)$ and a high pass filter P_H fades in the unvoiced part $U(\omega)$ with increasing frequency. The linear ramps with a slope of m_{LP}=-96 dB and m_{HP}=-48 dB per octave define the steepness of both filters. A higher value is chosen for m_{LP} since the F'_{VU} prediction may be very high for very tense voice qualities. A less steep fade out filter would not be effective enough.

3. VOICE QUALITY TRANSFORMATION

The study of [29] on the Just Noticable Differences (JND) of human auditory perception reports that changes in higher (lower) value regions of the Open Quotient OQ (the asymmetry coefficient α_m) require longer distances of ΔOQ ($\Delta\alpha_m$) to arise the sensation of a voice quality change in the perception of a listener. We spread according to that experimental results the original R_d^{gci} contour into several $R_d^{'gci}$ contours with positive and negative offsets covering the complete R_d range such that lower ΔR_d steps are placed in lower and higher ΔR_d steps in higher R_d value regions. One example is illustrated in fig. 1 on the phrase employed for the evaluation in section 5. Table 1 shows the mean R_d^{μ} values of the original R_d contour with index 0, and respectively 3 positive and 3 negative μ values for each voice quality change. $R_d^{\sigma^2}$ lists their variance σ^2. It increases with increasing R_d to reflect the objective of having to apply higher ΔR_d steps with higher R_d values. The R_d mean difference column ΔR_d^{μ} reflects the mean ΔR_d steps measured between each row index on the R_d^{μ} values to show that also the mean difference increases with increasing R_d^{μ} from a tense to a relaxed voice quality.

4. BASELINE METHOD SVLN

The method called "**S**eparation of the **V**ocal tract with the **L**iljencrants-Fant model plus **N**oise" detailed in [9, 30, 31]

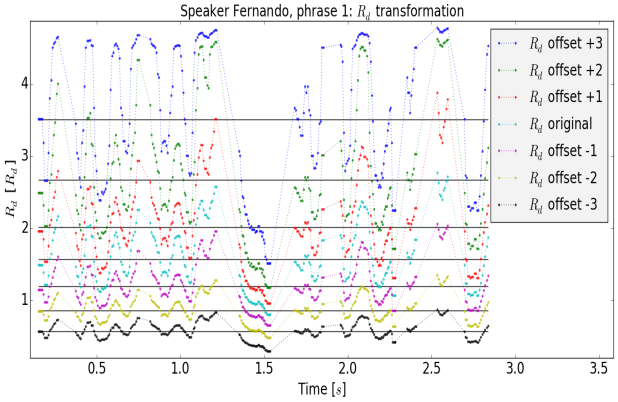


Figure 1: *Generated $R_d^{'gci}$ contour examples*

Voice quality (index)	R_d^{μ}	$R_d^{\sigma^2}$	ΔR_d^{μ}
Very relaxed (+3)	3.5109	0.9031	-0.8397
Relaxed (+2)	2.6711	0.7825	-0.6597
Modal to relaxed (+1)	2.0114	0.3631	-0.4442
Modal (original) (0)	1.5673	0.1937	
Tense to modal (-1)	1.1936	0.0941	-0.3737
Tense (-2)	0.8601	0.0341	-0.3335
Very tense (-3)	0.5704	0.0154	-0.2898

Table 1: *R_d mean, variance and mean difference values*

represents the baseline method on whose means the proposed system PaReSy is build upon. The main differences are the VTF representation, the energy model and the estimation of the stochastic component. SVLN constructs the latter by high pass filtering white noise, applying an amplitude modulation parameterized by the glottal pulse sequence, and cross fading between consecutive synthesized noise segments. The gain σ_g measures the energy level at F_{VU} while analysis to control the stochastic energy at the synthesis step. SVLN synthesizes glottal pulses with the LF model in the spectral domain to extract $C(\omega)$ below F_{VU}. The VTF above F_{VU} is taken from the signals spectral envelope. SVLN facilitates advanced pitch transposition or voice quality transformations while maintaining a high synthesis quality [9, 32].

5. EVALUATION

The evaluation section presents the results of two listening tests conducted on natural human speech of the Hispanic speaker "Fernando" speaking French. The voice quality assessment examines how well both synthesis systems are able to produce different voice quality characteristics. Test participants were asked to rate different synthesized voice qualities according to the same indices as in table 1. Each phrase is rated as well on their synthesis quality according to the Mean Opinion Scale (MOS).

The baseline method SVLN of section 4 and the proposed method PSY of section 2 received the same features R_d^{gci}, F_0 and F_{VU} as pre-estimated input to analyze their corresponding VTF $C(\omega)$. Please note that SVLN requires to smooth the voice descriptor contours. Due to the energy measure at F_{VU} it cannot handle value changes varying too

quickly in short-time segments [30]. For this test a median smoothing filter covering 100 ms was applied.

5.1 Manual R_d offsets and time domain mixing

A preliminary listening test has been conducted by 6 sound processing experts internally in the laboratory. The listening test is available online via: Manual offset test [1].

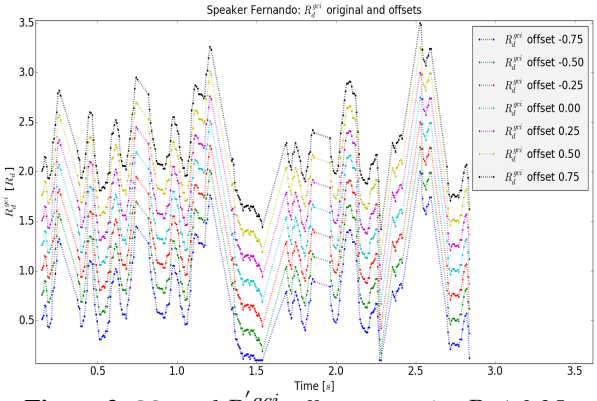


Figure 2: *Manual $R_d^{'gci}$ offsets, step size R_d±0.25*

Fig. 2 depicts the original R_d^{gci} contour in the middle shown in cyan colour, and six additional $R_d^{'gci}$ contours. Each positive and negative mean offset constitutes an empirically determined R_d offset of R_d±0.25 to the previous contour in its respective direction. The offset amount was chosen such that an R_d^{gci} offset contour reaches an R_d range border [0.1 5.0]. In this example the R_d^{gci} offset -0.75 saturates around ∼1.50 seconds on the lower R_d border.

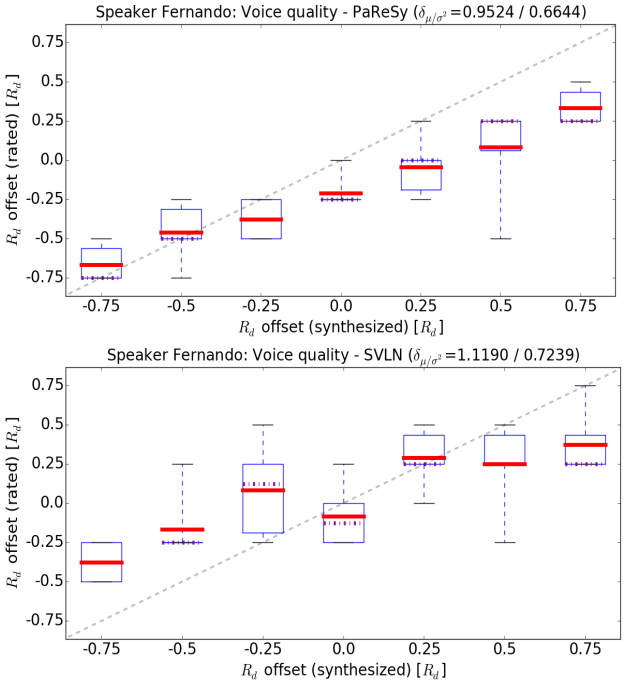


Figure 3: *Voice quality ratings - TD mixing*

Fig. 3 depicts the voice quality ratings for both speech systems. The horizontal grey lines at both ends (whiskers) are set to show the minimum and maximum value for each evaluation. The horizontal red (violet) lines reflect the mean (median) voice quality ratings of all participants per test phrase. The dialog grey dashed line exemplifies their ideal placement if each test participant would have been able to perceptually associate each synthesized voice quality example to its corresponding voice quality characteristic. The mean deviation value δ_μ=0.95 for PSY expresses the disagreement of the listeners, being ideally δ_μ=0.00. PSY received very low mean deviation δ_μ values for more tense voice qualities. The stronger the original modal voice quality is transformed towards a more relaxed voice quality the less well could the participants identify its perceptual sensation. Drawing a regression line through each mean value shown in red horizontal lines per rated R_d offset would result in a less step line than the ideal one depicted as grey dashed line. A higher mean deviation value δ_μ=1.12 as compared to PSY is shown for the baseline method SVLN in fig. 3. It indicates that the listeners could less well capture the different synthesized voice qualities and associate them with the corresponding offset indices. Clear voice quality associations can be concluded for both systems.

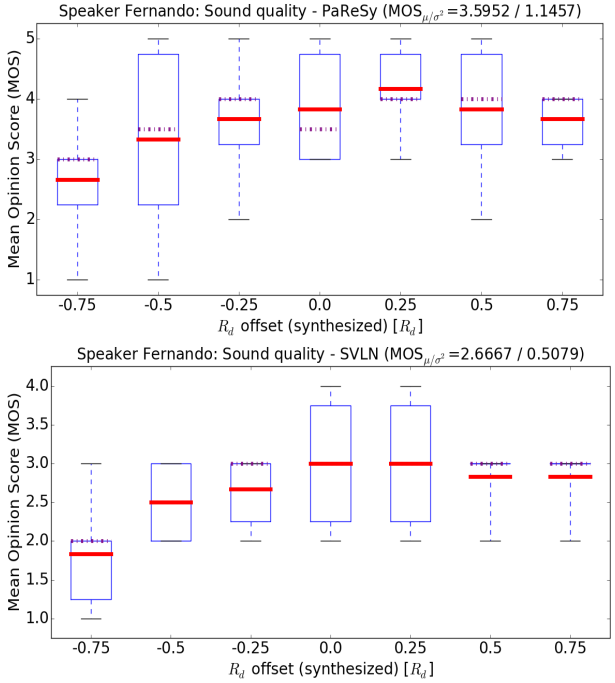


Figure 4: *MOS synthesis quality ratings - TD mixing*

The MOS synthesis quality result are shown in fig. 4. PSY exhibits partially highest ratings up to an excellent synthesis quality of 5 for all but the very tense and very relaxed voice quality characteristics with the R_d offsets ±0.75. Contrariwise, the voice qualities very tense and tense are partially rated with the lowest MOS synthesis quality poor. The mean synthesis quality MOS$_\mu$=2.67 of SVLN is comparably lower than MOS$_\mu$=3.60 for PSY. The very tense voice quality of SVLN received comparably lower MOS ratings than its other synthesized R_d offsets. Stronger voice quality changes are assessed with less good MOS synthesis qualities for both systems. PSY received in general a lower deviation from the true voice quality rating and a higher MOS synthesis quality compared to SVLN.
Fig. 5 illustrates the voice quality and the MOS synthesis quality ratings for the PSY synthesis variant using an

[1] Speaker Fernando: http://stefan.huber.rocks/phd/tests/RdMisterF/

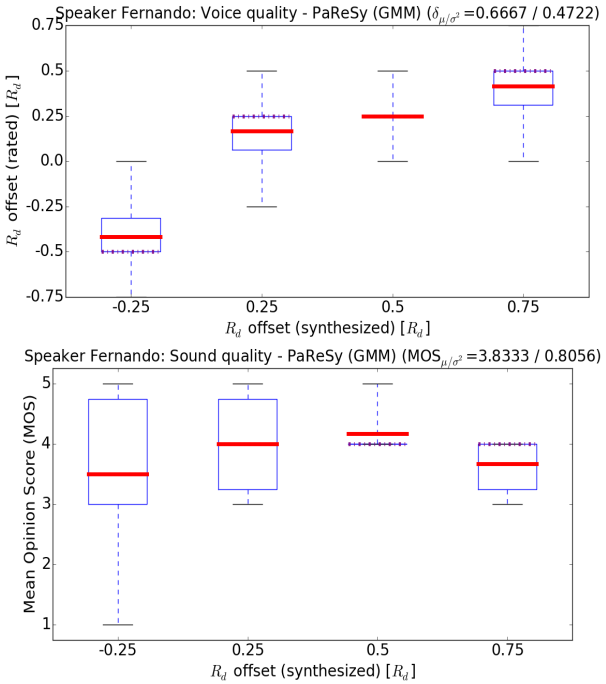


Figure 5: *Test results - PSY energy scaling*

additional energy scaling. The voiced $V(\omega)$ and unvoiced $U(\omega)$ component are scaled by the respective RMS energies predicted from a dedicated GMM energy model for each part. Please note that the two R_d offsets -0.75 for a very tense and -0.50 for a tense voice quality had to be excluded from the test for PSY (GMM). The predicted RMS energy contours resulted into amplitudes in the time domain being outside the valid range [-1 1]. In general it can be observed that the GMM-based energy scaling of PSY received roughly similar voice and MOS synthesis quality ratings as the standard PSY method. This suggests that the GMM predicted energy contours for the voiced $V(\omega)$ and unvoiced $U(\omega)$ parts do neither increase nor decrease the synthesis quality and the voice quality characteristic to a significant extent.

Method	Δ VQ$_\mu$	Δ VQ$_{\sigma^2}$	MOS$_\mu$	MOS$_{\sigma^2}$
PSY	0.9524	0.6644	3.5952	1.1457
PSY (GMM)	0.6667	0.4722	3.8333	0.8056
SVLN	1.1190	0.7239	2.6667	0.5079

Table 2: *Voice quality (VQ) and MOS sound quality*

Table 2 summarizes the mean deviation Δ VQ$_\mu$ and its variance Δ VQ$_{\sigma^2}$ from the optimal voice quality rating in the first two columns. The corresponding mean and variance of the MOS sound quality ratings are listed in the last two columns. The three synthesis approaches PSY time domain mixing in the first row, PSY time domain mixing using the additional GMM energy scaling of section 2.5.2, and the baseline method SVLN are compared. The lower VQ and higher MOS values for PSY (GMM) are partially a result of having omitted the two voice quality transformations towards a tense and very tense voice quality. The expectation for these two omitted test cases is that they would have decreased the good test results for PSY (GMM).

5.2 Transformed $R_d^{'gci}$ contours and spectral fading

The PSY spectral fading synthesis variant presented in 2.6.2 requires the F_{VU} prediction of section 2.4. An example is depicted in fig. 6. The transformed $R_d^{'gci}$ contours and the

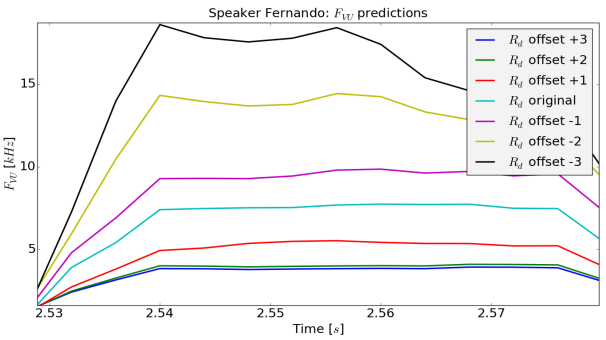


Figure 6: F_{VU} *prediction excerpt for PSY synthesis*

original R_d^{gci} contour of fig. 1 were employed by both systems for synthesis. Following the voice production model of equ. 2.1, a transformed glottal pulse $G'(\omega)$ leads to a transformed reconstructed signal $S'(\omega)$. The unvoiced component $U(\omega)$ remains unmodified. 11 participants rated each speech phrase by SVLN and PSY. Please note that the PSY energy prediction variant is due to the too huge scaling for tense voice qualities omitted. The listening test is available online via: Transformed $R_d^{'gci}$ test [2].

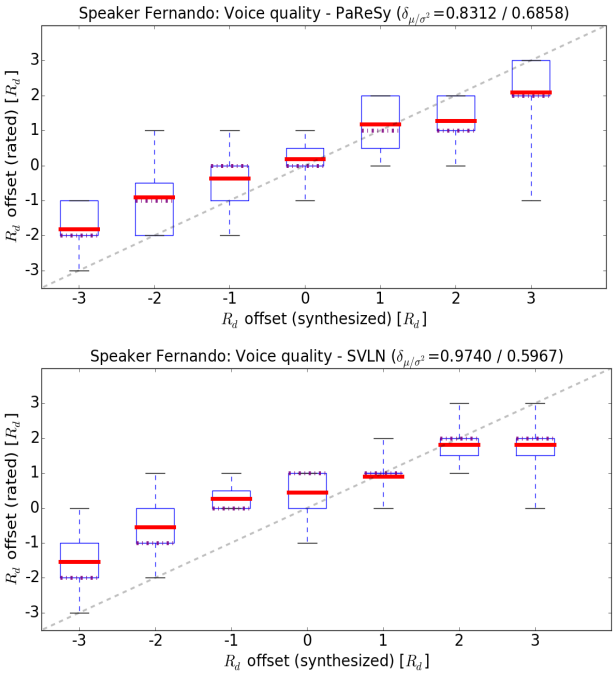


Figure 7: *Voice quality ratings - Spectral fading*

Fig. 7 shows again the voice quality ratings for both speech systems. The mean deviation value δ_μ=0.83 for PSY is lower than the corresponding δ_μ=0.97 for the SVLN. Clear voice quality associations can be concluded for both systems following closely the ideal dashed line. The deviations increase with higher transformations.

[2] Speaker Fernando: http://stefan.huber.rocks/phd/tests/vqMisterF/

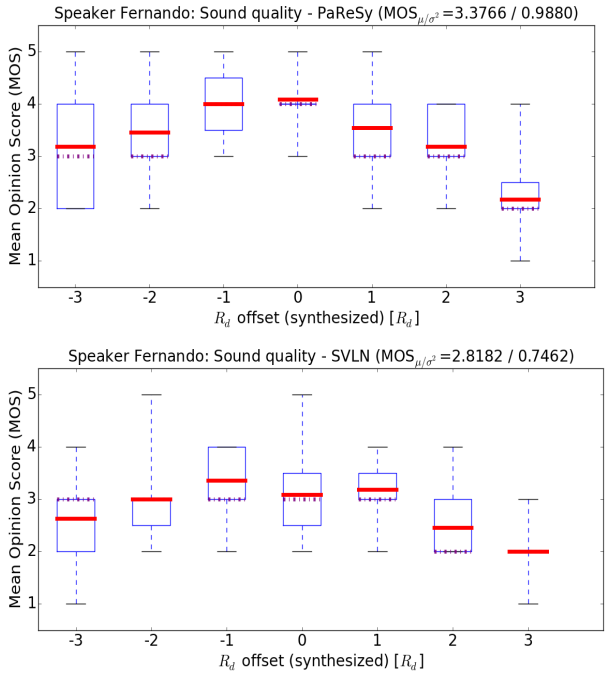


Figure 8: *MOS synthesis quality ratings - Spectral fading*

The MOS synthesis quality evaluation for PSY shown in fig. 8 exhibits partially highest ratings up to an excellent synthesis quality of 5 for all but the "relaxed" and "very relaxed" voice quality characteristics with index +2 and +3. The evaluated mean synthesis quality MOS$_\mu$=2.82 of SVLN is comparably lower than MOS$_\mu$=3.38 for PSY. Stronger voice quality changes are assessed with less good MOS synthesis qualities for both systems. PSY received in general a lower deviation from the true voice quality rating and a higher MOS synthesis quality related to the baseline method SVLN, shown in table 3.

Method	Δ VQ$_\mu$	Δ VQ$_{\sigma^2}$	MOS$_\mu$	MOS$_{\sigma^2}$
PSY	0.8312	0.6858	3.3766	0.9880
SVLN	0.9740	0.5967	2.8182	0.7462

Table 3: *Voice quality (VQ) and MOS sound quality*

6. CONCLUSIONS

The findings presented with the subjective listening test of section 5 suggest that the proposed novel speech analysis and synthesis system PSY is able to analyze an input speech phrase such that different re-synthesized versions carry the perception of different voice quality characteristics. Its assessed synthesis quality received partially very good judgements for minor changes in voice quality. Major voice quality changes are appraised of moderate quality for both the baseline and the proposed method. However, further work is required to render the GMM energy prediction applicable for all cases. Please note that the proposed speech framework will be integrated as system to synthesize singing voices within the ANR project ChaNTeR [3].

[3] ChaNTeR: anasynth.ircam.fr/home/projects/anr-project-chanter/

Acknowledgments

The main author was financed by a CIFRE contract as a former collaboration between the research institute IRCAM and the company Acapela Group. Currently he is financed by a grant from the ANR Project ChaNTeR to enhance the proposed system for singing voices synthesis. He is very grateful for the kind attendance by his supervisor Dr. Axel Röbel and the support from the Acapela Group.

7. REFERENCES

[1] D. G. Childers and C. K. Lee, "Vocal quality factors: analysis, synthesis, and perception." *Journal of the Acoustical Society of America*, vol. 90, no. 5, pp. 2394–410, 1991.

[2] J.-S. Liénard and C. Barras, "Fine-grain voice strength estimation from vowel spectral cues," in *14th Annual Conference of the Int. Speech Communication Association (Interspeech ISCA)*, Lyon, France, 2013, pp. 128–132.

[3] J. D. M. Laver, *The Phonetic Description of Voice Quality*. Cambridge University Press, 1980, vol. 31.

[4] J. Kane, S. Scherer, M. Aylett, L.-P. Morency, and C. Gobl, "Speaker and language independent voice quality classification applied to unlabeled corpora of expressive speech," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

[5] T. Drugman and T. Dutoit, "The deterministic plus stochastic model of the residual signal and its applications," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 968–981, 2012.

[6] J. P. Cabral and J. Carson-Berndsen, "Towards a better representation of the envelope modulation of aspiration noise," in *Advances in Nonlinear Speech Processing*, ser. Lecture Notes in Computer Science, T. Drugman and T. Dutoit, Eds. Springer Berlin Heidelberg, 2013, vol. 7911, pp. 67–74. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38847-7_9

[7] D. Vincent, O. Rosec, and T. Chonavel, "A new method for speech synthesis and transformation based on an arx-lf source-filter decomposition and hnm modeling," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, USA, 2007, p. 525–528.

[8] J. Cabral, S. Renals, K. Richmond, and J. Yamagishi, "Glottal spectral separation for parametric speech synthesis," in *9th Annual Conference of the Int. Speech Communication Association (Interspeech ISCA)*, Brisbane, Australia, September 2008, pp. 1829–1832.

[9] G. Degottex, P. Lanchantin, A. Röbel, and X. Rodet, "Mixed source model and its adapted vocal tract filter estimate for voice transformation and synthesis," *Speech Communication*, vol. 55, no. 2, pp. 278–294, 2013.

[10] X. Serra, *Musical Sound Modeling with Sinusoids plus Noise*. Swets and Zeitlinger, 1997, pp. 91–122. [Online]. Available: files/publications/MSM-1997-Xserra.pdf

[11] G. Fant, "The source filter concept in voice production," *Quaterly Progress and Status Report, Department of Speech, Music and Hearing, KTH, Stockholm, Sweden*, vol. 22, no. 1, pp. 021–037, 1981.

[12] R. Maia and Y. Stylianou, "Complex cepstrum factorization for statistical parametric synthesis," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2014, pp. 3839–3843.

[13] G. Fant, J. Liljencrants, and Q.-G. Lin, "A four-parameter model of glottal flow," *Quaterly Progress and Status Report, Department of Speech, Music and Hearing, KTH, Stockholm, Sweden*, vol. 26, no. 4, pp. 001–013, 1985.

[14] G. Fant, "The lf-model revisited. transformation and frequency domain analysis," *Quaterly Progress and Status Report, Department of Speech, Music and Hearing, KTH, Stockholm, Sweden*, vol. 36, no. 2-3, pp. 119–156, 1995.

[15] G. Fant, "The voice source in connected speech," *Speech Communication*, vol. 22, no. 2-3, pp. 125–139, 1997.

[16] S. Huber, A. Röbel, and G. Degottex, "Glottal source shape parameter estimation using phase minimization variants," in *13th Annual Conference of the Int. Speech Communication Association (Interspeech ISCA)*, ser. 1990-9772, Portland, Oregon, USA, 2012, pp. 1644–1647.

[17] S. Huber and A. Röbel, "On the use of voice descriptors for glottal source shape parameter estimation," *Computer, Speech, & Language*, vol. 28, no. 5, pp. 1170 – 1194, 2014.

[18] G. Degottex, A. Röbel, and X. Rodet, "Joint estimate of shape and time-synchronization of a glottal source model by phase flatness," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, USA, 2010, pp. 5058–5061.

[19] A. Röbel, F. Villavicencio, and X. Rodet, "On cepstral and all-pole based spectral envelope modelling with unknown model order," *Elsevier, Pattern Recognition Letters*, vol. 28, no. 11, pp. 1343 – 1350, 2007.

[20] C. d'Alessandro, B. Bozkurt, B. Doval, T. Dutoit, N. Henrich, V. Tuan, and N. Sturmel, "Phase-based methods for voice source analysis." in *Advances in Nonlinear Speech Processing*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4885, pp. 1–27.

[21] M. Zivanovic and A. Röbel, "Adaptive threshold determination for spectral peak classification," *Computer Music Journal*, vol. 32, no. 2, pp. 57–67, 2008.

[22] C. d'Alessandro, V. Darsinos, and B. Yegnanarayana, "Effectiveness of a periodic and aperiodic decomposition method for analysis of voice sources," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 1, pp. 12–23, 1998.

[23] Y. Pantazis and Y. Stylianou, "Improving the modeling of the noise part in the harmonic plus noise model of speech," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008, pp. 4609–4612.

[24] T. Drugman and Y. Stylianou, "Maximum voiced frequency estimation: Exploiting amplitude and phase spectra," *Signal Processing Letters, IEEE*, vol. 21, no. 10, pp. 1230–1234, Oct 2014.

[25] P. Lanchantin and X. Rodet, "Dynamic model selection for spectral voice conversion," in *11th Annual Conference of the Int. Speech Communication Association (Interspeech ISCA)*, Makuhari, Chiba, Japan, 2010, pp. 1720–1723.

[26] P. Lanchantin and X. Rodet, "Objective evaluation of the dynamic model selection method for spectral voice conversion," in *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 5132–5135.

[27] B. Doval and C. d'Alessandro, "The spectrum of glottal flow models," Laboratoire d'informatique pour la mécanique et les sciences de l'ingénieur (Orsay), Orsay, Tech. Rep. LIMSI 99-07, 1999.

[28] B. Doval, C. d'Alessandro, and N. Henrich, "The spectrum of glottal flow models," *Acta Acustica united with Acustica*, vol. 92, no. 6, pp. 1026–1046, 2006.

[29] N. Henrich, C. d'Alessandro, B. Doval, M. Castellengo, G. Sundin, and D.Ambroise, "Just noticeable differences of open quotient and asymmetry coefficient in singing voice," *Journal of Voice*, vol. 17, no. 4, pp. 481–494, 2003.

[30] G. Degottex, "Glottal source and vocal tract separation," Ph.D. dissertation, Institut de Recherche et Coordination Acoustique/Musique (IRCAM), Université de Pierre et Marie Curie (UPMC), Université de Paris 6, Paris, France, 2010.

[31] G. Degottex, A. Röbel, and X. Rodet, "Pitch transposition and breathiness modification using a glottal source model and its adapted vocal-tract filter," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 5128–5131.

[32] P. Lanchantin, G. Degottex, and X. Rodet, "A hmm-based speech synthesis system using a new glottal source and vocal-tract separation method," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, USA, 2010, pp. 4630–4633.

"VIRTUAL TETTIX" : CICADAS' SOUND ANALYSIS AND MODELING AT PLATO'S ACADEMY

Anastasia Georgaki
Music Department
University of Athens, Greece
georgaki@music.uoa.gr

Marcelo Queiroz
Computer Science Department
University of São Paulo, Brazil
mqz@ime.usp.br

ABSTRACT

This paper deals with the acoustic analysis of timbral and rhythmic patterns of the Cicada Orni sound activity, collected at the Plato Academy archaeological site during the summer period of 2014, comprising the Tettix soundscape database.

The main purpose here is to use sound analysis for understanding the basic patterns of cicada calls and shrilling sounds, and subsequently use the raw material provided by the Tettix database in a statistical modeling framework for creating virtual sounds of cicadas, allowing the control of synthesis parameters spanning micro, meso and macro temporal levels.

1. INTRODUCTION

The Plato Academy soundscape database has been collected in the context of the TETTIX project and consists of several sound recordings of cicadas, singing both individually and collectively, which were taken at the Gymnasium portion of the archaeological site of Plato's Academy (Athens) during the period of July through September 2014. Important philosophical and poetic references deal with the diachronic aesthetic value of this particular insect in Ancient Greek mythology, as also in Classical, Byzantine and Modern Greek literature (including Hesiod, Anacreon [1], Aesop, Homer, Plato, Aristotle, Thucydides, Ritsos, Elytis, Seferis); this discussion is part of an interdisciplinary research domain that goes under the label of tettigology [1], where entomologists, sociologists, cultural anthropologists and composers try to investigate the strange sonic events and chorusing of cicadas in different geographical areas during summer [2, 3].

This research, which deals with the rhythmic and timbral analysis of the Cicada Orni, has been inspired by the stochastic model of cicadas proposed by Iannis Xenakis. More precisely, Iannis Xenakis has continuously referred to the sonic texture and stochastic effect of the cicadas in order to describe the granular clouds of sounds he experienced during the war [4] [2]. Within this context, Xenakis underlines the fact that the statistical characterization of certain sonic events (e.g., a demonstration crowd or shouting guns) can be very similar when separated from their political or moral context, and when applied to the cicada chorus, this fact indicates the passage from total order to total disorder [6].

Many other composers have also been inspired by the cicada soundscape in different ways: Bartók evokes the cicada in his piano suite of 1926 (*Szabadban* or Out of Doors), Ligeti evokes the stochastic behavior of cicadas in his *Poème Symphonique* (1962) for 100 metronomes, Luc Ferrari and J. C. Risset use elaborated cicada sounds in their electroacoustic music works, and David Rothenberg tries to interact with the cicada choruses by playing music in nature [5].

Previous work is related typically to either isolated analysis or synthesis of cicada singing, and include chaos-theoretical models for Asian Cicadas [6], physical modeling of the sound-producing mechanism of the cicadas [7], assessing the correlation between sound level activity and size of cicada populations [8] and measuring the mechanical response in the female cicada *tympani* [9]. In counterpoint to these works, we propose here a three-layered hierarchical analysis-synthesis model for cicada sounds based on Gaussian and Markov statistical models.

The TETTIX project [3] is primarily focused on acoustic and musical issues associated to the collected signals, such as spatial distribution, diversity of timbres and rhythms, antiphonal and choral organizations, among others. These issues require musical analysis, statistical analysis and both analytic and synthetic methods of signal processing, which are used to create representation models and manipulation tools that will allow a thorough exploration of the artistic potential of timbres and rhythms associated to this soundscape.

[1] Tettigology (from the Ancient Greek word Τέττιξ, or Tettix, for cicada) has its roots in several ancient civilizations, from China to Greece; more specifically in Ancient Greek culture, the cicada has appeared on diverse fields, such as literature, visual arts, folklore, scientific writing, philosophy and religion. The cicada life cycle and its characteristic song has served as inspiration for many philosophers, poets and musicians of Ancient Greece.

Copyright: ©2015 Anastasia Georgaki 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.

[2] Xenakis was the first composer to use stochastic distributions as a tool for creating a music (as he writes in his book Formalized Music) which was similar to the sound of cicadas, or to that of rain falling on a tin roof.

[3] The TETTIX project deals with the cartography of cicadas singing in Greece and with the acoustic analysis of the soundscape in archaeological areas; evolutionary models of the cicada singing in different areas of Athens have been designed; this material has been used in artistic explorations of the cicada choruses and in sound installations.

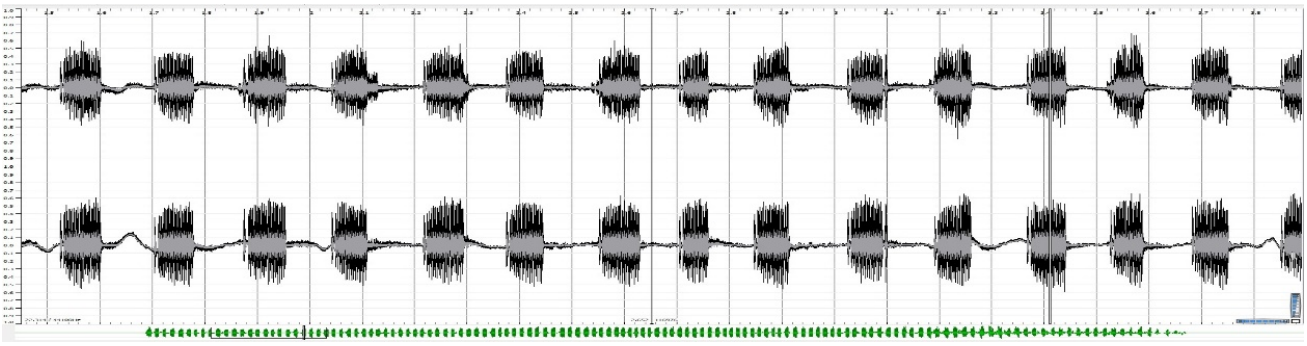


Figure 1. Echemes and interechemes of the Platonic Cicada Orni.

1.1 From the echeme to the Cicada choir

The sound-producing mechanism of the cicadas [4] has been studied since the 18th century, but only after sound recordings and electronic means in the 1950's the exact function of the morphological structures during cicada singing has been established [10–12]. This mechanism consists in a pair of membranes called *Tymbales* which are excited by the action of muscles with the same name, often in alternated motion; the sound thus produced is amplified by air bags positioned directly under the Tymbales. When this mechanism is set into motion the cicada emits its typical sound, which is referred to here as an echeme [5]. Figure 1 shows an example of a series of short echemes.

The sound produced by the cicada has several functions, such as

- a danger alarm for other cicadas,
- a defense mechanism from predators,
- calling and mating with other cicadas, and
- establishing territories,

which explains the emergence of a large number of rhythmic and timbral patterns easily discernible by the human ear [13–16].

The TETTIX project collected several examples of such patterns through sound recordings made in the archaeological site of Plato's Academy, in Athens, during the summer of 2014 (July, August and September). All these recordings were labeled according to diverse conditions that affected the sound material, such as the date and time, temperature, whether it was an individual or collective emission, and also the type of tree where the cicadas were. This material was analyzed according to their temporal macrostructure (density and duration of emissions, and intervals between *echemes*), and to their spectral distribution (spectral centroid, bandwidth and quartiles), to be used as sound material in the context of granular synthesis and swarm models [2].

Future plans for artistic use of this sound material in compositions and installations are coupled with the timbral and rhythmic modeling described in this paper. Timbral modeling is accomplished with parametric sound synthesis models, whereas rhythmic modeling is approached with statistical models, as discussed below.

The proposed modeling of timbre in Plato's Academy soundscape extends on previous studies of this type of signal [7, 11, 14] and consists of three levels of representation. At the lowest level, it provides statistical modeling of spectra in quasi-periodic portions of the signal, aiming at representing observed fragments at a micro-temporal scale (where quasi-periodicity holds), and providing a Gaussian spectral model and an amplitude-modulation model as functions of the signal type (according to date, time, temperature, number of cicadas and tree type).

Rhythmic modeling extends the formerly conducted statistical analysis, establishing a second-order Markov model. Taken together, these models comprise a three-level hierarchical statistical synthesis model. The upper level deals with macro-temporal transitions between different types of emissions, the intermediate level deals with meso-temporal subtler variations within a given emission regime, and the lower level represents the dynamically evolving spectra of the cicada song.

Timbral and rhythmic statistical modeling are viewed as low-level representations for the creation of synthetic instruments here referred to as virtual cicadas. The use of virtual cicadas in a musical context will involve also the consideration of musical issues dealing with rhythm and polyphony, and specifically to phrasing and choral structures that can be observed in the recordings.

Two different categories of sounds can be identified in the emissions of the Cicada Orni species: a continuous component (i.e. a long echeme or emission) and a broken component (separate echemes or emissions). These emissions may be viewed as phrases, which can be either simple and repetitive (monotonous or alternating short and long emissions) or syncopated (i.e. having a more complex rhythmic structure).

Another very important aspect of the songs of cicadas are choral relationships [16]. Concerning the choral organization in most cicadas species two main categories of interaction are to be found: synchrony and alteration. Synchronicity, which was formerly viewed as a cooperative mechanism among males, never seems to be perfect, and

[4] Technically only the male cicada has this mechanism, while female cicadas have corresponding sound receivers (tympani).

[5] An echeme is essentially an uninterrupted burst of sound, which can be as short as a tick or a click or it may continue for a longer period. When an echeme is viewed at an expanded time scale the separate pulses created by the buckling of the tymbale ribs can be seen.

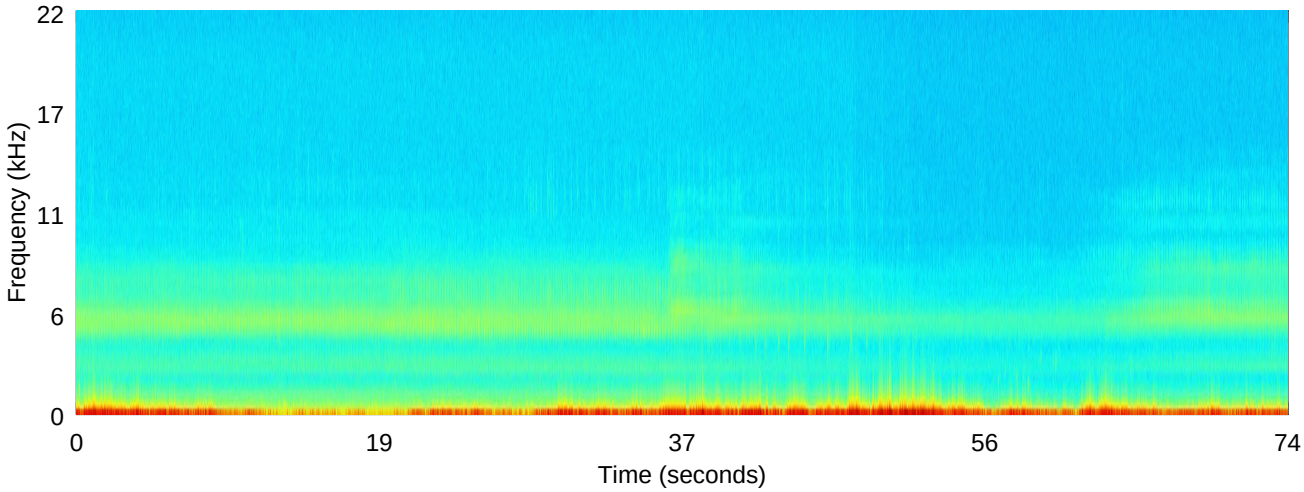


Figure 2. Spectrogram of a sample recording from the TETTIX database made on 24/7/2014. Colors (from blue to red) represent energy on a log scale.

this is related to the psychoacoustic precedence effect [6] . In cicadas, female phonotaxis is influenced by the precedence effect in the sense that the first of two or more closely synchronized calls is preferred. Thus, males are selected to adopt a timing mechanism of signal jamming activities, averting following calls in a synchronizing or alternating fashion [16]. In choral emissions we can also discriminate the domino effect and the effect of the last word [14].

2. TIMBRE AND RHYTHMIC MODELS

Timbre and rhythm in cicada singing are here represented in a three-layer hierarchical model, which considers the dynamic evolution of spectra on a micro-temporal scale (few milliseconds), the perceived fast-evolving fluctuations in amplitude (akin to tremolos) on a meso-temporal scale (on the order of a second), and longer-term evolutions, such as beginnings and ends of echemes and also variations in intensity and number of cicadas, on a macro-temporal scale (many seconds or minutes). As a running example we will use one particular recording, made on July 27th 2014, whose spectrogram is presented in Figure 2. It is sampled at 44100 Hz, and is windowed in frames of 2048 samples or approximately 46 milliseconds.

2.1 First layer: micro-temporal model

One distinguishing aspect of the spectrogram of Figure 2, which is not uncommon in several other recordings, is the presence of strong low-frequency components, corresponding to the red and yellow fringe at the bottom of the figure. These are possibly due to effects of the wind and other environmental sounds during the recording, and hide the real cicada call that requires analysis and modeling, which corresponds more closely to the greenish band around 6 kHz. The presence of these effects has an evident impact on the time-domain representation of the signal, as illustrated in the upper-left graph in Figure 3.

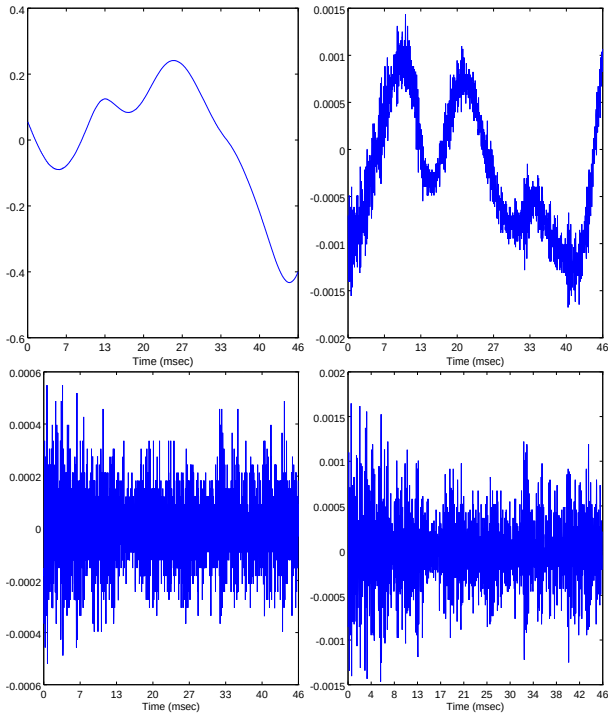


Figure 3. A windowed portion of the signal from Figure 2, 46 ms long (upper-left), a first high-passed filtered version (upper-right), the twice-filtered version (lower-left) and the corrected low-pass filtered version (lower-right) containing predominantly the cicada song.

We see a slowly-varying, apparently smooth profile, which correspond to very low frequencies (up to 60 Hz). The signal of interest lies hidden in much subtler fluctuations around this apparently smooth profile. By filtering once (Figure 3, upper-right) and twice with a high-pass (difference) filter (Figure 3, lower-left) the actual content that requires modeling starts to emerge. As it turns out, a further low-pass filtering step is required to deemphasize higher-

[6] The precedence effect describes a phenomenon by which two sounds arriving from different directions are perceived as a single auditory event, whose spatial location is determined mainly by the location of the leading sound.

frequency components that were distorted by the two-zero high-pass filter, producing the last graph in Figure 3 (lower-right).

On the spectral domain (Figure 4), we can see the discrepancy between the lower frequencies in the original signal (Figure 4 upper-left, spectrum starting at 0 kHz) and the higher frequencies (Figure 4 upper-right, same spectrum starting at 1 kHz). The twice low-pass filtered signal has the spectrum shown in Figure 4 lower-left, which illustrates the difference filter effect of distorting the higher-frequency components in comparison with the frequency range of interest (around 6 kHz). This is further compensated by a two-zero low-pass (averaging) filter, which produces the spectrum of Figure 4 (right), closer to the original spectrum in that range (Figure 4 lower-right), and that corresponds on the time-domain to the signal displayed in the lower-right graph of Figure 3.

This result, obtained by the application of simple filters (or equivalently a single four-zero pass-band filter), is preferred over the application of an ideal pass-band filter for several reasons, among them the lack of knowledge of the precise boundaries of the frequency range of interest, and the rippling (Gibbs) effects and spectral leakage that would be introduced by the use of a much sharper filter.

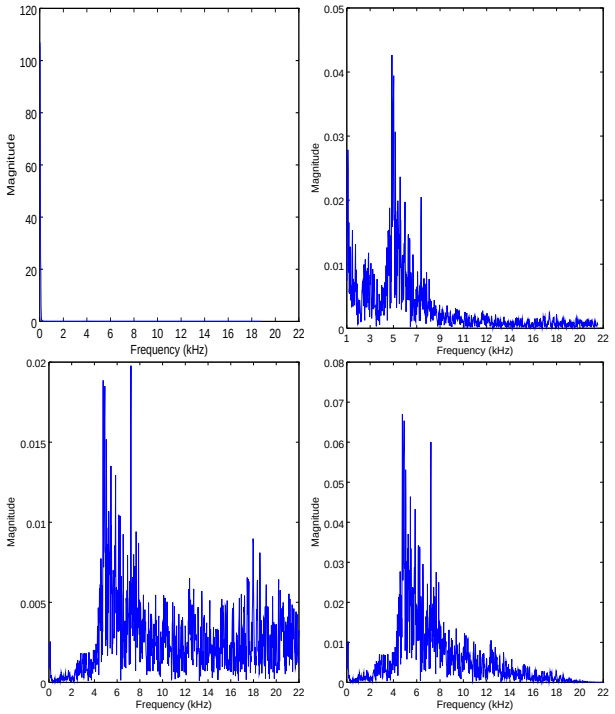


Figure 4. Spectra of the original window (upper-left), same spectrum starting at 1 kHz (upper-right), twice high-pass filtered signal (lower-left) and final low-pass filtered version (lower-right).

The next step in micro-temporal analysis is considering a sliding window along a relatively stable portion of the signal, to capture the dynamic aspects of the spectra observed. In Figure 5 (left) we see the same window which has been treated by the four-zero filter, and the average of the next 1000 consecutive windows, representing the average spectral pattern for 46 seconds of a steady cicada call (Figure 5, center). For each frequency, the variance of the energy around these mean values, as the window slides through these 46 seconds, is represented in the right-most graph of Figure 5. It can be seen that a somewhat similar profile is observed in the variance spectrum, meaning the higher the amplitudes of the spectral components the higher also the variance observed in consecutive windows, although variances are significantly smaller outside the range between 4 kHz and 10 kHz.

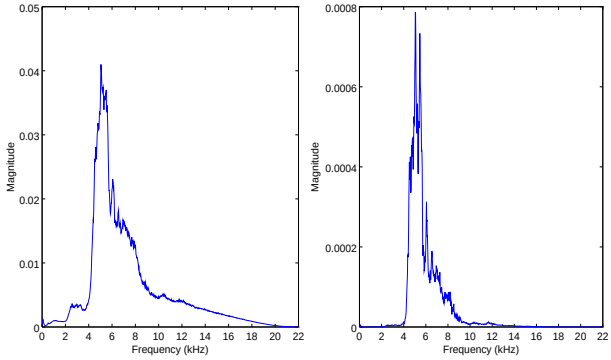


Figure 5. Average spectrum of 1000 consecutive windows (46 seconds) of a steady emission (center) and variance spectrum for the same windows (right).

These average and variance spectra are the input to a synthesis engine that produces virtual cicada timbres that serve as input for the subsequent meso and macro-temporal refining synthesis stages.

2.2 Second layer: meso-temporal model

Moving on to the meso-temporal scale, we want to model the amplitude fluctuations that are perceived as shrilling, an effect somewhat similar to tremolo in the sense that it can be viewed as a form of amplitude envelope applied to a steadier signal, although in the cicada call these are by no means periodic or quasi-periodic fluctuations. As opposed to the micro-temporal modeling, here these relatively smooth variations are the relevant part to be modeled, as an amplitude envelope that is going to be applied to the micro-temporal synthesis engine.

At this scale the superposition of undesired low-frequency effects (such as wind) with the shrilling is an important issue to be addressed. The two-zero high-pass filter applied in the first layer of the model destroys most of the low-frequency amplitude fluctuations we want to model, whereas not applying any high-pass filters will contaminate the meso-temporal model with external factors. A compromise solution is using a softer one-zero high-pass filter before further low-pass filtering. Figure 6 (left) shows an example of this soft high-pass followed by a 48-order low-pass filter applied to a window of size 2^{15} (about 743 milliseconds). Its spectrum, shown in Figure 6 (right), allows the identification of the range of frequencies (up to 50 Hz) predominantly involved in the shrilling effect.

Since this shrilling is typically not steady it wouldn't make much sense to repeat the model of the micro-temporal scale by considering a sliding window of size 2^{15}. Instead, here

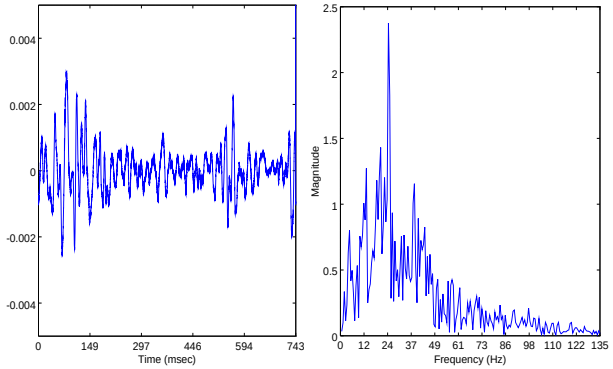


Figure 6. Low-passed amplitude values (left) for the meso-temporal scale representation (743 ms) and corresponding spectrum (right).

we consider a basic sinusoidal model with Gaussian random variables modeling amplitude and frequency values, which are estimated from the TETTIX database for each steady emission type. Essentially this Gaussian model represents frequency and amplitude of the main component [7] of the spectrum of the above processed signal, observed in consecutive windows of 2048 samples or 46 ms. During synthesis these random values are generated once for each 46 ms window, and are linearly interpolated to recreate the smoothness of the amplitude envelope that is applied to the micro-temporal synthesis engine.

2.3 Third layer: macro-temporal model

Moving now to the macro-temporal level, here considered as the large-scale fluctuations in amplitude observed over many seconds, including the occurrence of rhythmic patterns such as stops and resumes in singing (echemes), we have in Figure 7 the RMS envelope (blue line) computed on windows of size 8192 (186 ms) and spanning 2^{21} samples or 47.554 seconds from the initial portion of the recording displayed in Figure 2. This amplitude profile allows us to recognize moments where the emission stops (between 10 and 20 seconds) and also the occurrence of interrupted echemes (especially visible from 30s to 45s).

Considering the kind of profile exhibited by those RMS values over large-scale temporal ranges, of which Figure 7 is a reasonable example, we propose to use a second-order Markov model for updating amplitude information on the synthesis macro-temporal engine.

This model begins with the quantization of amplitude profiles using N linearly spaced amplitude values, which become nodes in the Markov chain, and then generating transition matrices from the observed RMS data in the TETTIX database. For instance, in the above example we might have N=5 different quantized amplitude levels (green dots in Figure 7), each associated with a node in the Markov chain, and second-order transition probabilities reflecting the observed temporal sequence of quantized amplitude levels. Specifically, each triple of adjacent quantized values (a_1, a_2, a_3) observed in the sequence increases a counter $P_{a_1 a_2 a_3}$ in the transition matrix, and finally each line is normalized to satisfy the condition $\sum_{a_3} P_{a_1 a_2 a_3} = 1$.

[7] In Figure 6 (right) we can identify this main component around 25 Hz.

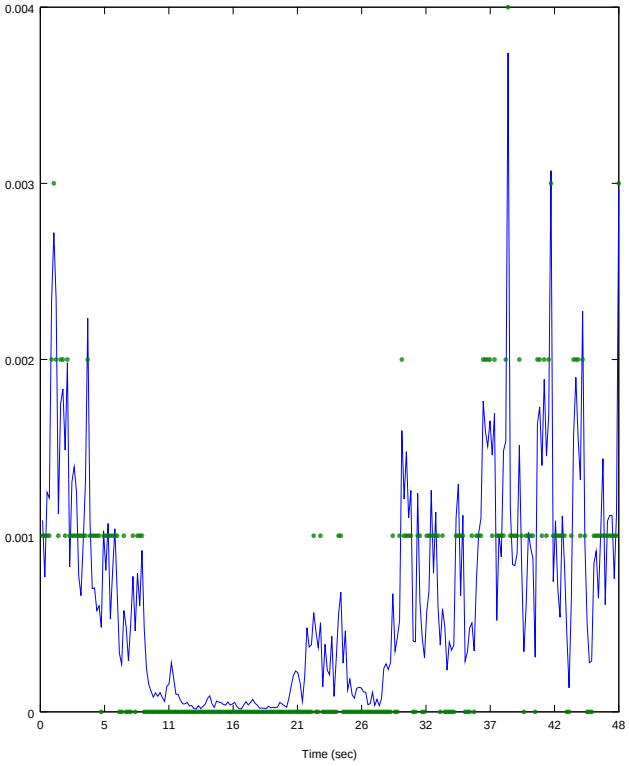


Figure 7. RMS values (blue line) on a larger temporal scale (47.554 s) and quantized values (green dots) used in the Markov model.

In order not to produce over-fitted overly-sparse transition matrices (N too large), and also not overly-dense matrices with poor fitness with respect to the data (N too small), it is important to choose intermediate values for N, which of course depend on the length of the observed RMS data available. Considering that the transition matrix is of size N^3 and that the RMS sequence is of size M, having a 50% occupation of the matrix would require a maximum value of N of the order of $\sqrt[3]{2M}$ (assuming all transitions are different); using half of this estimate, i.e. $N = \frac{\sqrt[3]{2M}}{2}$, makes the observed transitions relatively 8 times higher, with a more evenly-distributed transition matrix for the Markov model.

This second-order Markov model is used in the third-level of the synthesis hierarchy in order to recreate both rhythmic patterns (when the Markov chain jumps between low-level and high-level values of amplitude) and choral aspects such as the variation in the number of insects leading to sudden amplitude changes (assuming there is not also a sudden variation in timbre, in which case the model assumes this is a different emission type and redefines the synthesis engine on all three levels of the hierarchy).

3. FROM MICROSTRUCTURE TO MACROSTRUCTURE: CREATING HETEROTOPIAN VIRTUAL CICADA SPACES

The rhythmic and timbral analysis and synthesis model of the cicada orni call song presented in the previous section serves as a starting point for discussion on the use of this sound synthesis model in artistic applications. For the moment, we will focus on the use of this cicada call through signal processing techniques in order to reconstruct the Plato's Academy soundscape, in an allegorical way, based on the notion of heterotopia [8] in time or in space.

One interpretation of the term heterotopia corresponds to a real place where several other spaces overlap. In this way, Plato's Academy may be transformed into a hybrid real/virtual environment with real Greek cicadas and virtual cicadas from around the world. Another interpretation considers heterotopia in time, where space is treated like a sound museum and sonic events from different moments are used to create heterochronistic metaphors (for instance the suggestion of summer warmth brought by the virtual cicada call during winter). This kind of illusory space is easily recreated by mixing virtual cicada sounds of different types and virtual cicadas from different places using the Pd environment. The three-level hierarchical virtual cicada model previously presented serves as basic building blocks for the construction of a heterotopical soundscape, both in the temporal and spatial senses discussed.

Time and space also serve as metaphors for compositional aspects of heterotopical soundscapes. In the temporal domain different rhythmic patterns can be elaborated by using deterministic and stochastic models to control the parameters of the virtual cicadas, particularly aiming at creating choral effects. In order to make the perception of cicada choirs easier the domino effect and the precedence effect can be explored, as also synchronization and alternation of virtual cicada voices in a polyphonic texture.

In the spatial domain some relevant aspects are the use of different phases, counterpoints and antiphonies between different sound source positions corresponding to a spatially distributed choir of cicadas. Extrapolating this time and space metaphor and extending the spatial aspect to the frequency domain, virtual cicada choirs can be made to produce vowel-like alterations of the original pattern by coloring chosen central frequencies with chosen bandwidths (FOF synthesis).

This virtual sonic environment based on the heterotopian cicada song call can also be treated as a metaphor for the duality and contradictions of the cicada's autochtony [9].

4. CONCLUSIONS

In this paper we have discussed the process of sound analysis and statistical modeling of the Cicada Orni sounds collected by the TETTIX project at Plato's Academy in Athens during the summer of 2014. This sonic database has been analyzed on micro, meso and macro-temporal levels, and on each level a specific synthesis model has been proposed, which combined produce a three-level hierarchical synthesis model for virtual cicadas.

For the micro-temporal level a Gaussian model based on average and variance spectra has been developed, which allows the representation of several different timbres appearing on the recordings, due to changes in temperature, time of the day, emission type and tree type, among others. For the meso-temporal level a Gaussian sinusoidal model has been used to represent the shrilling that occurs and adds complex patterns of amplitude variation on the lowest frequency range (0-50 Hz). On the macro-temporal level a second-order Markov model was proposed for handling both the rhythms appearing due to interrupting/resuming echemes, as well as variations in amplitude due to choral structures observed in the data.

As further research the following topics are planned to be tackled in the near future:

Augmented Aurality: we intend to compose a collaborative soundscape for mobile phones, that will augment the sensorial dimensions of the experience of hearing the cicadas. The participants will be introduced to the processes of soundscape composition, sound design and sound mapping, within the framework of site-specific artistic practice with the use of innovative locative media applications. In this framework mobile phones can be used by people near a place where cicadas sing to acquire and also mix in real time both real and virtual cicadas in an utopian music environment, instrumentalizing a particular form of human-nature interaction.

Evolutionary heterochronistic model: we are planning to develop a computational model for representing cicada sonic behavior evolving over longer periods of time (several months or years). Making such an extended database and the corresponding computational model available online would allow the exploration of real and processed sound of cicadas in the context of an evolutionary model, and also to extract different patterns from this evolution.

Multi-ethnic Heterotopical Soundscape: through modeling and using different species of cicadas that exist in specific geographical sites (for instance mixing the Brazilian Fidicnoides Picea or Quesada Gigas with the Greek Cicada Orni), it would be possible to create multi-ethnic heterotopical soundscapes that would not be observed anywhere in the real world due to ecological and biological constraints of these species.

Acknowledgments

The second author acknowledges the partial support of FAPESP grant 2014/25686-5.

[8] The term heterotopia, introduced by philosopher Michel Foucault in a lecture delivered in Paris on March 14th, 1967 [17], has been used to describe spaces that have added layers of meaning and refer to other spaces, or spaces of otherness, which are neither here nor elsewhere.

[9] Cicadas used to be the emblem of Athenian autochtony.

5. REFERENCES

[1] Anacreon and M. L. W. (Ed.), *Carmina Anacreonta*. Leipzig: B. G. Teubner, 1984.

[2] A. Georgaki, "Listening to the cicada chorus in the plato academy: soundscape research," in *Filigrane. Musique, esthétique, sciences, société*, Musique et écologies du son, Université Paris VIII, 2014.

[3] ——, "Ο χορός των τζιτζικιών της σύγχρονης Αθήνας (the dance of the cicadas in contemporary athens, in greek)." in *Journal Highlights*, Athens, 2003.

[4] I. Xenakis, *Formalized music: thought and mathematics in composition*. Pendragon Press, 1992, no. 6.

[5] D. Rothenberg, *Bug music: how insects gave us rhythm and noise*. Macmillan, 2013.

[6] T. P. Benko and M. Perc, "Deterministic chaos in sounds of asian cicadas," *Journal of Biological Systems*, vol. 14, no. 04, pp. 555–566, 2006.

[7] T. Smyth and J. O. Smith, "A musical instrument based on a bioacoustic model of a cicada," *Proceeding of ICMC 2001*, 2001.

[8] I. J. Patterson, G. Massei, and P. Genov, "The density of cicadas cicada orni in mediterranean coastal habitats," *Italian Journal of Zoology*, vol. 64, no. 2, pp. 141–146, 1997.

[9] J. F. C. Windmill, J. Sueur, and D. Robert, "The next step in cicada audition: measuring pico-mechanics in the cicada's ear," *The Journal of experimental biology*, vol. 212, no. 24, pp. 4079–4083, 2009.

[10] M. F. Claridge, "Acoustic signals in the homoptera: behavior, taxonomy, and evolution," *Annual review of entomology*, vol. 30, no. 1, pp. 297–317, 1985.

[11] D. Young and H. Bennet-Clark, "The role of the tymbal in cicada sound production," *The Journal of experimental biology*, vol. 198, no. 4, pp. 1001–1020, 1995.

[12] H. C. Bennet-Clark, "How cicadas make their noise," *Scientific American (USA)*, 1998.

[13] P. C. Simões, M. Boulard, M. T. Rebelo, S. Drosopoulos, and M. F. Claridge, "Differences in the male calling songs of two sibling species of cicada (hemiptera: Cicadoidea) in greece," *Eur. J. Entomol*, vol. 97, pp. 437–440, 2000.

[14] J. Sueur and T. Aubin, "Acoustic communication in the palaearctic red cicada, tibicina haematodes: chorus organisation, calling-song structure, and signal recognition," *Canadian journal of zoology*, vol. 80, no. 1, pp. 126–136, 2002.

[15] G. Pinto-Juma, P. C. Simões, S. G. Seabra, and J. A. Quartau, "Calling song structure and geographic variation in cicada orni linnaeus (hemiptera: Cicadidae)," *Zoological Studies*, vol. 44, no. 1, pp. 81–94, 2005.

[16] M. D. Greenfield, "Synchronous and alternating choruses in insects and anurans: common mechanisms and diverse functions," *American Zoologist*, vol. 34, no. 6, pp. 605–615, 1994.

[17] M. Foucault, "Des espaces autres," *Empan*, vol. 54, no. 2, pp. 12–19, 2004.

An Augmented Guitar with Active Acoustics

Otso Lähdeoja
University of the Arts, Sibelius Academy
Helsinki, Finland
otso.lahdeoja@uniarts.fi

ABSTRACT

The present article describes and discusses an acoustic guitar augmented with structure-borne sound drivers attached on its soundboard. The sound drivers enable to drive electronic sounds into the guitar, transforming the soundboard into a loudspeaker and building a second layer of sonic activity on the instrument. The article presents the system implementation and its associated design process, as well as a set of sonic augmentations. The sound aesthetics of augmented acoustic instruments are discussed and compared to instruments comprising separate loudspeakers.

1. INTRODUCTION

This paper presents an implementation of "active acoustics" on a steel-stringed acoustic guitar. The term "active acoustics" is employed here to signify a solid surface turned into a sound source with structure-borne sound drivers (or "vibration speakers"). In our case, an acoustic instrument's soundboard is excited with a sound driver, building a new layer of acoustic activity on the instrument.

The present work is part of the augmented instruments paradigm, where the sonic possibilities of an existing instrument are expanded via electronic means. However, the existing body of research on augmented instruments has made wide use of loudspeakers that are external to the initial instrument. Our approach investigates the possibility to conduct the sonic augmentation into the instrument itself, transforming the instrument into an electro-acoustic object (taken literally here) which radiates both the instrument's natural sound as well as the electronic sound. Our project thus stands at the crossroads of two research fields: augmented instruments and structure-borne sound. More specifically, the work explores the advantages provided by a hexaphonic pickup and two parallel vibration speakers in an active acoustic guitar design.

The methodological bias used in this project is one of artistic research. The design is driven by concrete musical needs stemming from an aesthetic preoccupation with the concept of "electro-acoustic chamber music". The work is part of a current trend towards a widening of the scope of electronic music beyond the loudspeaker paradigm and its associated social practices [1].

Copyright: © 2015 Otso Lähdeoja. 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.

2. BACKGROUND

A significant body of research has been achieved on augmented instruments, starting from the seminal work of Tod Machover's hyperinstruments project [2] and, even earlier, Gordon Mumma's hornpipe and related works [3]. More recently, the guitar has gained attention, with an ensemble of projects on electric guitar augmentation [4], the exploration on hexaphonic signal processing [5], as well as radical redesigns of the instrument [6]. The electric guitar itself can be seen as an augmentation stemming from the acoustic guitar via the electromagnetic microphone and analog (from the 80's onward, digital) signal processing [7].

In the domain of acoustic instruments, the area of "active control" of vibrating systems has gained significant attention [8][9]. Active control has been used successfully to modify the modal characteristics of string and wind instruments, with the aim of creating flexible, user controllable acoustics on physical instruments. Stemming from the work on active control, the IRCAM is currently developing an augmented acoustic instruments project under the title "SmartInstruments" [10] [11].

In the consumer electronics domain, a recent patent and a successful quickstarter project features a structure-borne sound driver for acoustic guitar, providing a set of basic audio effects driven into the guitar's back panel [12]. Named "Tonewood amp", the product targets a widespread market with traditional guitar effects. In contrast, our project aims for a radical sonic augmentation of the acoustic guitar, using hexaphonic signals and more adventurous digital signal processing techniques.

3. DESIGN PRINCIPLE

3.1 Design and implementation

The target of our design process is a fully user-controlled augmented acoustic guitar. The sonic possibilities of the instrument should be significantly expanded without hindering the instruments traditional playing techniques, and there should be no need for an external sound technician to operate the instrument. Moreover, after the initial trial period, the augmentation should be compacted to fit into the guitar or into its immediate periphery.

Our project is based on a Breedlove C20 professional-quality acoustic steel string guitar. The guitar is equipped with an Ubertar hexaphonic electromagnetic pickup [13]. Capturing the sound of each individual string is at the basis of our design. A monophonic pickup would provide a summed signal for all the polyphonic elements in the playing, leading to the impossibility to use, for example,

any detailed pitch-tracking, onset detection, or detailed spectral processing algorithms. Moreover, monophonic signal processing is easily stamped with the sonic imprint of the iconic "guitar effects", well developed since the analog era and heard on countless records and live shows. Our project aims to distance itself from the "guitar effects" and seek a drastic transformation of the guitar towards an electro-acoustic instrument, a tool for operating in the sonic domains of contemporary electronic music. The hexaphonic pickup enables to work on the spectral content of each string, as well as to perform detailed analysis on the playing which can be used as control input for signal processing. The hexaphonic output is preamplified and routed into Max/MSP via a RME Fireface audio interface. Audio processing takes place in a Max patch and the output is amplified and routed into two Hiwave 32C30-4B structure-borne audio drivers attached inside the guitar's soundboard. Figure 1 shows an outline of the system as well as the signal flow from the guitar strings back into the soundboard.

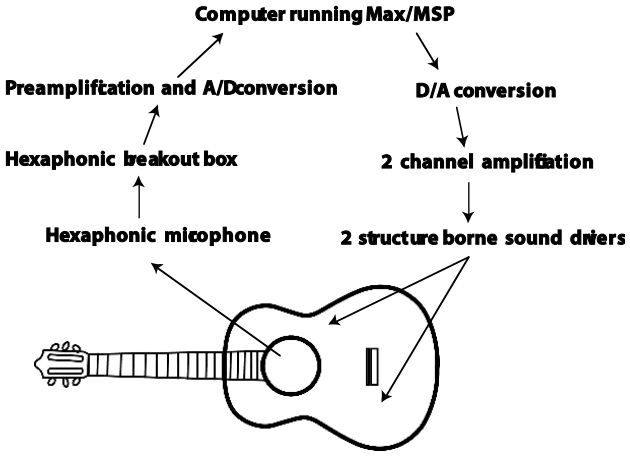


Figure 1. System schema and signal flow from the strings through the processing modules and into the soundboard.

The choice of two sound drivers stems from a perceptual work with the instrument and is not linked to the concept of stereo signal. The localization of a sound driver on a surface affects the sound color dramatically as different resonant modes are being excited. This phenomenon is emphasized by the irregular shape and the bracing of the guitar's soundboard. Starting out with just one sound driver, we were unable to find a single entirely satisfying location on the soundboard. Locations rich in low-end would have deep cuts in middle frequencies, producing a muddy sound. Locations with defined trebles showed a lack of low-frequency content. Thus via experimentation on the instrument, we found out that using two transducers enables to produce a smoother overall spectral response, as well as a more convincing sound diffusion from the instrument as one can feel the whole soundboard radiating. The driver placement was found via a perceptual trial and error. Figure 2. illustrates the placement of the sound drivers inside the guitar.

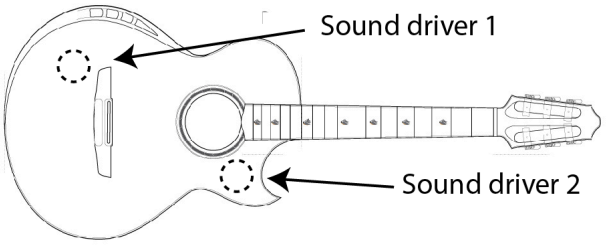


Figure 2. The position of the structure-borne sound drivers on the soundboard, inside the guitar.

3.2 Acoustic study of the system

Following this first exploratory phase we set out to measure the frequency response of each driver-plus-soundboard location couples on our specific guitar model. The HISS Tools Max/MSP library [14] sine sweep method was used in an acoustically controlled studio space. The goal was to provide measured data about the overall response and the prominent modal responses of each of the drivers, as well as of the whole system. The frequency responses are presented in figure 3.:

1) Driver 1 placed behind the bridge, shows a strong response around 100Hz, a huge cut at 250Hz and a series of peaks at approx. 450, 900, 1500 and 2500Hz.

2) Driver 2 placed near the fingerboard shows a weak low-end response, but a smoother overall profile. It may be used to compensate for the low-mid (150-300Hz) loss in driver 1. Prominent modes are found at 450, 1500, 2500 and 4000Hz.

3) The overall system response is shows a consistent mix of the individual driver's characteristics. Prominent modal resonance regions are found around 100Hz, between 450 and 1000Hz, as well as a gradual high-cut profile upwards from 1kHz.

The frequency response analysis of the system provides a basis for an informed equalization of the signals sent to the drivers. The system's frequency response may be compensated via digital filtering [15]. The main modal resonances of the overall system are identified. In the signals driven into the instrument they may be avoided for a clearer overall sound, or enhanced in order to provoke feedback.

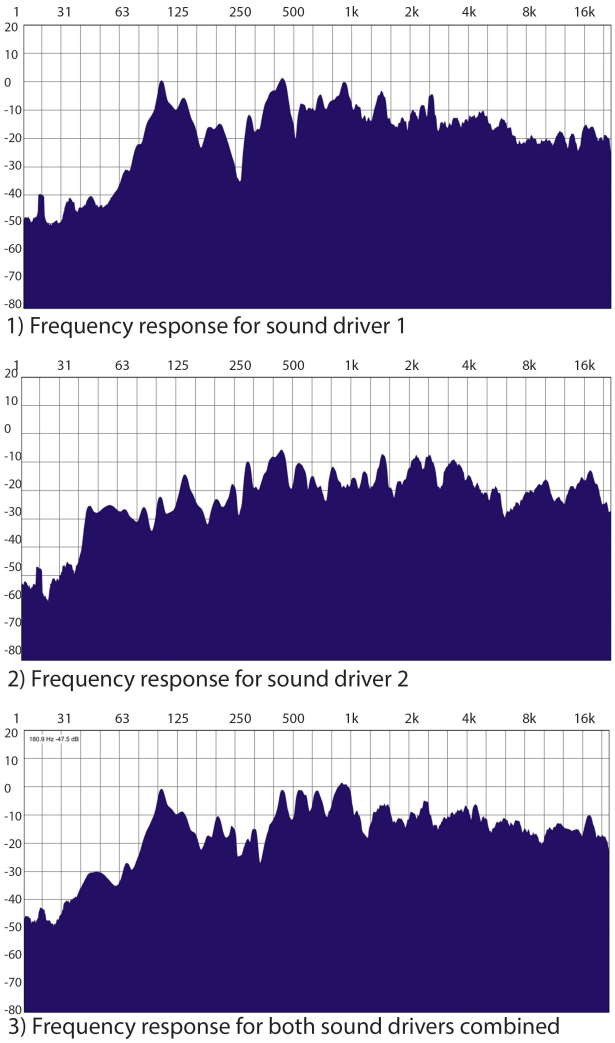


Figure 3. Frequency response representations for the sound driver - soundboard couples: 1) driver 1, 2) driver 2, 3) both drivers.

3.3 A modular design

For the moment our system is modular, composed of separate A/D conversion, (pre)amplification and processing stages, as well as a considerable amount of cables. The system's present state is not in line with our design goal of an integrated, compact instrument where the electronics would fuse with the instrument. However, at the present stage we are still in the process of constituting a proof of concept for the sonic relevancy of an acoustic augmented guitar. We are aiming towards a research-creation loop with a concert praxis with the present instrument. Real-life music-making experience will provide invaluable feedback for the design process. When this trial stage is completed, the possibility of miniaturization will be explored, in connection with the instrument's augmented part's interface design.

4. SOUND DESIGN AND AESTHETICS

Instrument augmentation seeks to enlarge the instrument's sonic possibilities while retaining its entire playability with the traditional techniques. In addition to this initial agenda, our work is motivated by four personal considerations which are of aesthetic and artistic nature:

- Merge electronics and acoustics into one instrument
- The hybrid instrument should have an aural imprint similar to an acoustic instrument (spatial diffusion, spectral characteristics, volume)
- The electronics are not used for amplification, instead-for augmentation of the instrument's sonic possibilities
- Avoid iconic guitar effects

A significant limitation of an acoustic augmented instrument is the inability to deduct the "original" acoustic sound from the instrument's sonic aggregate. The soundboard cannot be damped as the drivers' amplification depend on it, and in our current guitar the strings may not be decoupled from the soundboard. The augmented instrument presents itself as a "guitar plus electronics" system. It is thus impossible escape from the guitar-like sonic signature of the instrument and slide into a completely electronic soundscape. Of course, one may choose to not play and use the instrument as an object-loudspeaker for sound diffusion.

Our project's approach to sound design and aesthetics is based on an idea of sonic typology and complementarity. The acoustic guitar produces a range of sounds typical for a plucked string instrument. The sounds are harmonic (except on some specific playing techniques such as crossed strings or *golpe* hits), with a strong attack portion and a relatively short duration depending on the pitch and the acoustics of the instrument and strings, as there is no energy added to the vibration after the initial attack. By comparison with other traditional instruments and electronic sounds, it is straightforward to deduct a set of sound types the acoustic guitar is lacking. For example, one could think of sustained, percussive or inharmonic/noisy sounds. Also, any kind of modulation or modification of the sound after the attack is alien to the original guitar sound typology. In our perspective, one possible strategy for a fecund augmentation of an instrument is to identify sonic typologies which are absent from the instrument and work to expand it towards them. The aforementioned sound types have served as a guide in the sonic design process, producing four augmentations presented here: 1) long sounds; granular synthesis, 2) long sounds; modal feedback, 3) percussive sounds: attack timbre modification, and 4) Noisy sounds: cross-synthesis with flute fluttertongue sounds.

5. AUGMENTATIONS

A series of experimental augmentations for the active acoustic guitar have been implemented in Max&MSP. Video documentation of the augmentations can be found at http://otsola.org/?page_id=790.

Prior to these augmentation, all six microphone signals are individually equalized for better isolation from crosstalk and to de-emphasize frequency regions which correspond to the measured resonant modes (see figure 3.).

5.1 Hexaphonic granular synthesis

This augmentation uses a synthesized sound to prolong the natural sound of the plucked strings, creating an adjustable resonance to each note played. Each string is routed into a live sampling granular synthesis object (msp *munger~* external), and each string's input is monitored for attacks (with msp *bonk~* external). A detected attack actions a gain ramp, gradually bringing the synthesized granular sound into the mix in order to replace the waning natural sound. A "freeze" option runs the granular synthesizer in loop mode and offering the possibility to prolong the synthesized tones without limitations. The *munger~* object allows for a rich shaping of the granulation parameters, moreover with six channels running simultaneously.

5.2 Modal feedback

The modal feedbacker augmentation is a dynamic equalizer tuned to the main resonant modes of the guitar (100, 450, 1000Hz). These key frequencies are the "sensitive spots" of our guitar; any substantial energy driven into it at these frequencies will be trapped as a standing wave in the soundboard, amplified and easily transferred into the strings, thus forming a feedback loop with the microphone. Our augmentation uses a frequency-specific analyzer on the summed output of all the strings (msp *sigmund~* external) to monitor the energy on the key frequency bands. The analyzer output is mapped to the gain of three high-Q resonant filters (msp *cascade~*), augmenting the gains when the sound contains little energy at the key frequencies, and lowering the gains in case of overload. With careful tuning of the system, the augmentation produces a "hot" feedback on the guitar, just under control but giving a sense of the guitar responding to the playing with extreme sensitivity.

5.3 Attack timbre modification

Each string is monitored for attacks with the msp *bonk~* external. A detected attack triggers a percussive sample which is cross-synthesized with the original signal and driven into the soundboard. In our experiments, we used flute flutter-tongue attack samples and the *swinger~* cross synthesis external from the fftease library [16]. The system augments the percussive character of the guitar. It proves particularly effective with the palm-mute playing technique, producing hybrid "flute-guitar" attack sounds. We also experimented with further convolution processing of the hybrid timbres, adding an adjustable feeling of space in the sound. An interesting option is the application of a different impulse response convolution on each string, for example, larger halls for the lower strings and tighter spaces or objects for the upper strings. The guitar thus gains a sense of different spatial percepts - from large and reverberant to tiny and constricted - according to the string played.

5.4 Noisy signals

The guitar produces mainly harmonic sounds. In our sense, it would be musically engaging to be able to produce inharmonic timbres, and especially towards the noisier types of spectra. Our augmentation uses the same hexaphonic attack detection and sample playback system as the attack timbre modification patch described above. However, instead of short samples, long samples of flute air-noise are employed here, real-time cross-synthesized with the original signal. In addition, pitch detection (msp *sigmund~* external) on each string is used to dynamically drive a bank of 1-band resonant filters which constrict the noise spectrum to a narrow band around the note played. The result is an airy flute-like resonance added on the acoustic sound of the guitar.

6. DISCUSSION

While being at their very early and experimental stages, the augmentations presented in this paper provide engaging new sound possibilities for the acoustic guitar. Driving the processed sound into the guitar itself creates a coherent ensemble with the acoustic and the electronic sounds radiating from the same source. In this sense, the usual sonic dichotomy of the instrument - loudspeaker combination is bypassed in favor of a single instrumental entity. The aural image given by the electro-acoustic instrument is radically different from a loudspeaker: the guitar's directionality points towards a cardioid pattern over the whole spectrum, while loudspeakers radiate in a closed angle on high frequencies. Also, the audio spectrum modified by the guitar's tonewood results in a characteristic aural imprint for both acoustic and electronic sounds.

For the instrumentalist, the newfound simplicity of a single comprehensive sound tool can be an opportunity to focus and work in a tighter multimodal feedback loop with the instrument. One might argue in favor of a strong sense of embodiment with a single vibrating object between the hands and pressed against the torso, as opposed to a larger modular system with separate speakers, producing a sense of distance with the sound. With the sound radiating from the instrument itself, problems related to onstage monitoring are greatly reduced. Also, the sound level is that of an acoustic instrument - a desirable factor in the aesthetic perspective that motivates us, living with the aural fatigue of decades of (amplified music and urban soundscapes.

However, the present state of our system discards the question of the interface. The augmented part of the instrument is still running on a separate PC, interfaced with a mouse, keyboard and a screen. While our augmented instrument (re)gains its integrity in the domain of sound radiation, it still portrays a divide on the level of the interface. The current interface is an incoherent sum of a refined haptic instrument on one side, and a visual, symbolic computer interface on the other side. Thus the work towards a more integrated augmented instrument faces a major challenge. This ideal instrument would have not only the electronic and the acoustic sounds emanating from a single vibrating object, but would also present a coherent set of affordances tightly integrated on the instrument, and which would not counter the existing playing techniques.

On the technical level, the electromagnetic pickup used in this project would be favorably replaced by a hexaphonic piezoelectric one, producing less noise and being less prone to external disturbances such as stage lights. The level of crosstalk between strings/microphone channels would also be reduced, resulting in a higher precision in digital sound processing and analysis. Feedback is an issue when working with sounds close to the guitar's natural output, emphasizing its resonant frequencies and with high gain. We do not experience significant feedback problems on sounds which avoid reproducing the guitar's tone and its resonant frequencies, with sound intensity levels close to the initial acoustic sound. The possibility of feedback can also be used creatively, as in the modal feedback augmentation (see section 5.). Processing latency is a sensitive parameter for all live processing. In our augmentations, latency is especially apparent on more percussive sonic typologies such as the attack timbre morphing. Perceptually, with a typical ~20 millisecond latency the playing experience is still coherent, but with a sense of fuzziness on the attacks.



Figure 4. Our Breedlove C20 active acoustics augmented guitar, with a hexaphonic pickup and two sound drivers. Two extra drivers are shown here on top of the soundboard for illustration purposes.

7. FUTURE WORK

The augmented guitar presented in this article provides a proof of concept for the implementation of active acoustics with a hexaphonic pickup and a double sound driver system on a folk guitar. The work points towards further research on the issues of the interface and compacting the electronics into the guitar itself. The intention is to develop the guitar within a framework of constant dialogue between artistic praxis (composition and performance) and technological research. The next generation of the active acoustic guitar will be constructed with a luthier in order to optimize the sound driver placement and sonic characteristics. The beginning of a concert praxis with the instrument is planned for autumn 2015.

Acknowledgments

The author would like to thank the Academy of Finland for the generous support on the present postdoctoral research.

8. REFERENCES

[1] S. Trower, "Senses of vibration", Continuum International Publishing Group, London, 2012

[2] T. Machover, "Hyperinstruments – A Progress Report 1987-1991", Technical report, Massachusetts Institute of Technology, 1992.

[3] G. Mumma, Creative Aspects of Live Electronic Music Technology, Audio Engineering Society, *Papers of 33rd National Convention*, New York 1967.

[4] O. Lähdeoja, "An approach to instrument augmentation: the electric guitar." *Proceedings of the International Conference on New Interfaces for Musical Expression*, Genova, Italy. 2008.

[5] R. Graham, "The Expansion of Electronic Guitar Performance through the Development of Interactive Digital Music Systems", Ph.D. Thesis, University of Ulster.

[6] O. Lähdeoja, et al. "The electric guitar: an augmented instrument and a tool for musical composition." Journal of Interdisciplinary Music studies 4.2 (2010): 37-54.

[7] Touch screen guitar: http://misa-digital.myshopify.com

[8] S. Benacchio, et al. "Active control applied to string instruments", *acoustics 2012*, Nantes, 2012

[9] M. Van Walstijn, and P. Rebelo. "The prosthetic conga: Towards an actively controlled hybrid musical instrument." *Proceedings of the International Computer Music Conference*. 2005.

[10] Ircam SmatrInstruments project http://instrum.ircam.fr/smartinstruments/

[11] A. Mamou-Mani, "Adjusting the soundboard's modal parameters without mechanical change: A modal active control approach", *ASA meeting*, Indianapolis. 2014

[12] Tonewood amp http://www.tonewoodamp.com

[13] http://www.ubertar.com/hexaphonic/

[14] A. Harker, and P.A. Tremblay. "The HISSTools impulse response toolbox: Convolution for the masses." *Proceedings of the International Computer Music Conference*, 2012.

[15] O. Lähdeoja, A. Haapaniemi, and Vesa Välimäki. "Sonic Scenography-Equalized Structure-borne Sound for Aurally Active Set Design." Proceedings of the *International Computer Music Conference/Sound and Music Computing Conference*, 2014.

[16] E. Lyon, and C. Penrose, "FFTease: A Collection of Spectral Signal Processors for Max/MSP" In *Proceedings of the International Computer Music Conference*, 2000

Synchronizing Spatially Distributed Musical Ensembles

Aristotelis Hadjakos, Axel Berndt, Simon Waloschek
Center of Music and Film Informatics (CeMFI)
University of Music Detmold / University of Applied Sciences OWL
{hadjakos,berndt,waloschek}@hfm-detmold.de

ABSTRACT

Spatially distributed musical ensembles play together while being distributed in space, e.g., in a park or in a historic building. Despite the distance between the musicians they should be able to play together with high synchronicity and perform complex rhythms (as far as the speed of sound permits). In this paper we propose systematic support of such ensembles based on electronic music stands that are synchronized to each other without using a permanent computer network or any network at all.

1. INTRODUCTION

First attempts to explicitly design spatiality in music can already be found in the Renaissance and Baroque area. Giovanni Gabrieli (1557-1612) positioned trumpet players on the side galleries of his church and at times alternated between the trumpet groups [1]. In the further course of music history such spatial concepts where artistically explored again and again, from Berlioz (1803-1869) in his Symphonie fantastique [2], where an oboist enters the concert hall while playing, up to todays artificial spatiality through the use of surround sound.

In this paper we explore how to support spatially distributed musical ensembles. Such ensembles could, e.g., play in a park, making it possible for the audience to explore the piece by moving around. Or the ensemble members could be placed in co-located rooms in a building or spread out in the lobby of a concert hall during the intermission. Due to the distance, the musician's own instrument will usually drown out the sound of the other ensemble members, making it difficult to play synchronously. Furthermore, the musicians may not be able to see each other, making visual cues impossible. Previous realizations have relied on conductors that were visible for all ensemble members (e.g., to synchronize the musicians in the orchestra pit and the performers on stage) or they have relied on click tracks that were transmitted over wireless headphones.

To understand how click tracks are currently created, we performed informal interviews with musicians, composers, and electronic music artists. They used a variety of non-specialized software tools. In particular they used audio editing tools like Audacity or music notation tools for that purpose (Figure 1). We created WebMaestro, a web-based click track editor and player (see Section 3) to make it easier to create click tracks and provide better support for rehearsal situations. In addition to auditory cues, we wanted to provide the musicians with a visual display that gives them a representation of the musical beat and the current position in the piece. Since the musicians do not hear each other well enough at all times, this makes sure that the performance does not fall apart when, e.g., one musician miscounts rest bars. Our interactive music stand, the "M-Sync Player" (see Section 4) provides visual as well as auditory cues for synchronizing spatially distributed ensembles.

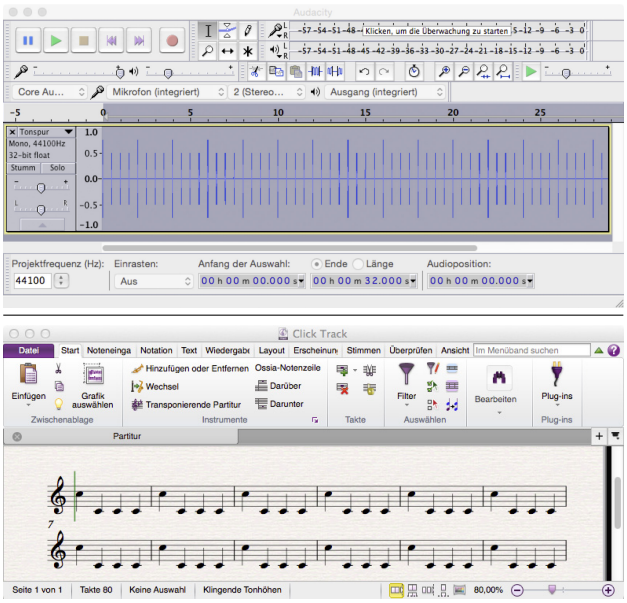


Figure 1. Audio editing (top) and music notation tools (bottom) are commonly used to create click tracks.

As a wired or wireless network may not always be present (e.g, outside in a park, in a historic building) or accessible (e.g., in a big concert venue), we were interested in synchronizing the M-Sync Players without having to rely on a network. We discuss (Section 5) and evaluate (Section 6) different synchronization strategies.

2. RELATED WORK

Many very different projects are faced with the situation of a distributed music-making and its key challenge of affording inner-ensemble communication. Besides the use

Copyright: ©2015 Aristotelis Hadjakos, Axel Berndt, Simon Waloschek 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.

of synchronized click tracks and low-latency audio transmission, this situation motivates the augmentation of traditional music stands and the use of networked digital music stands as platform to mediate communicative cues between the players. This section pinpoints some representative works in the field of distributed music-making to give an impression of the variety of scenarios. Then it introduces digital music stands and related research.

2.1 Networked Music-Making, Performance, Tuition

Networked music-making often requires a more or less complex hard- and software setup. With the JamBerry Meier et al. present a very compact stand-alone device, which is based on the Raspberry Pi, extended by a high-quality audio interface and a touchscreen [3]. The JamBerry focusses on the low latency audio transmission. Further means for communication between the players are not implemented so far.

Inner-ensemble communication is a complex and often very subtle combination of visual and auditory cues. Typical examples are facial expressions, body movements and breathing. Schober [4] provides an overview of such coordinating cues and discusses their translation into virtual environments where players can be collocated even if physically distant. Distributed music rehearsal systems are presented by Konstantas et al. [5] and Alexandraki et al. [6].

Duffy & Healey [7] compare music tuition in collocated and video mediated situations. Among other observations, they point out the importance and efficiency of gesture interaction on the shared music score which gets lost in the video mediated setup: "The importance of the shared score to lesson interaction was evidenced by problems managing interaction such as turn control when participants were separated and could no longer share the same physical representation of the music." They motivate "to involve an interactive visual layer over a digitised representation of the physical score, which shows the separated participants where each person is gesturing on the music. Ideally both participants should be able to mark their layer in a way which allows the student to take an annotated copy away, and return with it for the next lesson. There should be a way for the tutor to communicate intent to interrupt the student's performance through visualization of gestures on the music."

A dynamic digital medium such as a digital music stand can display not only static scores. Brown [8] generates the score live at its performance, which requires the human player to have great sight-reading skills. Freeman's [9] interactive realtime score generation and distribution to live performing players goes even a step further. Here, the audience can interactively influence the score generation process while it is being performed by human players. Not only can the composer be replaced by virtual instances but also parts of the ensemble, letting humans play together with computer instruments. A fully automated digital and spatially distributed music ensemble is described by Kim et al. [10]. In today's concert practice such cooperative human-computer music performances are typically coordinated by click tracks. These force the human to follow the computer. Some approaches also make the virtual performer responsive to human musicians, such as Liang et al.'s framework for coordination and synchronization of media [11].

2.2 Digital Music Stand Technology

The typical functionality of electronic music stands, besides score display, comprises the management of a sheet music database, the possibility of adding annotations and performance instructions, metronome and pitch tuner integration, and hands-free page turning (a key feature of electronic music stands, traditionally triggered via foot pedal).

Commercial products and patents exist for more than a decade now, like the *eStand Electronic Music Stand* [1] (a review of the eStand is given by Cross [12]), the *MusicPad Pro* and its successor the *MusicOne* stand [2], and patents like Kumarova's digital music stand [13]. Besides these commercial instances several academic research projects deal with the development of electronic/digital music stands and related issues, like the *Espresso* digital music stand of Bell et al. [14]. In one of the first concept papers on digital music stands Graefe et al. [15] introduced the *muse* concept that never came to a full technical implementation but inspired many subsequent projects.

The MICON system is a music stand for interactive conducting of orchestral audio and video recordings [16]. The system is part of an exhibit with a focus on non-professional users. The exhibit implements a conducting gesture recognition which is connected to video and audio time stretching so that the music and the video of the orchestra react to the user's gestures. The MICON features several different score visualizations, automatic page turning animations, and an animated visual cueing system that indicates the current playback position within the score. In his study, Picking [17] already noted that such visual cues are very popular. MICON's beat visualization is a potential candidate for a visual click track.

With their Multimodal Music Stand, Bell et al. [18] introduced an augmented traditional music stand that seamlessly blends into a musical instrument. Equipped with microphones, cameras, and electronic field sensors the stand "augments the performance space, rather than the instrument itself, allowing touch-free sensing and the ability to capture the expressive bodily movements of the performer" [18]. The sensor data may provide a prospective starting point to integrate a new approach to inter-player communication.

Communication capabilities within the orchestra, i.e., with other music stands, were already part of the muse concept [15]. The MOODS (Music Object-Oriented Distributed System) is designed to equip a whole orchestra [19] and features corresponding networking capabilities. It interfaces with a score database, automatically generates parts, allows cooperative editing, managing/versioning, and distribution of the scores throughout the orchestra. Similar networking capabilities are described by Romero

[1] published by eStand, Inc., `http://www.estand.com` (last access: Apr. 2015)

[2] both, MusicPad Pro and Music One, are published by SightRead Ltd., `http://www.sightread.co.uk` (last access: Apr. 2015)

& Fitzpatrick [20] and Connick [21]. Laundry's developments on the music typesetting and annotation of music scores complements this work [22].

2.3 Further Contextual Research and Studies

Contextual studies on electronic/digital music stands has been performed by Picking [17] amongst others. Picking compares music reading on paper with music reading on screen (static and animated). He notes that the study participants preferred an animated score presentation over the static and paper presentation. The use of cursor-like markings that indicated the current (playback) position in the music turned out to be very popular among the participants. Here, research on automated score following and music alignment provides the potential technical complement [23–25]. These indications are most interesting for player synchronization tasks and serve as a replacement of traditional visual click tracks.

Bell et al. [26] investigate two further core aspects of the visual score presentation: page turning animation and image size. A user study compared six page turning variants, including cascaded blending, horizontal scrolling, and vertical scrolling [27], of which the participants preferred to keep control over changes instead of fully automatic animations. A similar experiment is described by Blinov [28]. In their image size study Bell et al. did not observe significant differences in the participants' performances while proofreading on large and small scales. But the participants favored the larger scale for convenience reasons.

Kosakaya et al. refined their page turning scheme via time delays based on glance analyses [29]. The muse concept employs a microphone for audio-to-score alignment to estimate appropriate page turning moments automatically. Research and development on page turning are continued until today [27, 30, 31].

3. WEBMAESTRO

WebMaestro [3] is a web-based click track editor and player. It is a self-contained application and can be used to edit and play back click tracks instead of using non-specialized software like audio editors or music notation editors for this task (see Section 1). WebMaestro can further be used as a pure editor, preparing a representation that is played back by synchronized M-Sync Players (see Section 4). An overview of WebMaestro's user interface is shown in Figure 2. In 2014 WebMaestro was used at the Internationale Ferienkurse für Neue Musik in Darmstadt for the rehearsal and performance of the piece à tue-tête for nine spatially distributed wind players by Fabien Lévy. The piece was performed by the ensemble "Klangforum Wien". [4]

3.1 Models and Tempo

Our solution uses two models: the editor model and the playback model. The editor model represents the parts that are relevant for editing a click track, including time signatures, tempo, accelerando and ritardando, etc. This is the model that the composer generates and modifies with the help of WebMaestro's user interface. The playback model on the other hand addresses the timed succession of events. In particular, tempo and tempo changes are boiled down to delta times (time differences or inter-onset intervals) between successive events.

The editor model represents the piece as a sequence of sections. A section is a sequence of bars with the same musical meter and the same tempo. The following code example represents a section with a tempo change:

```
{
  "bars": "5-8",
  "signature": "3/4",
  "bpm": "60-96",
  "tempoChange": {
    "begin": "6:2",
    "end": "9:1",
    "curve": "Natural"
  }
}
```

The section extends from bar 5 to bar 9, with a 3/4 time signature, and a tempo change that begins on the second beat in measure six, with a "natural" (quadratic interpolation) tempo curve. We use quadratic interpolation by default as this has been shown to be close to what musicians typically do [32].

The playback model is a representation that is simple to render. Similar to the MIDI file format it is based on delta times, i.e., time differences between subsequent events. Each event is defined by its delta time, its type and its content. The following representation denotes the second beat in a 4/4 time signature. It has a delta time of 1 second to the previous event:

```
{
  "delta": 1,
  "type": "beat",
  "content": "2/4"
}
```

Having a separate playback model greatly simplifies the implementation of the M-Sync Player, since tempo calculations are already contained in the delta times. This is important as we plan to port the M-Sync Player to different platforms including Windows, Android and iOS.

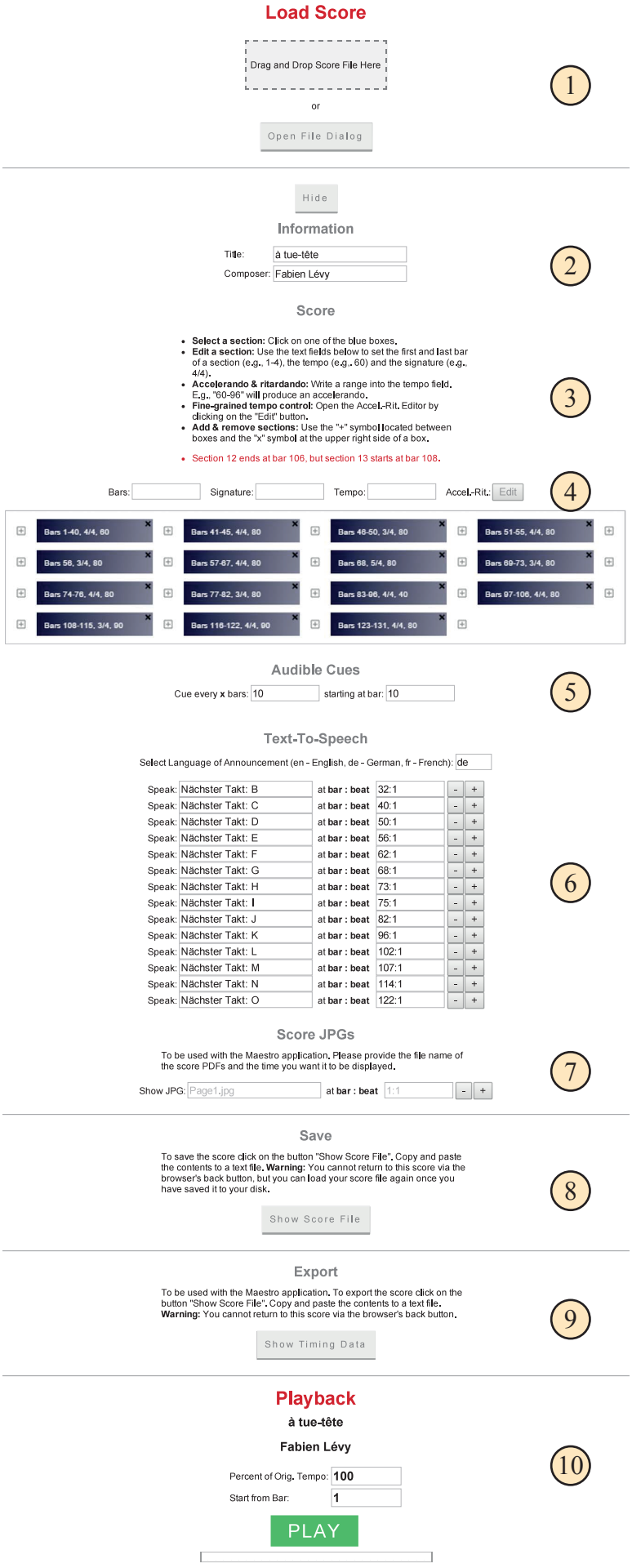
3.2 Implementation

WebMaestro's audio output was realized using the Web Audio API. Audio samples and JPEGs were encoded with base64 directly as JavaScript strings contained in the HTML file. This makes WebMaestro usable without network as a single self-contained HTML file, which is sometimes useful in rehearsal situations without network access.

4. M-SYNC PLAYER

The M-Sync Player (see Figure 3) displays and advances the score, visualizes the musical beat and also plays the sound of a metronome. All of that is done synchronously

[3] http://zemfi.de/downloads/WebMaestro.html
[4] http://www.klangforum.at/



Explanations:

1. Score files can be loaded either by dropping them onto the marked box area or by using a dialog.

2. The title and the name of the composer can be entered.

3. A short manual is provided that describes the basic functionality of the editor. Common user errors are displayed immediately as red warnings.

4. Sections are defined by providing the bar numbers as well as the signature and the new tempo. Accelerando and ritardando can be specified by providing a tempo range, e.g., 60-96, in the tempo field. An Accel.-Rit.-Editor provides fine-grained control over the tempo change.

5. Audible cues can be generated periodically every n bars.

6. A web-based speech synthesis may be used to give vocal cues at given bars and beats.

7. The M-Sync Player is able to display the score with automatic page turning if the corresponding JPEG files are provided.

8. Edited click tracks can be saved as plain text files and loaded at a later time again.

9. The timing data can be exported for later usage with the M-Sync Player.

10. Finally, the click track can be played directly in the web browser. For rehearsals, the user can select from which bar to start and change the overall tempo of the entire playback.

Figure 2. The user interface of the WebMaestro

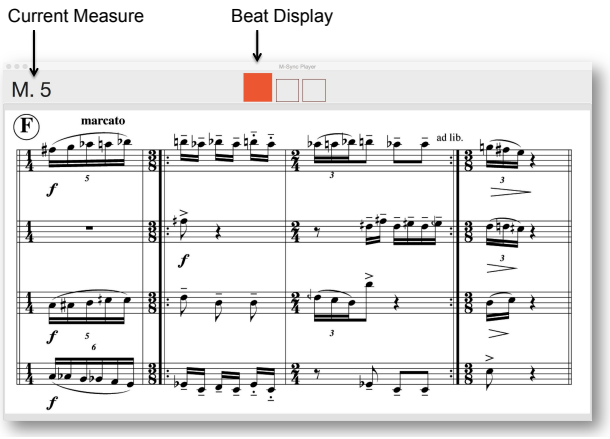


Figure 3. The M-Sync Player

on all computers using one of the synchronization methods described in Section 5. To display the content, the M-Sync Player uses the playback model generated by WebMaestro. The M-Sync Player provides a beat display that indicates the current beat with a filled box while all other beats are shown as outlined boxes. The boxes are relatively large to make it easier for the musician to follow those visual cues in peripheral vision while looking at the musical score below. At the upper left, the current beat is displayed. Together with the automatic advancement of the score, this ensures that the performance does not fall apart if one player looses track of the current position, e.g., by miscounting rest bars. Such errors can otherwise be difficult to spot since the ensemble members may not hear each other well enough in the targeted distributed situations. In addition to the visual cues, the M-Sync Player also generates auditory cues with separate sounds for the first and the following beats of a bar.

Furthermore, the M-Sync Player generates OSC messages that can be received by other applications on the same machine. This can be used to synchronize electronic music, e.g., generated by a Max patch, or a visualization, e.g., generated by Processing, to the performance of the ensemble.

5. SYNCHRONIZATION

The performance of spatially distributed music can take place in parks, historic buildings or big concert venues where it may be difficult to get access to a wired or wireless computer network. Therefore, we examine different synchronization options that require no (Distributed Button, radio time signals, GPS) or no permanent (NTP) network connection. In order to display the score and play the click track simultaneously on multiple computers, their clocks have to be synchronized with great accuracy. However, computer clocks may not only deviate by a static time interval but the clocks may also drift due to slightly different speeds (see Figure 4).

We distinguish one-shot synchronization and continuous synchronization. In one-shot synchronization, the systems are synchronized once, i.e. before the performance has started. In continuous synchronization, the computers are

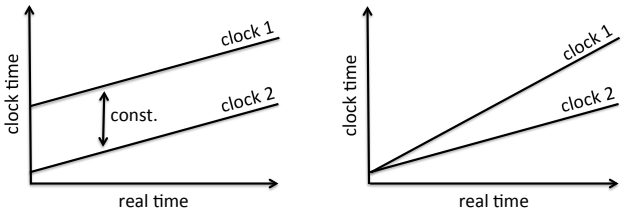


Figure 4. Clock offset (left): the clock readouts differ by a constant amount. Drift (right): Although the clocks are initially synchronous they continuously drift apart since one clock runs faster than the other.

connected to an external clock that corrects the computer clock in regular intervals.

5.1 One-Shot Synchronization

5.1.1 NTP

The Network Time Protocol (NTP) is a protocol to synchronize computers via the Internet. Clock synchronization is acquired by exchanging four messages. For each exchanged message the sender and the receiver measure the send and reception time with their local unsynchronized clock. Based on this information, the offset between the two clocks and the transmission delay can be calculated, making it possible to adjust the client's clock to the right time. However, the transmission delay has to equal in both directions (or close to equal) for NTP to work properly.

5.1.2 Distributed Button

For the user, the Distributed Button is a big box with USB connectors and a button on top (see Figure 5). First, the users connect their computers to the box and then one user presses the button on top. This event is received on all computers simultaneously and used to synchronize all M-Sync Players.



Figure 5. The Distributed Button

5.2 Continuous Synchronization

5.2.1 Radio Time Signals

Radio time signals transport an encoding of the current time over radio waves. Typically, amplitude or frequency modulation is used to encode the bit representation of date

and time on long, medium or short waves. Radio time signals are available all over the world.

Being located in Europe, we used DCF77 signals. DCF77 is a long wave radio time station located near Frankfurt, Germany. It provides radio time signals that can be received in large parts of Europe. DCF77 uses amplitude modulation and generates pulses of varying length each second. The bits are encoded by changes in pulse lengths: A 100 ms pulse is a zero and a 200 ms pulse a one. The bits encode the current date and time and also provide parity bits, which provide error detection to single bit errors. No pulse is sent on the last second of each minute. Then the next pulse indicates the beginning of a new minute. We used an Arduino shield with a DCF77 receiver [5] (see Figure 6).

5.2.2 GPS

The Global Positioning System (GPS) is based on a multitude of satellites orbiting Earth. Each satellite sends its position in space together with a highly accurate time stamp obtained from an onboard atomic clock. The signal spreads out with the propagation speed of light and eventually reaches the receiver. The intersection of those signal spheres from multiple satellites determines the position of the receiver. In order to calculate this intersection point however, the receiver needs to have a very accurate clock in order to determine the distance from a satellite as a function of the time stamp from the satellite and reception time. Since GPS receivers need to be cheap, a clock signal is reconstructed from the satellite signals. In essence, four-dimensional hyper-spheres originating from multiple satellites are intersected to calculate 3D position and the current time. While GPS users are typically more interested in the position signal, the time signal can be used to synchronize spatially distributed musical ensembles.

GPS receivers are available at relatively low cost and compatible to popular physical computing platforms. We used two GROVE GPS sensor modules [6] (version 1.1 and 1.2) and interfaced them to an Arduino Leonardo using a SPINE shield (see Figure 6).

6. EVALUATION

6.1 Procedure

We wanted to assess the synchronization accuracy that can be achieved with the different synchronization methods. We employed the following evaluation procedure: Two M-Sync Players running on two different computers were synchronized with one of the said methods. The M-Sync Players were triggered to begin playing at a particular time and rendered a half-hour long steady 60 bpm pulse in 4/4 time. We recorded the audio output of the two M-Sync Players using a custom-made cable with two singal-in headphone connectors and one signal-out headphone connector. The signal-in connectors where connected to the headphone outputs of the two computers and the signal-out connector was connected to the line-in of a separate computer that we used as a recording device. This provided us with a stereo signal where the left channel originates from the M-Sync Player of one computer and the right channel from the other. In the experiments we used a MacBook Pro (Retina, 15", mid 2014) and a MacBook Pro (13", mid 2012). Both computers where running OS 10.10.2.

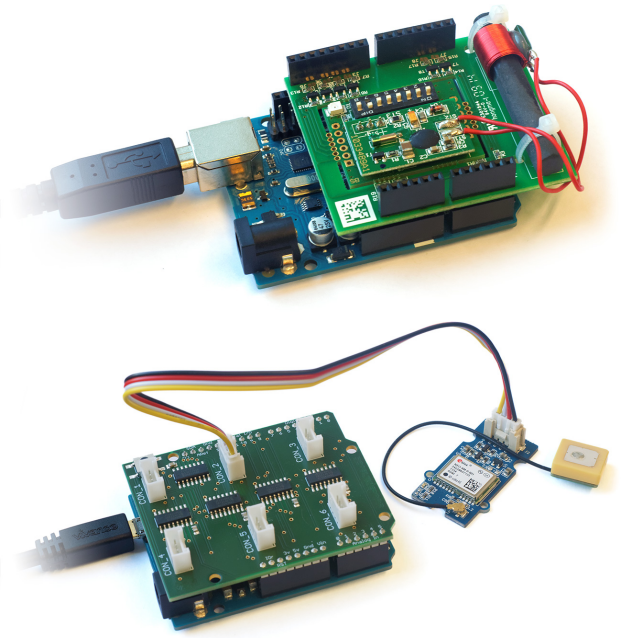


Figure 6. An Arduino with DCF77 shield (top) and an Arduino with a SPINE shield connected to a GROVE GPS module (bottom)

6.2 Results

We examined the timing deviations between synchronized M-Sync Players. For this purpose the time difference of the onsets on the left audio channel and the corresponding onset on the right channel were determined with a MATLAB script, which detected beat onsets with thresholding.

One-shot synchronization: For NTP, we manually initiated the computers to synchronize themselves to an NTP server on the Internet before we started the M-Sync Players. While the Distributed Button provides more accurate clock offset compensation than NTP, i.e., the computers start out with less deviation, the computer clocks drift apart with increasing differences of about 3 ms/min (see Figure 7). This drift makes it problematic to perform pieces that are more than a few minutes long. Instead of using the clock offered by the operating system, we then measured time by counting the number of samples that are sent to the built-in sound card at a fixed rate of 44.1 kHz. The drift rate sank to about 1 ms/min. We then measured the overall deviation after 30 minutes and computed the (almost) constant deviation of the audio rates of the two computers. By compensating for that exact amount, we were able to achieve a drift rate of about 0.0367 ms/min between the two M-Sync Player (see Figure 8). Using this method, the two M-Sync Players drift only about 1 ms apart after 30 minutes, which is well below what is musically relevant.

[5] `http://bit.ly/1CtXOR8`
[6] `http://www.seeedstudio.com/wiki/Grove_-_GPS`

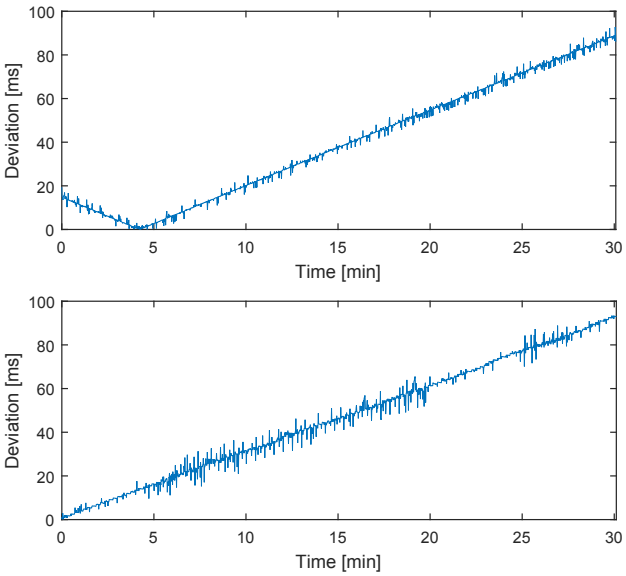


Figure 7. One-shot synchronization: NTP (top) and Distributed Button (bottom). The Distributed Button provides a better clock offset compensation than NTP.

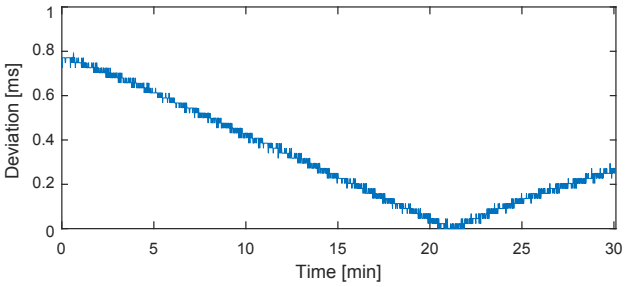


Figure 8. Interchannel deviation using the Distributed Button and the internal audio clock.

Continuous synchronization: We then examined GPS and DCF77-based synchronization. The GPS modules we used did not drift but had substantial timing irregularities (see Figure 9, top), making them unusable for our purposes. However, we observed distinct differences between different GPS modules so that there probably is a suited GPS module, which we have not identified yet. DCF77 on the other hand provides good synchronization with a maximum deviation of 11.43 ms and a mean deviation of 2.4 ms without introducing any long-term drift (see Figure 9, bottom).

7. CONCLUSION

In this paper, we have explored how to systematically support spatially distributed musical ensembles. The WebMaestro click track editor and player lets the user define and play back complex click tracks with changes in tempo & meter, accelerando & ritardando together with text-to-speech announcements. Furthermore, WebMaestro lets the user export a playback model, which can be used in conjunction with the M-Sync Player to visualize the musical score and provide visual cues for beats together with auditory metronome beats. In many places where one would

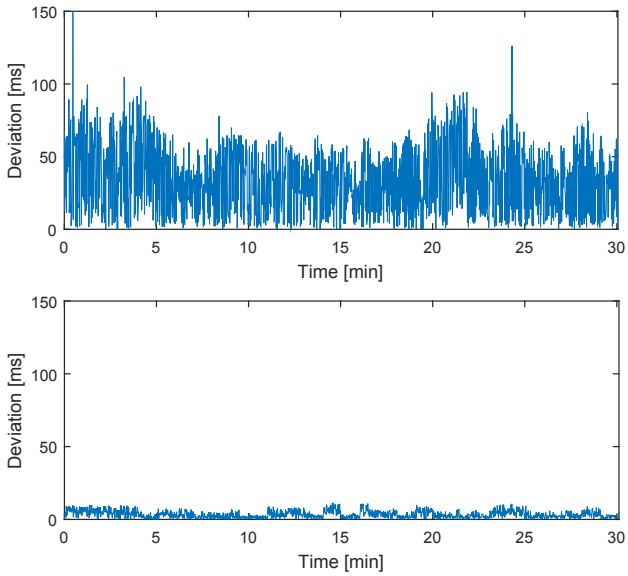


Figure 9. Continuous synchronization: GPS (top) and DCF77 (bottom). In our experiments, DCF77-based synchronization worked significantly better (about one order of magnitude).

want to perform with a spatially distributed musical ensemble, it is oftentimes difficult to get access to a wired or wireless computer network. Therefore, we explored and evaluated a variety of synchronization methods that can be realized without (or without permanent) network access. The Distributed Button (best one-shot synchronization) and radio time signal synchronization (best continuous synchronization) turned out to be the best options. Additionally, we experienced that the internal clock sources of audio interfaces built into computers are more accurate than regular system clocks.

8. REFERENCES

[1] G. Gabrieli, *Sacrae Symphoniae*. Venice, Italy: Apud Angelum Gardanum, 1597, vol. 1.

[2] H. Berlioz, *Symphonie fantastique*, N. Temperley, Ed. Kassel, Germany: Bärenreiter-Verlag, 2012.

[3] F. Meier, M. Fink, and U. Zölzer, "The JamBerry—A Stand-Alone Device for Networked Music Performance Based on the Raspberry Pi," in *Linux Audio Conference*, vol. 2014, 2014.

[4] M. F. Schober, "Virtual environments for creative work in collaborative music-making," *Virtual Reality*, vol. 10, no. 2, pp. 85–94, 2006.

[5] D. Konstantas, Y. Orlarey, O. Carbonel, and S. Gibbs, "The distributed musical rehearsal environment," *IEEE Multimedia*, vol. 6, no. 3, pp. 54–64, 1999.

[6] C. Alexandraki and D. Akoumianakis, "Exploring new perspectives in network music performance: the DIAMOUSES framework," *Computer Music Journal*, vol. 34, no. 2, pp. 66–83, 2010.

[7] S. Duffy and P. G. Healey, "Spatial co-ordination in music tuition," in *Proceedings of the 34th annual conference of the cognitive science society*. Cognitive Science Society Sapporo, 2012, pp. 1512–1517.

[8] A. R. Brown, "Generative music in live performance," in *Australian Computer Music Conference*. Brisbane, Australia: Australasian Computer Music Association, 2005, pp. 23–26.

[9] J. Freeman, "Extreme sight-reading, mediated expression, and audience participation: Real-time music notation in live performance," *Computer Music Journal*, vol. 32, no. 3, pp. 25–41, 2008.

[10] D.-H. Kim, E. Henrich, C. Im, M.-C. Kim, S.-J. Kim, Y. Li, S. Liu, S.-M. Yoo, L.-C. Zheng, Q. Zhou *et al.*, "Distributed computing based streaming and play of music ensemble realized through TMO programming," in *10th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems, 2005. WORDS 2005.* IEEE, 2005, pp. 129–136.

[11] D. Liang, G. Xia, and R. B. Dannenberg, "A framework for coordination and synchronization of media," in *Proc. of the Int. Conf. on New Interfaces for Musical Expression (NIME11)*, 2011, pp. 167–172.

[12] J. Cross, "eStand TM Electronic Music Stand (review)," *Notes*, vol. 60, no. 3, pp. 754–756, 2004.

[13] M. Kumarova, "Digital music stand," Aug. 2007, US Patent App. 11/587,180. [Online]. Available: http://www.google.com/patents/US20070175316

[14] T. Bell, D. Blizzard, R. D. Green, and D. Bainbridge, "Design of a Digital Music Stand," in *ISMIR*, 2005, pp. 430–433.

[15] O. Dasna, C. Graefe, J. Maguire, and D. Wahila, "muse: A Digital Music Stand for Symphony Musicians," *interactions*, vol. 3, no. 3, pp. 26–35, May/June 1996.

[16] J. Borchers, A. Hadjakos, and M. Mühlhäuser, "MICON: A Music Stand for Interactive Conducting," in *Proc. of the 2006 Int. Conf. on New Interfaces for Musical Expression (NIME06)*. Paris, France: IRCAM – Centre Pompidou, 2006, pp. 254–259.

[17] R. Picking, "Reading music from screens vs paper," *Behaviour & Information Technology*, vol. 16, no. 2, pp. 72–78, 1997.

[18] B. Bell, J. Kleban, D. Overholt, L. Putnam, J. Thompson, and J. Kuchera-Morin, "The multimodal music stand," in *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, ser. NIME '07. New York, NY, USA: ACM, 2007, pp. 62–65. [Online]. Available: http://doi.acm.org/10.1145/1279740.1279750

[19] P. Bellini, F. Fioravanti, and P. Nesi, "Managing music in orchestras," *Computer*, vol. 32, no. 9, pp. 26–34, 1999.

[20] E. Romero and G. Fitzpatrick, "Networked electronic music display stands," June 1998, US Patent 5,760,323.

[21] H. Connick, "System and method for coordinating music display among players in an orchestra," Feb. 2002, US Patent 6,348,648. [Online]. Available: http://www.google.com/patents/US6348648

[22] B. A. Laundry, "Sheet Music Unbound: A fluid approach to sheet music display and annotation on a multi-touch screen," Ph.D. dissertation, University of Waikato, 2011.

[23] N. Orio, S. Lemouton, and D. Schwarz, "Score following: State of the art and new developments," in *Proceedings of the 2003 Int. Conf. on New Interfaces for Musical Expression (NIME03)*. National University of Singapore, 2003, pp. 36–41.

[24] R. B. Dannenberg and C. Raphael, "Music score alignment and computer accompaniment," *Communications of the ACM*, vol. 49, no. 8, pp. 38–43, 2006.

[25] V. Thomas, C. Fremerey, M. Müller, and M. Clausen, "Linking Sheet Music and Audio—Challenges and New Approaches," in *Multimodal Music Processing*, ser. Dagstuhl Follow-Ups, Dagstuhl, Germany, 2012, vol. 3, pp. 1–22.

[26] T. Bell, A. Church, J. Mc Pherson, and D. Bainbridge, "Page turning and image size in digital music stand," in *International Computer Music Conference*, 2005.

[27] J. McPherson, "Page turning—Score Automation for Musicians," *Honours project report, Department of Computer Science, University of Canterbury, Christchurch, NZ*, 1999.

[28] A. Blinov, "An interaction study of a digital music stand," *Honours project report, Department of Computer Science and Software Engineering*, 2007.

[29] J. Kosakaya, Y. Takii, M. Kizaki, A. Esashi, and T. Kiryu, "Research and evaluation of a performer-friendly electronic music stand," in *Proc. of the Int. Conf. on Active Media Technology (AMT) 2005*. IEEE, 2005, pp. 11–15.

[30] J. Pagwiwoko, "Improvements To A Digital Music Stand," Master's thesis, University of Canterbury, Christchurch, New Zealand, Nov. 2008.

[31] I. Yasue, K. Susumu, and F. Yositomo, "Design and production of an electronic musical score system to reduce the load of page turning for wind orchestra," in *IEEE 13th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. IEEE, 2014, pp. 242–246.

[32] A. Friberg and J. Sundberg, "Does Music Performance Allude to Locomotion? A Model of Final Ritardandi Derived from Measurements of Stopping Runners," *The Journal of the Acoustical Society of America*, vol. 105, no. 3, pp. 1469–1484, March 1999.

A MUSIC PERFORMANCE ASSISTANCE SYSTEM BASED ON VOCAL, HARMONIC, AND PERCUSSIVE SOURCE SEPARATION AND CONTENT VISUALIZATION FOR MUSIC AUDIO SIGNALS

Ayaka Dobashi **Yukara Ikemiya** **Katsutoshi Itoyama** **Kazuyoshi Yoshii**
Department of Intelligence Science and Technology
Graduate School of Informatics, Kyoto University, Japan
{dobashi, ikemiya, itoyama, yoshii}@kuis.kyoto-u.ac.jp

ABSTRACT

This paper presents a music performance assistance system that enables a user to sing, play a musical instrument producing harmonic sounds (*e.g.*, guitar), or play drums while playing back a karaoke or minus-one version of an existing music audio signal from which the sounds of the user part (singing voices, harmonic instrument sounds, or drum sounds) have been removed. The beat times, chords, and vocal F0 contour of the original music signal are visualized and are automatically scrolled from right to left in synchronization with the music play-back. To help a user practice singing effectively, the F0 contour of the user's singing voice is estimated and visualized in real time. The core functions of the proposed system are vocal, harmonic, and percussive source separation and content visualization for music audio signals. To provide the first function, vocal-and-accompaniment source separation based on RPCA and harmonic-and-percussive source separation based on median filtering are performed in a cascading manner. To provide the second function, content annotations (estimated automatically and partially corrected by users) are collected from a Web service called Songle. Subjective experimental results showed the effectiveness of the proposed system.

1. INTRODUCTION

In our daily lives, we often enjoy music in an active way, *e.g.*, sing a song or play a musical instrument. Although only a limited number of commercial music CDs include accompaniment (karaoke) tracks, karaoke companies provide those tracks for most major songs. To attain this, every time a new CD is released, a music expert is asked to manually transcribe the music (make a MIDI file). One of the main issues of this labor-intensive approach is that the sound quality of accompaniment tracks generated by MIDI synthesizers is often far below that of the original tracks generated by real musical instruments. In addition, the karaoke tracks that are originally available are usually completely instrumental and do not include chorus voices. The situation is much worse for people who want to sing minor songs or play musical instruments because they cannot use any karaoke or minus-one recordings (music recordings without particular instrument parts).

Copyright: ©2015 Ayaka Dobashi 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.

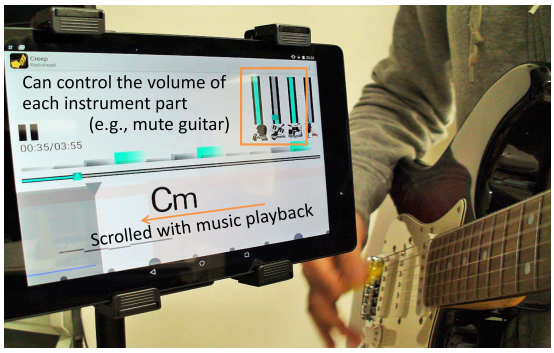


Figure 1. Example of how the proposed system is used: A user is playing a guitar with the playback of the other instrument parts (singing and drums) during seeing beat times and chord progressions displayed on a tablet.

In this paper we describe a music performance assistance system that enables a user to sing a song, play a harmonic musical instrument (*e.g.*, guitar), or play drums while playing back a minus-one version of an existing music recording (Fig. 1). When a user wants to sing a song, for example, the system plays back the accompaniment sounds by removing only predominant singing voices from the original recording (karaoke mode). An advantage of the proposed system is that accompaniment sounds it provides include chorus voices included in the original recording. Since the F0 of the user's singing voice is estimated and recorded in real time, the user can easily review his or her singing by comparing the F0 contour of the user's singing voice with that of the professional singing voice. When a user wants to play a harmonic instrument or drums, the system works similarly as it does in the karaoke mode. To further help a user, the beat times and chord progressions are displayed and automatically scrolled with the music playback. Since this system is implemented on a tablet computer, users can enjoy music in an active way anywhere.

To implement the system, we tackle two problems: vocal, harmonic, and percussive source separation (VHPSS) and content visualization for music audio signals. For the first, we propose a new method that combines vocal-and-accompaniment source separation (VASS) based on robust principal component analysis (RPCA) [1] and harmonic-and-percussive source separation (HPSS) based on median filtering [2] in a cascading manner. For the second, we col-

lect music-content annotations on music recordings from a Web service called Songle [3]. This service can automatically analyze four kinds of musical elements (beat times, chord progressions, vocal F0s, and musical structure) for arbitrary music audio signals available on the Web and visualize the analysis results on the user's browser. A key feature of Songle is that users can, as in Wikipedia, correct the analysis results if they find errors. Using this crowdsourcing Web service, the proposed system can keep the music content shown to users up-to-date.

2. RELATED WORK

This section reviews several studies related to our system in terms of three aspects: automatic accompaniment, active music listening, and sound source separation.

2.1 Automatic accompaniment

Automatic accompaniment systems using score information (MIDI data) of accompaniment parts have been developed for the two decades [4, 5]. Tekin *et al.* [6] and Pardo *et al.* [7], for example, proposed score following systems that can play back accompaniment sounds in synchronization with the performance of a user including tempo fluctuations and repeats of particular regions. Nakamura *et al.* [8] developed an improved system called Eurydice that can deal with repeats of arbitrary regions. Although some studies have tried to synchronize high-quality audio signals of accompaniment parts with user performances [9, 10], it is generally difficult to follow user performances played by polyphonic instruments (*e.g.*, piano). Mauch *et al.* [11] proposed a system called SongPrompter that can generate accompaniment sounds (drums and bass guitar) for any music audio signals without using the score information. To achieve this, the beat times and the F0s of bass lines are automatically estimated from music audio signals. The lyrics and chords given by a user are automatically synchronized with those signals and a display of the lyrics and chords is automatically scrolled in time to the music.

2.2 Active music listening

Active music listening [12] has recently been considered to be a promising research direction. "Active" means any active experience to enjoy listening to music (*e.g.*, touching-up music while playing it). Improved end-user's computing environments and music analysis techniques are making interaction with music more active. Goto *et al.* [3], for example, developed a Web service called Songle that helps a user better understand the content of a musical piece (repeated sections, beat times, chords, and vocal F0 contour) while listening to music by automatically estimating and visualizing the musical content. Yoshii *et al.* [13] proposed an audio player called Drumix that enables a user to intuitively customize drum parts included in the audio signal of a popular song without affecting the other sounds. Itoyama *et al.* [14] proposed a system that allows a user to control the volumes of individual instrument parts in real time by using a method of score-informed source separation. Yasuraoka *et al.* [15] proposed a method that enables a user to

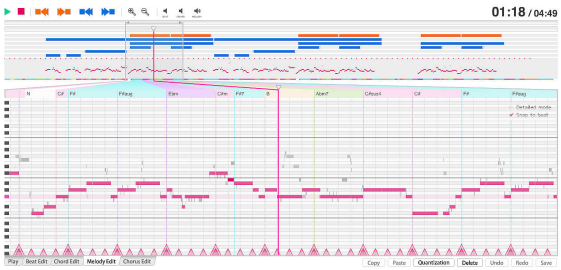


Figure 2. A screenshot of the web service called Songle: The repeated sections, beat times, chords, and vocal F0s of music audio signals are visualized on the browser.

freely edit a phrase of a particular instrument part in music audio signals while preserving the original timbre of the instrument. Fukayama and Goto [16] proposed a system that allows a user to mix the characteristics of chord progressions used in different music audio signals. Giraldo and Ramirez [17] proposed a system that changes the emotion of music in real time according to brain activity data detected by a brain-computer interface. Mancini *et al.* [18] proposed a system that, by analyzing user's motion, allows a user with mobile devices and environmental sensors to physically navigate in a physical or virtual orchestra space in real time. Chandra *et al.* [19] proposed a system that allows a group of participants with little or no musical training to play together in a band-like setting by sensing their motion with mobile devices. Tsuzaki *et al.* [20] proposed a system that assists a user to create derivative chorus music by mashing up multiple cover songs.

2.3 Source separation

A lot of effort has recently been devoted to vocal-and-accompaniment source separation (VASS) for music audio signals. Rafii and Pardo [21], for example, proposed a method called REPET that separates each short segment of a target music spectrogram into vocal components that significantly differ from those of the adjacent segments and accompaniment components that repeatedly appear in the adjacent segments. Liutkus *et al.* [22] generalized the concept of REPET in terms of kernel-based modeling by assuming that a source component at a time-frequency bin can be estimated by referring to other particular bins that are defined according to a source-specific proximity kernel. Huang *et al.* [23, 24] pioneered to use robust principal component analysis (RPCA) or deep neural networks for singing voice separation in an unsupervised or supervised manner. To improve the performance of VASS, Rafii *et al.* proposed a method that combines singing voice separation based on REPET with vocal F0 estimation. A similar method was proposed by Ikemiya *et al.* [1]. A key feature of this method is that only the singing voices corresponding to a predominant F0 contour are extracted and the other singing voices (*e.g.*, chorus voices) are separated as accompaniment sounds.

Several attempts have also been made to harmonic-and-percussive sound separation (HPSS). Yoshii *et al.* [13, 25] proposed a method that detects the onset times of drums by using a template adaptation-and-matching method and

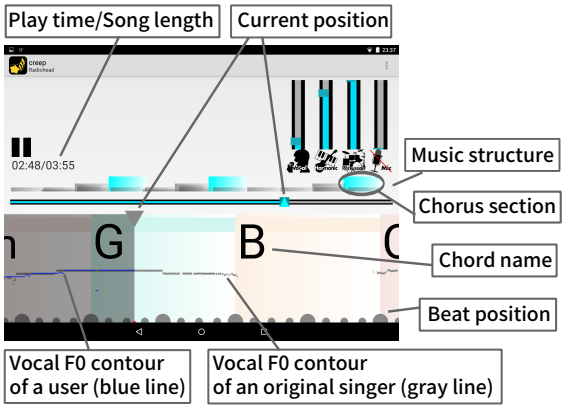


Figure 3. A screenshot of the user interface.

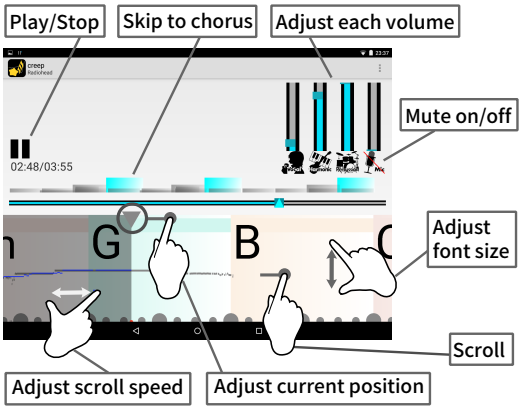


Figure 4. How to use the proposed system.

subtracts the drum sounds. Gillet and Richard [26] proposed a method that estimates a time-frequency subspace mask and then uses Wiener filtering. Rigaud *et al.* [27] proposed a method to extract drum sounds from polyphonic music by using a parametric model of the evolution of the short-time Fourier transform (STFT) magnitude. Miyamoto *et al.* [28] focused on the difference of isotropic characteristics between harmonic and percussive components and separated those components by minimizing a cost function. Fitzgerald *et al.* [2] also focused on the anisotropy, but used median filtering instead of a cost function.

3. USER INTERFACE

This section describes the GUI of the proposed system implemented on an Android tablet (HTC Nexus9). Figure 3 shows the components of the interface and Figure 4 shows how it is used. This interface provide two main functions: instrument-based volume control and music-content visualization. Although the proposed system is originally intended for music performance assistance, it is also useful for active music listening. Using the volume control function, users can listen to music while focusing on a particular instrument part. In addition, users can enjoy the music content visualized in real time as in the web service called Songle [3]. This helps a user better understand music and play a musical instrument in a musically meaningful and expressive manner.

3.1 Instrument-based volume control

The system allows a user to independently adjust the volumes of main vocals, harmonic instruments (including chorus vocals), and drums. In the upper right of the interface, three volume sliders corresponding to the different parts are provided. Another rightmost slider is used for controlling the volume of the microphone input in the karaoke mode (Figure 3). When the volume of a part the user sings or plays is turned down, the system plays back a karaoke or minus-one version of the original music audio signal. This function is useful for the practice purpose. When a vocalist of a typical rock band wants to practice a singing skill, for example, the volume of singing voices can be turned down. If the vocalist wants to sing a song and play a guitar simultaneously, the volumes of both singing voices and harmonic accompaniment sounds can be turned down. If all members of the rock band cannot meet together, it would be useful to play back only musical instrument parts corresponding to absent members. Because the system is implemented on a tablet computer and can be easily carried, it allows users to get sounds by the whole band whenever and wherever.

In the current implementation, vocal, harmonic, and percussive source separation (VHPSS) should be performed on a standard desktop computer in advance of using the volume control function. Since the computational power of recent embedded CPUs has rapidly grown, stand-alone VHPSS for any music audio signal stored on the tablet could be achieved in the near future.

3.2 Music content visualization

A display of chords and beat times is automatically scrolled in synchronization with the playback of the music. Since the chords, beat times, and vocal F0s are displayed at the bottom (Figure 3), a guitarist can play a guitar while watching the chords, and a vocalist can sing while checking his or her own F0s. The gray and blue lines on the chord pane are the vocal F0 contour of the original singing voices and that of the user's singing voices. The system allows a user to adjust the playback position by swiping horizontally on the chord pane (playback position can also be adjusted with the seek bar at the center). The system allows a user to adjust the display range of chord progressions by using a horizontal pinch in/out. A narrow display range allows a user to read each cord name clearly, while a broad one allows a user to read the following chords earlier. The playback speed is not changed by this operation. The system allows a user to adjust the font size of chord names by using a vertical pinch in/out. The triangle at the top of the chord pane indicates the current playback position. The system allows a user to adjust the location of the triangle by making a long press and horizontal swipe. Moving it to right allows a user to easily read the current chord and check the user's vocal F0s in real time, while moving it to the left allows a user to read the following chords easily.

A lot of overlapping rectangles over the central seek bar shows a hierarchical structure of a target music audio signal. The triangular mark on the seek bar shows where the current position on the structure is. The rectangles having

the same height indicate repeated sections. The light blue rectangles indicate chorus sections. When one of the blue rectangles is tapped on the screen, the playback position directly jumps to the start of the corresponding chorus section. This function helps a user practice playing the same section repeatedly.

4. SYSTEM IMPLEMENTATION

This section explains the technical implementation of the proposed system. The two main functions of the user interface described in Section 3 call for the development of vocal, harmonic, and percussive source separation (VHPSS) and automatic content analysis for music audio signals.

4.1 Source separation of music audio signals

We aim to separate music audio signals into singing voices, harmonic accompaniment sounds, and percussive sounds. To do this, the audio signals are first separated into singing voices and the other accompaniment sounds, which are further separated into harmonic sounds and percussive sounds. Figure 5 shows an overview of our approach.

4.1.1 Vocal and accompaniment source separation

We use the state-of-the-art method of singing voice separation [1] because it achieved the best performance in the singing voice separation track of MIREX 2014. As shown in Figure 6, robust principal component analysis (RPCA) is used for separating the amplitude spectrogram of a target music audio signal into a sparse matrix corresponding to singing voices and a low-rank matrix corresponsing to accompanying sounds. After a binary mask is made by comparing the two matrices in an element-wise manner, the vocal spectrogram is roughly obtained by applying the mask to the input mixture spectrogram.

The vocal F0 contour is then estimated by an extension of subharmonic summation (SHS) [1]. This method yields more accurate and smooth F0 contours than Songle. The following salience function $H(t, s)$ on a logarithmic scale is used [29]:

$$H(t,s) = \sum_{n=1}^{N} h_n P(t, s + 1200 \log_2 n), \tag{1}$$

where t is a frame index, s is a log-frequency [cents], $P(t, s)$ is the amplitude at time t and frequency s, N is the number of harmonic partials considered, and h_n is a partial weight. The A-weighting function considering the nonlinearity of the human auditory system is applied to the vocal spectrogram before computing $H(t, s)$, and the vocal F0 contour $\hat{S}$ is estimated by using the Viterbi algorithm as follows:

$$\hat{S} = \underset{S_1,\ldots,S_T}{\arg\max} \sum_{t=1}^{T-1} \{\log a_t H(t, s_t) + \log T(s_t, s_{t+1})\}, \tag{2}$$

where $T(s_t, s_{t+1})$ is a transition probability from the current F0 s_t to the next F0 s_{t+1} and a_t is a normalization factor. The basic SHS method without temporal continuity is also used for estimating the F0 contour of the user's singing voice in real time. A harmonic mask that passes only harmonic partials of given F0s is made on the assumption that the energy of vocal spectra is localized on harmonic partials. After the RPCA and harmonic masks are integrated, the vocal spectrogram is finally obtained by applying the integrated mask to the input spectrogram.

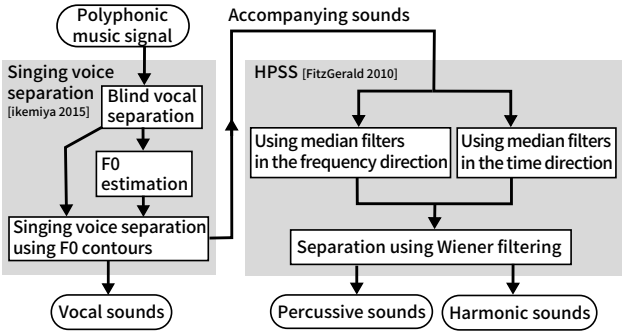


Figure 5. Source separation

4.1.2 Harmonic and percussive source separation

We use a method of harmonic and percussive source separation based on median filtering [2] because of its high performance, easy implementation and low computation cost.

The elements $P_{t,h}$, $H_{t,h}$ and $W_{t,h}$ of the percussive amplitude spectrogram $\mathbf{P}$, the harmonic spectrogram $\mathbf{H}$, and the given spectrogram $\mathbf{W}$ satisfy the following conditions:

1. $P_{t,h} \geq 0$;
2. $H_{t,h} \geq 0$;
3. $P_{t,h} + H_{t,h} = W_{t,h}$;

where t is a frame index and h is a frequency. As Figure 7 shows, this method focuses on the following observations:

1. Harmonic instrument sounds in a spectrogram are stable in the time-axis direction;
2. Percussive sounds in a spectrogram are stable in the frequency-axis direction;

Therefore it is possible to obtain $H_{t,h}$ and $P_{t,h}$ by removing the steep parts with median filters. Soft masks based on Wiener filtering are obtained by

$$M_{H_{t,h}} = \frac{H_{t,h}^p}{(H_{t,h}^p + P_{t,h}^p)}, \tag{3}$$

$$M_{P_{t,h}} = \frac{P_{t,h}^p}{(H_{t,h}^p + P_{t,h}^p)}, \tag{4}$$

where p is the power to which each individual element of the spectrograms is raised. Output spectrograms $\hat{\mathbf{H}}$ and $\hat{\mathbf{P}}$ are defined as follows:

$$\hat{\mathbf{H}} = \hat{\mathbf{S}} \otimes \mathbf{M_H} \tag{5}$$

$$\hat{\mathbf{P}} = \hat{\mathbf{S}} \otimes \mathbf{M_P} \tag{6}$$

where $\otimes$ represents element-wise multiplication and $\hat{\mathbf{S}}$ is the input mixture spectrogram.

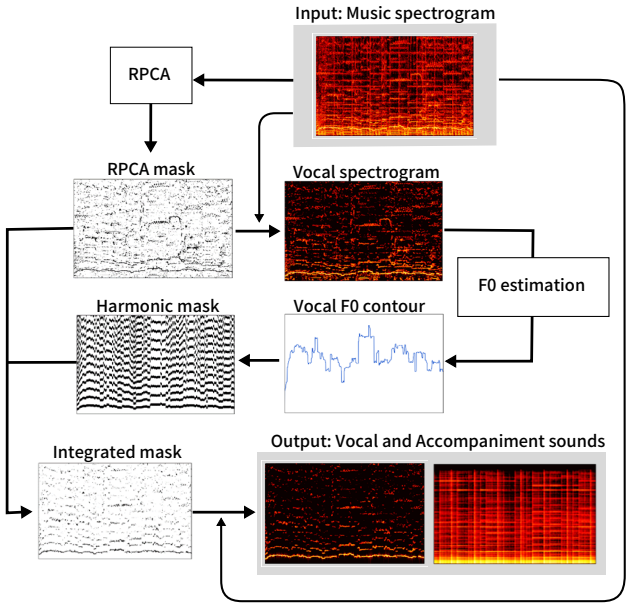


Figure 6. Vocal-and-accompaniment source separation

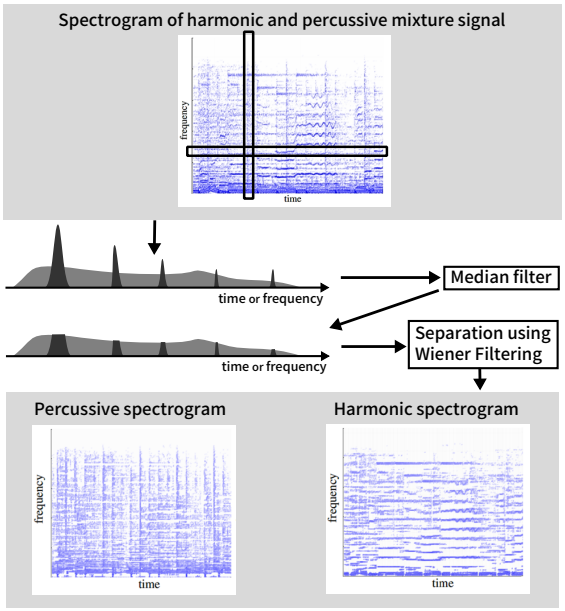


Figure 7. Harmonic-and-percussive source separation

4.2 Content analysis of music audio signals

The information for playing is obtained from Songle, which is a web service for active music listening [3]. It displays, for any music on the Web, automatic analysis results for various aspects, such as repeating structure, beat time, chords and vocal F0s. The user can correct errors by making annotations, so its accuracy gradually increases. By using this annotation mechanism, it is possible to sequentially update the data on a tablet.

5. EXPERIMENT

This section reports a subjective experiment conducted for evaluating the effectiveness of the proposed system.

5.1 Experimental conditions

A subject was asked to play a guitar according to the play-back of a Japanese popular song while using the proposed system. The subject was a 23-year-old university student who had played guitar for eight years. The effects on playing were examined with regard to three differences. More specifically, whether the system displayed information of music content or did not, whether the music-content information was correct or not, and whether or not the subject was allowed to adjust the volume of the guitar were investigated by observation and in interviews after the experiment. The detailed instructions were as follows:

1. Listen to the song;
2. Play guitar without performance support;
3. Play guitar with performance support.

The subject did as instructed under each of the following three conditions:

A) Chord and beat information were displayed or not (artist: Spitz, song: Cherry),

B) Automatic analysis results were corrected or not (artist: Perfume, song: polyrhythm),

C) Accompaniment volume was adjusted or not (artist: Aiko, song: Atashi no mukou).

Note that since music-content information is downloaded from Songle, estimation errors of music content are often included in an actual use if no users have corrected them.

5.2 Experimental results

The proposed system worked well as we expected and the perceptual quality of accompaniment sounds generated by the instrument-based volume control function reached a practical level. The result of the condition A showed that the visualization of chord information facilitated the music performance of the user. The result of the condition B indicated that although the visualization of automatic chord recognition results has some support effects, recognition errors often make it difficult to play the guitar in a comfortable way. This indicates the effectiveness of using Songle for keeping the music content shown to the user up-to-date.

Several kinds of improvements were suggested in terms of system usability. First, it would be better to show chord diagrams because unfamiliar chords often appear. Second, showing the highlights of a song would be helpful for planning a performance. A key transpose function would often be useful for making the performance easier [1] .

6. CONCLUSION

This paper presented a music performance assistance system based on vocal, harmonic, and percussive source separation of music audio signals. The beat times, chords, and vocal F0 contour are collected from Songle and are automatically scrolled from right to left in synchronization with the music play-back. To help a user practice singing effectively, the F0 contour of the user's singing voice is estimated and visualized in real time. The subjective experi-

[1] A demo video is available on http://winnie.kuis.kyoto-u.ac.jp/members/dobashi/smc2015/

mental results indicated that the system actually facilitates playing and increases a sense of play.

We plan to develop an intelligent function that follows the performance of a user including tempo fluctuations. In addition, we will tackle the implementation of all separation and analysis algorithms on a tablet computer.

Acknowledgments

This paper was partially supported by JST OngaCREST Project and KAKENHI No. 24220006, No. 26700020, and No. 26280089.

7. REFERENCES

[1] Y. Ikemiya, K. Yoshii, and K. Itoyama, "Singing voice analysis and editing based on mutually dependent F0 estimation and source separation," in *ICASSP*, 2015.

[2] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *DAFX*, 2010.

[3] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and N. Tomoyasu, "Songle: An active music listening service enabling users to contribute by correcting errors," *Interaction*, pp. 1363–1372, 2012.

[4] R. Dannenberg, "An on-line algorithm for real-time accompaniment," in *ICMC*, 1984, pp. 193–198.

[5] B. Vercoe, "The synthetic performer in the context of live performance," in *ICMC*, 1984, pp. 199–200.

[6] M. E. Tekin, C. Anagnostopoulou, and Y. Tomita, "Towards an intelligent score following system: Handling of mistakes and jumps encountered during piano practicing," in *CMMR*, 2005, pp. 211–219.

[7] B. Pardo and W. Birmingham, "Modeling form for on-line following of musical performances," in *AAAI*, 2005, pp. 1018–1023.

[8] E. Nakamura, H. Takeda, R. Yamamoto, Y. Saito, S. Sako, and S. Sagayama, "Score following handling performances with arbitrary repeats and skips and automatic accompaniment," *IPSJ Journal*, pp. 1338–1349, 2013.

[9] C. Raphael, "Music plus one: A system for flexible and expressive musical accompaniment," in *ICMC*, 2001, pp. 159–162.

[10] A. Cont, "ANTESCOFO: Anticipatory synchronization and control of interactive parameters in computer music," in *ICMC*, 2008, pp. 33–40.

[11] M. Mauch, H. Fujihara, and M. Goto, "SongPrompter: An accompaniment system based on the automatic alignment of lyrics and chords to audio," in *ISMIR*, 2010.

[12] M. Goto, "Active music listening interfaces based on signal processing," in *ICASSP*, 2007, pp. 1441–1444.

[13] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Drumix: An audio player with real-time drum-part rearrangement functions for active music listening," *Information and Media Technologies*, pp. 601–611, 2007.

[14] K. Itoyama, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Instrument equalizer for query-by-example retrieval: Improving sound source separation based on integrated harmonic and inharmonic models." in *ISMIR*, 2008, pp. 133–138.

[15] N. Yasuraoka, T. Abe, K. Itoyama, T. Takahashi, T. Ogata, and H. G. Okuno, "Changing timbre and phrase in existing musical performances as you like: manipulations of single part using harmonic and inharmonic models," in *ACM Multimedia*, 2009, pp. 203–212.

[16] S. Fukayama and M. Goto, "Harmonymixer: Mixing the character of chords among polyphonic audio," *ICMC-SMC*, pp. 1503–1510, 2014.

[17] S. Giraldo and R. Ramirez, "Brain-activity-driven real-time music emotive control," in *ICME*, 2013.

[18] M. Mancini, A. Camurri, and G. Volpe, "A system for mobile music authoring and active listening," *Entertainment Computing*, pp. 205–212, 2013.

[19] A. Chandra, K. Nymoen, A. Voldsund, A. R. Jensenius, K. H. Glette, and J. Tørresen, "Enabling participants to play rhythmic solos within a group via auctions," in *CMMR*, 2012, pp. 674–689.

[20] K. Tsuzuki, T. Nakano, M. Goto, T. Yamada, and S. Makino, "Unisoner: An interactive interface for derivative chorus creation from various singing voices on the web," *SMC*, 2014.

[21] Z. Rafii and B. Pardo, "Music/voice separation using the similarity matrix," in *ISMIR*, 2012, pp. 583–588.

[22] A. Liutkus, D. Fitzgerald, Z. Rafii, B. Pardo, and L. Daudet, "Kernel additive models for source separation," in *IEEE Trans. on Signal Processing*, 2014.

[23] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," *ISMIR*, 2014.

[24] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *ICASSP*, 2012, pp. 57–60.

[25] K. Yoshii, M. Goto, and H. G. Okuno, "Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression," *IEEE Trans. on Audio, Speech, and Language Processing*, pp. 333–345, 2007.

[26] O. Gillet and G. Richard, "Transcription and separation of drum signals from polyphonic music," *IEEE Trans. on Audio, Speech, and Language Processing*, pp. 529–540, 2008.

[27] F. Rigaud, M. Lagrange, A. Robel, and G. Peeters, "Drum extraction from polyphonic music based on a spectro-temporal model of percussive sounds," in *ICASSP*, 2011, pp. 381–384.

[28] K. Miyamoto, H. Kameoka, N. Ono, and S. Sagayama, "Separation of harmonic and non-harmonic sounds based on anisotropy in spectrogram," in *Acoustical Society of Japan Autumn conference*, 2008, pp. 903–904.

[29] D. J. Hermes, "Measurement of pitch by subharmonic summation," *Journal of the acoustical society of America*, pp. 257–264, 1988.

A SCORE-INFORMED PIANO TUTORING SYSTEM WITH MISTAKE DETECTION AND SCORE SIMPLIFICATION

Tsubasa Fukuda Yukara Ikemiya Katsutoshi Itoyama Kazuyoshi Yoshii
Graduate School of Informatics, Kyoto University
{tfukuda, ikemiya, itoyama, yoshii}@kuis.kyoto-u.ac.jp

ABSTRACT

This paper presents a novel piano tutoring system that encourages a user to practice playing a piano by simplifying difficult parts of a musical score according to the playing skill of the user. To identify the difficult parts to be simplified, the system is capable of accurately detecting mistakes of a user's performance by referring to the musical score. More specifically, the audio recording of the user's performance is transcribed by using supervised non-negative matrix factorization (NMF) whose basis spectra are trained from isolated sounds of the same piano in advance. Then the audio recording is synchronized with the musical score using dynamic time warping (DTW). The user's mistakes are then detected by comparing those two kinds of data. Finally, the detected parts are simplified according to three kinds of rules: removing some musical notes from a complicated chord, thinning out some notes from a fast passage, and removing octave jumps. The experimental results showed that the first rule can simplify musical scores naturally. The second rule, however, often simplified the scores awkwardly when the passage formed a melody line.

1. INTRODUCTION

Thanks to the recent development of audio signal analysis technology, many applications have appeared that enable users to practice playing musical instruments without the guidance of a teacher. A system called SongPrompter [1], for example, automatically displays the information (*e.g.*, chord progression, tempo, lyrics) for assisting a user to play a guitar and sing a song. An application [2] estimates the chord progressions of user's favorite songs taken from an iPhone or iPod and creates chord scores for them.

In this paper we propose a novel piano tutoring system that can detect mistakes of piano performances and simplify the difficult parts of musical scores[1] because the piano is one of the most popular musical instruments. Although players at an intermediate level want to play their favorite musical pieces, the scores of those pieces are often difficult for those players to play, causing them to lose their motivations. One of the effective solutions for this problem is to simplify the musical scores so that the difficulty of those scores matches the user's playing skills. To effectively assist a user to improve his or her playing skill, it is important to gradually increase the difficulty level of a musical score to recover the original difficulty level by changing the score simplification level. As the first step toward this goal, in this paper we focus on how to simplify mistakenly-played parts of musical scores.

The proposed system takes an audio signal of users' actual performance and the original score as inputs, and outputs a piano roll that shows user's mistakes and a simplified version of the original score. First, two piano rolls are created from the input audio signal and musical score respectively. More specifically, one is converted from the audio signal by using a multipitch estimation method based on nonnegative matrix factorization (NMF), and the other is obtained by synchronizing the original score with the users' performance with dynamic time warping (DTW). User's mistakes are then detected by comparing these two piano rolls, and the original score is simplified in accordance with the parts in which the mistakes are detected. We define three kinds of rules for simplifying musical scores and how to apply those rules.

Two experiments using actual music performances were conducted to evaluate the performance of the proposed system. The first experiment focused on the accuracy of mistake detection. Although double or half octave errors are the main cause of decreasing accuracy in multipitch estimation, those errors can be ignored for the purpose of detecting performance mistakes because it is rare for a user to mistakenly play double- or half-pitch notes. The second experiment focused on the effectiveness of score simplification. The results showed that some musical notes can be removed naturally from complicated chords and that removing musical notes from fast passages should be avoided when the passage constituted a melody line.

2. RELATED WORK

This section introduces related work on multipitch estimation and score simplification.

2.1 Multipitch estimation and mistake detection

It is necessary for revealing a user's weak points to detect mistakes by comparing the result of multipitch estimation and the original score.

Copyright: ©2015 Tsubasa Fukuda Yukara Ikemiya Katsutoshi Itoyama Kazuyoshi Yoshii 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] A demo video is available on http://winnie.kuis.kyoto-u.ac.jp/members/tfukuda/smc2015/

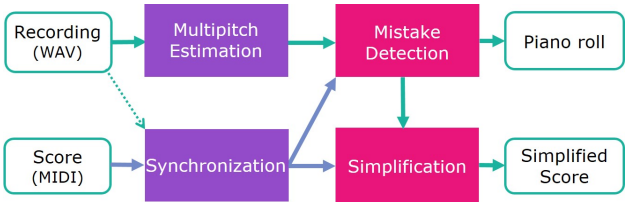


Figure 1. An overview of the proposed system

Tsuchiya *et al.* [3] proposed a novel Bayesian model that combines acoustic and language models for automatic music transcription. They tested the model on the RWC music database [4] and showed the result of transcription. They categorized transcription errors into three types, that is, deletion errors, pitch errors, and octave errors. As shown in the result of an experiment, octave errors are the majority of detected errors.

Azuma *et al.* [5] proposed a method of automatic transcription for a piano performance with both hands by focusing on harmonic structures in the frequency domain. This method automatically separates an obtained score into melody and accompaniment parts by focusing on the probability of the pitch transition. This system takes only an audio signal as an input, and many octave errors occur in the result of transcription.

Emiya *et al.* [6] and Sakaue *et al.* [7] showed that modeling the harmonic structures of musical instruments improves the accuracy of multipitch estimation. Since piano sounds also have the harmonic structures, the accuracy improvement is expected by integrating the prior information of harmonic structures into our system.

Benetos *et al.* [8] proposed a score-informed transcription method for automatic piano tutoring. Although the F-measure of automatic transcription was about 95%, many octave errors occurred. Our main contributions is to take into account those errors for mistake detection and to combine mistake detection [8] with score simplification for effectively assisting a user to practice playing the piano.

2.2 Score simplification

Simplifying a musical score according to the player's skill motivates him or her to practice the piano effectively. Just removing the notes from the score is, however, insufficient, for the score simplification. Since it is necessary to preserve the characteristics of the original score, how to simplify the score is a very important problem.

Yazawa *et al.* [9] proposed a method of guitar tablature transcription from audio signals. In this method, playing difficulty costs are given to several features such as positions of player's hands, the number of fingers used, and the migration length of the wrist. Then, it creates a tablature matching player's skills based on the cost.

Fujita *et al.* [10] proposed a method that modifies a musical score consisting of several instrument parts according to the player's skill. Player's skills are categorized into three types, and simplification is done based on four factors *i.e.*, the number of simultaneous notes, the width of a chord, the width of a passage, and the tempo of a passage.

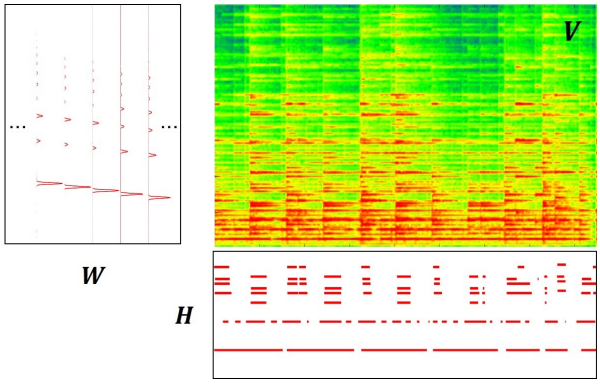


Figure 2. Multipitch estimation using NMF. An activation matrix H is calculated from an input spectrogram V by using a pretrained basis matrix W.

3. PROPOSED SYSTEM

This section describes a proposed system that can simplify difficult parts of a musical score according to the playing skill of the user. An overview of the proposed system is shown in Figure 1. The inputs are 1) an audio recording of a user's piano performance and 2) the original musical score. The outputs are 1) a piano roll indicating mistakes and 2) a simplified score.

The score is simplified by first calculating an activation matrix from the input recording using non-negative matrix factorization (NMF). The activation matrix is then converted into a piano roll by thresholding. The musical score is next synchronized with the audio recording by stretching the onset times and duration of the musical notes using dynamic time warping (DTW). The synchronized score is then converted into a piano roll.

Mistakes are detected by comparing the two synchronized piano rolls. Detection accuracy is improved by ignoring the octave errors that rarely occur during actual playing. The parts where mistakes were detected are classified into three patterns and simplified in accordance with predefined simplification rules.

3.1 Multipitch estimation

The user's playing performance is evaluated using the result of multipitch estimation. The input is the recording, and the output is a piano roll of the recording.

The estimation is done using the NMF algorithm with β-divergence [11]. This algorithm factorize a matrix $\boldsymbol{V}$ is factorized into two matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ ($\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H}$) that have no negative elements.

First, the recording is converted into a spectrogram as a matrix $\boldsymbol{V} \in \mathbb{R}^{f \times n}$ using constant-Q transform (CQT) [12], which has 24 frequency bins per octave and can handle frequencies as low as 60 Hz. The NMF algorithm takes matrix $\boldsymbol{V}$ as input and factorizes it into $\boldsymbol{W}$ and $\boldsymbol{H}$. Here, $\boldsymbol{W} \in \mathbb{R}^{f \times 88}$ is the base spectrum matrix. It consists of 88 base spectra from A0 to C8. The activation matrix is $\boldsymbol{H} \in \mathbb{R}^{88 \times n}$. It contains the amplitudes of the base spectra. NMF typically factorizes $\boldsymbol{V}$ by iteratively updating $\boldsymbol{W}$ and $\boldsymbol{H}$. Since $\boldsymbol{W}$ is fixed here, only $\boldsymbol{H}$ is updated in

accordance with the following rule:

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \otimes \frac{\boldsymbol{W}^T((\boldsymbol{V} \otimes \boldsymbol{W}\boldsymbol{H})^{\beta-2})}{\boldsymbol{W}^T(\boldsymbol{W}\boldsymbol{H})^{\beta-1})}, \tag{1}$$

where $\otimes$ is the element-wise product, the exponentiation is element-wise exponential and the fraction means element-wise division. We used $\beta = 0.6$, which has been shown to produce the best multipitch estimation of piano sounds in previous studies [8, 11, 13].

Base spectrum matrix $\boldsymbol{W}$ is estimated in advance from the sound of each pitch using an electronic piano. Application of CQT to each recording produces 88 spectrograms $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{88}$. NMF with a single basis spectrum is applied to each $\boldsymbol{X}_i$, *i.e.*, $\boldsymbol{X}_i \in \mathbb{R}^{f \times n}$ is factorized into vectors $\boldsymbol{w}_i \in \mathbb{R}^{f \times 1}$ and $\boldsymbol{h}_i \in \mathbb{R}^{1 \times n}$ as follows:

$$\boldsymbol{X}_i = \boldsymbol{w}_i \boldsymbol{h}_i. \tag{2}$$

Base spectrum matrix $\boldsymbol{W}$ is finally obtained by concatenating these vectors horizontally as follows:

$$\boldsymbol{W} = [\boldsymbol{w}_1 \boldsymbol{w}_2 \cdots \boldsymbol{w}_{88}]. \tag{3}$$

After update rule (1) converges, a piano roll of the recording is obtained by thresholding activation matrix $\boldsymbol{H}$ appropriately. An example of the NMF results is shown in Figure 2. The sample musical piece is from the RWC music database.

3.2 Score-to-audio synchronization

We obtained the audio recording using a YAMAHA P-255 electronic piano which can record an actual performance as an audio signal. There were several temporal gaps between the musical score and the actual performance no matter how exactly it was played. If such gaps are counted as mistakes, true mistakes cannot be detected appropriately. We avoid this problem by synchronizing the musical score with the recording in advance. The inputs are 1) the recording and 2) the musical score, and the outputs are the corresponding synchronized piano rolls.

For synchronization, we use the dynamic time warping (DTW) algorithm of Muller [14] to measure the similarity between two temporal sequences. Since this algorithm is a kind of dynamic programming, it can obtain the optimum solution, which means that the temporal correspondence between these sequences can be obtained by using it. Specifically, inputs are converted into spectrograms in advance. These spectrograms are next converted into chroma vectors, which are the 12-dimensional vectors. A chroma vector has the amplitude of each pitch name (C, C#,. . .,B), and the cosine distance of two chroma vectors are used as the distance in DTW.

3.3 Mistake detection

Comparing the piano roll created from the recording with the synchronized piano roll reveals where the user played the song incorrectly. The inputs are 1) the piano roll from the recording and 2) the synchronized piano roll, and the output is a piano roll comparing the inputs. The system compares the two input scores and indicates where the user made mistakes in his or her performance. An example output piano roll is shown in Figure 3. The black marks correspond to the notes the user played, and red marks correspond to notes in the original score. There were two mistakes in this example, as shown by the two blue marks.

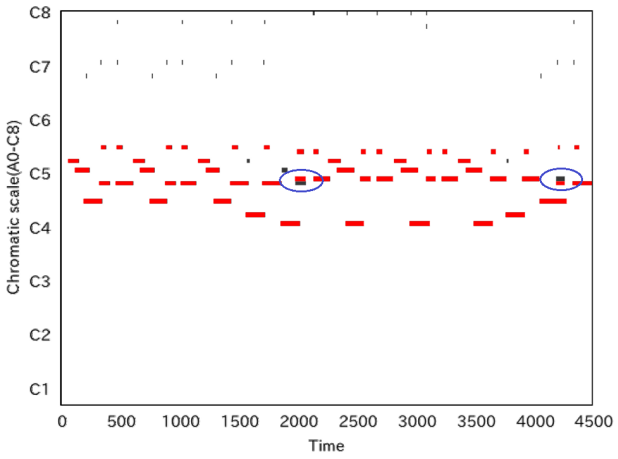


Figure 3. Example of mistake detection. Red marks correspond to the notes in an original score and black marks correspond to the notes estimated by NMF.

The system judges the weak points in the user's performance on the basis of where the mistakes are in the recording. Since octave errors often occur in multipitch estimation using NMF, the system sometimes misjudges the location of the mistakes. In fact, many short notes (around C7 and C8) that were not actually played by the user are detected in Figure 3. Since playing these notes rarely occur in the actual performance of a piano solo, we ignore octave errors to improve the accuracy of the multipitch estimation. Specifically, a detected mistake is ignored if there is another note whose pitch differs from the pitch of the detected mistake by octaves.

3.4 Score simplification

Simplifying the difficult scores to match the user's playing skills helps motivate the user to practice the piano. The inputs are 1) the original score and 2) the parts to be simplified, and the output is the simplified score.

In this part, scores are classified under three patterns, and are simplified according to simplification rules given in advance for each pattern.

Pattern 1. Parts with many notes to be played at the same time

Pattern 2. Parts that require fast fingering

Pattern 3. Parts that have adjacent notes, one is over an octave distant from the other.

Here we describe simplification process by using examples.



Figure 4. Example of pattern 1. Removing some notes from chords.

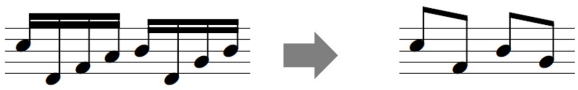


Figure 5. Example of pattern 2. Removing some notes from a part that requires fast fingering.

3.4.1 Pattern 1: Simplifying chords

When there are many notes to be played at the same time, that part is simplified removing some notes from the chords, as shown in Figure 4. Priority is given to each pitch of the chord, and the notes with lower priority are removed.

More specifically, the melody line often consists of a note with the highest pitch of the chord and is one of the most important notes. A note with the lowest pitch of the chord, called the root of the chord, is also important. These two notes are especially important and thus should not be removed, as shown in previous study [15]. The other notes are less important and can be removed if necessary. As a result, chords that are difficult to play are simplified.

3.4.2 Pattern 2: Simplifying fast passages

A part that requires fast fingering is simplified by removing the sequential notes that are to be played faster than a threshold, as shown in Figure 5.

3.4.3 Pattern 3: Removing octave jumps

Parts that have adjacent notes, one is over an octave distant from the other are called "leaps" and are further classified into two patterns.

3-A There is another leap after the leap.

3-B There is no leap after the leap.

In pattern 3-A, a note that generates the leap is difficult to play, so it would be removed as shown in Figure 6 (a).

In pattern 3-B, notes around the leap are removed in accordance with the rule for pattern 1 or 2. That is, if there is a chord around the leap, it is simplified, and if there is a fast passage around the leap, it is simplified as shown in Figure 6 (b).

4. EVALUATION

This section reports two experiments that were conducted for evaluating the performance of score-informed multipitch estimation and score simplification.

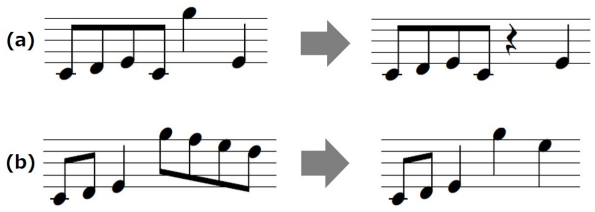


Figure 6. Examples of pattern 3. Removing some notes around a leaping.

Octave errors	Precision	Recall	F-measure
NOT ignore	0.943	0.988	0.965
ignore	0.995	0.988	0.991

Table 1. Accuracy of multipitch estimation

4.1 Multipitch estimation

We calculated the accuracy of multipitch estimation by comparing two piano rolls. One was obtained by analyzing the recording of an actual performance, and the other was created from an original musical score. The audio signals were recorded using a YAMAHA P-255 electronic piano played by a intermediate player, and were converted into a spectrogram using CQT which had 24 frequency bins per octave. This spectrogram was then factorized into a basis spectrum matrix and an activation matrix by using NMF. The activation matrix was finally converted into a piano roll by thresholding. On the other hand, pitch, onset time, and duration of each note were obtained by the original score, and the correct piano roll was created by synchronizing with the actual performance using DTW.

As shown in Table 1, the F-measure was calculated by comparing those piano rolls while ignoring octave errors and in the not case by way of comparison. About first ten seconds of *The Flea Waltz* was used as test data. According to the result, the F-measure was improved by ignoring octave errors. Using an appropriate value in thresholding helps to obtain the high accuracy.

We plan to employ a more reliable method for binarization of an activation matrix that is obtained by NMF. More specifically, a hidden Markov model (HMM) can be employed instead of thresholding. This model helps to reduce very short notes that are often occurred as octave errors in the result of NMF algorithm.

4.2 Score simplification

We evaluated the effectiveness of score simplification. The Grande valse brillante in E-flat major, Op. 18 and Étude Op. 10, No. 12 were used for this evaluation. Here, we tried simplifying parts that were selected at random on the assumption that the parts were played incorrectly by a user.

As simplifying a score, we prepared the score data that has pitch, onset time, and duration of each note. This time we chose the last twenty seconds of the sample music and simplified them. Simplified scores are shown in Figure 7

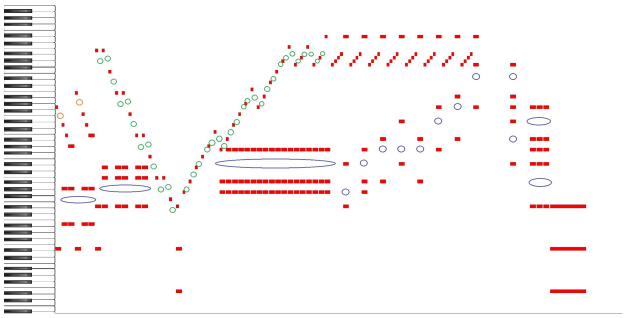


Figure 7. Simplified score of The Grande valse brillante in E-flat major, Op. 18. Blue marks correspond to the notes simplified in pattern 1, green ones correspond to pattern 2, and orange ones correspond to pattern 3.

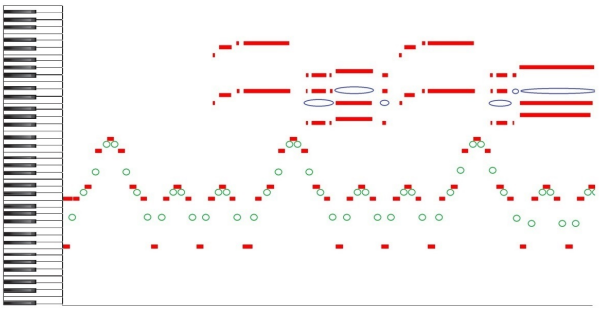


Figure 8. Simplified score of Étude Op. 10, No. 12

and Figure 8. In these figures, blue marks correspond to the notes simplified in pattern 1, green ones correspond to pattern 2, and orange ones correspond to pattern 3.

According to the result, we found that the simplification was correctly done. It was felt that something was a little off about simplification in pattern 2 and there was room for improvement. In patterns 1 and 3, it was felt that the result of simplification was naturally done. We will focus on appropriateness for the rules of simplification and automation of them in future work.

5. CONCLUSION

Our score-informed piano tutoring system with mistake detection and score synchronization works well by analyzing a recording of an actual performance. Intermediate players are often faced by a problem that the scores of their favorite pieces are often difficult to play and this makes them lose their motivations. The proposed system helps those players effectively continue to practice playing the piano. It detects the parts of the score that should be simplified so that the user can easily play those parts. Since the system can use the audio recording of a user's performance, it is unnecessary to set detailed parameters. Since the kinds of scores in which playing errors are likely to be occurred are identified to a certain level, the proposed system categorizes those errors into three types and simplifies the score according to predefined rules for each type.

Possible future works on the proposed system are as follows:

- Carry on additional experiments
- Improve each algorithm
- Improve score simplification

First, the amount of experiments is insufficient and there is a possibility that the evaluation is incorrect. We have to carry on additional experiments for various conditions to confirm that the evaluation is appropriate. It is also necessary to carry on an experiment through the whole system.

Second, improving each algorithm is necessary. DTW, employed in the synchronization, obtains optimal solution, but is a bit computationally expensive. An alternative solution is to use windowed time warping (WTW) [16]. Although this algorithm requires the distant paths to be contained in the correct paths, this requirement is met between an actual performance and the original score.

Finally, score simplification could be improved by using a wide variety of criteria. Future work will focus on the fingering to detect the notes that are difficult to play.

Acknowledgments

This study was supported in part by the OngaCREST project and JSPS KAKENHI 24220006, 26700020, and 24700168.

6. REFERENCES

[1] M. Matthias, H. Fujihara, and M. Goto, "SongPrompter: An accompaniment system based on automatic alignment of lyrics and chords," in *ISMIR2010*, 2010.

[2] CASIO COMPUTER CO., LTD., "Chordana viewer," http://world.casio.com/emi/app/ja/viewer/, 2015.

[3] H. Kameoka, K. Ochiai, M. Nakano, M. Tsuchiya, and S. Sagayama, "Context-free 2D tree structure model of musical notes for Bayesian modeling of polyphonic spectrograms." in *ISMIR2012*, 2012, pp. 307–312.

[4] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical and jazz music databases." in *ISMIR2002*, vol. 2, 2002, pp. 287–288.

[5] M. Azuma and W. Mitsuhashi, "Automated transcription for polyphonic piano music with a focus on harmonics in log-frequency domain," *IPSJ SIG Technical Reports*, vol. 89, no. 28, pp. 1–6, 2011.

[6] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. on Audio, Speech, and Lang. Process.*, vol. 18, no. 6, pp. 1643–1654, 2010.

[7] D. Sakaue, K. Itoyama, T. Ogata, and H. G. Okuno, "Initialization-robust multipitch estimation based on latent harmonic allocation using overtone corpus," in *ICASSP2012*, 2012, pp. 425–428.

[8] E. Benetos, A. Klapuri, and S. Dixon, "Score-informed transcription for automatic piano tutoring," in *EUSIPCO 2012*, 2012, pp. 2153–2157.

[9] K. Yazawa, D. Sakaue, K. Nagira, K. Itoyama, and H. G. Okuno, "Audio-based guitar tablature transcription using multipitch analysis and playability constraints," in *ICASSP2013*. IEEE, 2013, pp. 196–200.

[10] K. Fujita, H. Oono, and H. Inazumi, "A proposal for piano score generation that considers proficiency from multiple part," *IPSJ SIG Technical Reports*, vol. 77, no. 89, pp. 47–52, 2008.

[11] A. Dessein, A. Cont, and G. Lemaitre, "Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence," in *ISMIR2010*, 2010, pp. 489–494.

[12] C. Schörkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in *SMC 2010*, 2010, pp. 3–64.

[13] J. Fritsch and M. Plumbley, "Score informed audio source separation using constrained nonnegative matrix factorization and score synthesis," in *ICASSP2013*, 2013, pp. 888–891.

[14] M. Müller, "Dynamic time warping," in *Information Retrieval for Music and Motion*. Springer, 2007, pp. 69–84.

[15] G. Hori, H. Kameoka, and S. Sagayama, "Input-output HMM applied to automatic arrangement for guitars," *Information and Media Technologies*, vol. 8, no. 2, pp. 477–484, 2013.

[16] R. Macrae and S. Dixon, "Accurate real-time windowed time warping." in *ISMIR 2010*, 2010, pp. 423–428.

Target-Based Rhythmic Pattern Generation and Variation with Genetic Algorithms

Cárthach Ó Nuanáin, Perfecto Herrera and Sergi Jordà
Music Technology Group
Universitat Pompeu Fabra
Barcelona, Spain
carthach.onuanain@upf.edu

ABSTRACT

Composing drum patterns and musically developing them through repetition and variation is a typical task in electronic music production. We propose a system that, given an input pattern, automatically creates related patterns using a genetic algorithm. Two distance measures (the Hamming distance and directed-swap distance) that relate to rhythmic similarity are shown to derive usable fitness functions for the algorithm. A software instrument in the *Max for Live* environment presents how this can be used in real musical applications. Finally, a user survey was carried out to examine and compare the effectiveness of the fitness metrics in determining rhythmic similarity as well as the usefulness of the instrument for musical creation.

1. INTRODUCTION

When composing drum tracks in electronic music, it is typical that a producer begins with a basic loop or pattern which is then iterated on. Depending on the nature of the music, the amount of variation that the producer imparts can range from very little to quite a considerable amount. Contrast for example, the subtle changes that occur in stylistically minimal techno to the constantly shifting, complex patterns prevalent in IDM (Intelligent Dance Music).

Finding ways of automating this kind of activity can be useful for producers. For example, it could help to quickly lay a rhythmic foundation for a work in progress track, allowing the producer to focus on the "bigger picture", or provide an intelligent agent that accompanies a laptop performer in live situations. To address this, we outline a method that, when given a target input drum pattern, generates patterns with increasing similarity to the input pattern by using genetic algorithms.

Genetic Algorithms are a class of algorithm for solving search problems inspired by the biological metaphor of evolution. The core operation of genetic algorithms entails the generation of populations of potential solutions by simulating the process of "natural selection". Fitter candidates are selectively paired together to spawn new offspring using crossover and mutation. This approach is useful in situations where the search space is large and prohibitively expensive to search exhaustively by more conventional techniques such as depth-first and breadth-first search.

Genetic algorithms have been shown to assist in computer music composition. *GenJam* by Al Biles is perhaps one of the most well-known early realisations of this, whereby solo phrases in the jazz idiom are generated continuously [1]. One of the attractive aspects of genetic algorithms is the possibility of a human interactively appraising the output of the algorithm as it progresses. This is done in hope of reconciling the domain agnostic "objective" operation of the algorithm with the subjective, artistic goals of the critic.

The problem with this, of course, is the complexity of analysing many candidates from the algorithm and, specifically for temporal domains like music, sequential analysis of those candidates. This is known as the "fitness bottleneck" [2]. Solutions to the fitness bottleneck are dealt with in myriad ways. Biles, for example, proposes the elimination of the role of the fitness function completely, just preserving the aspects of crossover and mutation [3].

In the approach we describe, we derive our fitness function formally and programmatically by evaluating the similarity of each candidate pattern with respect to a target pattern which the composer determines initially. To compare two rhythmic patterns to each other in terms of similarity, the algorithm needs a distance function that can produce a value accordingly. Based on a review of the literature dealing with perceptual rhythmic similarity, we chose and implemented two such measures, namely the Hamming distance and the directed-swap distance. In Section 2 these will be explained in more detail and the evaluation section discusses how the participants responded to them in the user survey. To our knowledge, this is the first reported research that integrates rhythmic similarity perception with genetic algorithms for pattern generation.

The structure of the paper is as follows. The next section will examine the state of the art in musical genetic algorithms and similarity measures for rhythmic patterns. The methodology section explains our approach and shows the operation of the software instrument designed. Results of the user evaluation present our findings regarding the efficacy of the similarity measures and the musical output.

Copyright: ©2015 Cárthach Ó Nuanáin, Perfecto Herrera and Sergi Jordà 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.

2. EXISTING RESEARCH

2.1 Genetic Algorithms and Rhythm

A number of papers dealing with genetic algorithms and rhythm are presented here, and specific attention is directed to the derivation of a fitness function in each case.

Eigenfeldt [4] describes his Kinetic Engine: a software component that generates rhythms in general, not specific to drum sounds. His approach to fitness evaluation is perhaps a controversial one but not uncommon in musical applications. As in Al Biles' *GenJam* system, the role of fitness is simply eliminated. Thus the only real elements of genetic algorithms conserved are crossover and mutation. One may rightfully suspect that such a simplification renders the algorithm commensurate with a randomised search. Consequently a common solution used in such scenarios is to seed the initial population with a known dataset of good input. Another approach could be to embed some rules-based logic that restricts what type of candidates can be generated legally in the seeding process, as used by one of the authors of this paper in [5].

Bernardes et al. approach genetic drum pattern generation from the point of view of style emulation with statistical analysis of existing musical material [6]. Like Eigenfeldt, they do not incorporate a fitness function, but choose to perform prior analysis on user-supplied or preset MIDI files. This process gathers a probability distribution of weighted possible onset times in a 16-step pattern. The population is then seeded with candidates that have patterns generated according to this distribution.

A short paper by Horowitz suggests a multi-dimensional objective fitness function based on the combination of functional evaluations of syncopation, density, downbeat, beat repetition etc. [7]. Unfortunately the publication is rather scant on actual details regarding the implementation and evaluation of such a method.

Kaliakatsos–Papakostas et al. also use the notion of a a target pattern in evoDrummer [8], and define their fitness function by determining what they refer to as divergence in terms of "mean relative distance" to a base rhythm. The distance is computed based on a set of 40 features extracted from patterns, including descriptions of density, syncopation and loudness intensity. Indeed this is perhaps the most related work to our proposed system. However, to concentrate on the impact of distance measures between two simple patterns of onsets we have refrained from integrating such extensive feature vectors. The next section takes a look at various measures of rhythmic similarity in more detail.

2.2 Measures of Rhythmic Similarity

The most basic measure of similarity between two strings in general is the notion of edit distance. The edit distance defines the number of discrete insertion, deletion and substitution operations required to make one string match another [9]. Substitutions tend to be the most economical. For strings of the same length, and by restricting the operations to substitutions, this then corresponds to the Hamming distance, or simply the number of positions in which they differ [9]. Furthermore, if the strings are binary the distance can be computed quickly with the logical XOR operation. This is useful as chromosome representations in genetic algorithms are frequently represented in this manner, which we will see in more detail in the next section. Paiement et al. [10] have reported the distance to outperform Hidden Markov Models in distance modelling of rhythmic data, and suggest that derived models could be used for drum machines.

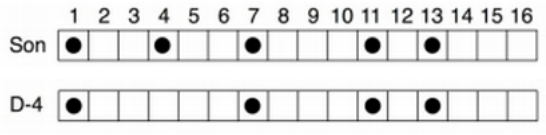


Figure 1. Son Pattern and a Variation [1]

Post and Toussaint have investigated the perceptual merit of the edit distance as a measure of rhythmic similarity [9]. It is compared to the swap distance, which considers the adjacent distance cost of mapping onsets between patterns. A swap cost of 1 is assigned for every adjacent movement an onset makes from its original position to a new position. An additional stipulation the swap distance model makes is that for two patterns with different number of onsets, every onset from the pattern with the greater number of onsets must be mapped onto an onset in the pattern with less onsets. For example, the cost of mapping the clave son (the "clave" is a fundamental rhythmic motif in Latin music) to the pattern D-4 (Fig. 1 above) would be 3, since onset 2 at position 4 needs to move 3 positions to onset 1 in position 1 or onset 2 in position 7 of D-4. Informally one would suppose that such a measure may better reflect the level of syncopation between two patterns.

Overall Toussaint concluded that the edit distance was a more robust measure of similarity that correlated quite well with listener judgements, when compared to the swap distance. This paper examines how these two measures can be applied to creating an automatic fitness function for a genetic algorithm that creates rhythm patterns. The next section will describe how this is implemented.

3. METHOD

This section introduces the tools we created in our research, namely *SimpleGA*: the genetic algorithm itself and *GenDrum*: the final *Max for Live* instrument. This is followed by a detailed discussion of the implementation of the distance measures and representational issues. Finally there is a description of the design of the experiment for evaluating the research.

3.1 Software Implementation

SimpleGA is a basic, general purpose genetic algorithm external object we developed for the *Pure Data* [11] and *Max/MSP* environments in C++ using the *Flext* framework [12]. It can handle binary, numerical and alphanumerical

[1] Image and example from [9]

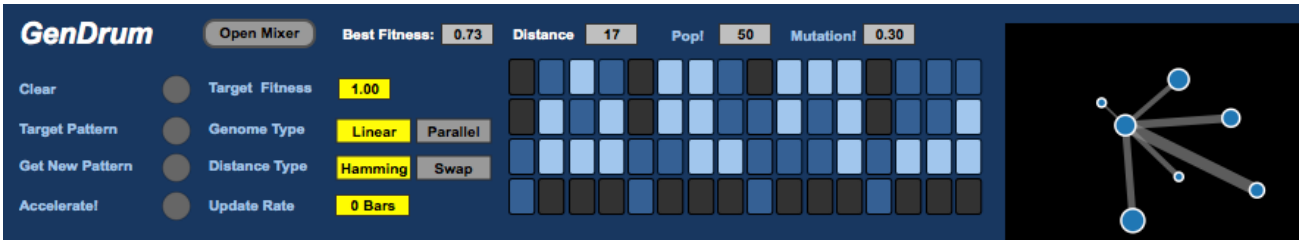


Figure 2. GenDrum Interface

strings. A target string is supplied to the object and fitness is determined by measuring its distance to the generated strings using the Hamming or directed-swap distance, which will be explained in more detail later. Bang messages to the leftmost "hot" inlet causes the genetic algorithm to undergo a generational stage of evolution then output the fittest individual from the pool.

Next we created the *GenDrum* instrument, a *Max for Live* device that uses the *SimpleGA* genetic algorithm to create polyphonic drum patterns of synthesised kick, snare, hi-hat and crash/clap type sounds (Fig. 2). These four sounds are mapped onto a single 16-step row to create a 4x16 drum pattern matrix as is evident in the figure. The operation of the instrument is intended to be as simple as possible. The user inputs a desired pattern into the pattern matrix and assigns this as the target pattern. New target patterns can then be generated by clicking on "Get New Pattern". This assigns the best candidate in the current population to the grid and reports its fitness in the number box. In the yellow number box a target fitness can be issued to the genetic algorithm and clicking "Accelerate!" will increase the speed of the algorithm until it reaches this target. Patterns that are close to perfect fitness now emerge from the algorithm, repeating and contrasting with the target pattern.

The instrument also attempts to provide some visual feedback on the evolutionary process using a type of force-directed graph. The fittest individual from each population is represented as a node with its fitness determining the size and distance in relation to the target individual.

3.2 Algorithm Details

3.2.1 Linear vs. Parallel Operation

As the previous section explains, the instrument is intended as a fully functional drum machine, and hence is polyphonic with the possibility of multiple sounds concurring in time. As seen in the literature, similarity studies are predominantly focused with examining monophonic patterns such as the single sound clave son. How we deal with the polyphonic implications in our research is outlined here.

Our first naive implementation of the genetic algorithm converts the 4x16 drum pattern matrix into a single "linear" 64 digit binary string. Evidently this is a simplistic "brute force" approach that does not explicitly take into account any musical or perceptual aspect of the application. Specifically it does not embed any knowledge about the constituent sounds and the separation between them in the genome string. Essentially, we're treating it as a single sound 64-step pattern, with crossover and mutation happening at the halfway point between the first 16 bits of the kick and snare pattern and the second 16 bits of the cymbals/crashes.

Another approach is to force some kind of logical separation between the different polyphonic timbres in the pattern. By assigning a separate instance of the genetic algorithm to each timbre an overall mean fitness can be derived from the individual outputs. This we implemented and labelled as "parallel" mode. Since the genome pattern length is now split from one single 64-bit string to four concurrent 8-bit strings, the time it takes for the genetic algorithms to reach the target fitness is considerably reduced. Some tweaking of the parameters is required to reduce this convergence time and maintain a healthy level of diversity. Setting the population size parameter to 30 and increasing the mutation rate parameter to between 20% to 40% has been found to work well in our experience. Next we turn our attention to the distance measures used to derive the fitness function of these genome patterns.

3.2.2 Hamming Distance

In the review of the state of the art, we referenced how Post and Toussaint have surveyed the effectiveness of the edit distance in determining rhythmic similarity between two binary patterns [9]. Recall that the edit distance allows for insertion deletion and substitution of symbols within the string. A simplification can be made if operations are restricted to substitutions only, and string lengths are equal (as is always the case in our representations) in which case it becomes the Hamming distance.

If a and b are two strings, a fitness function can be derived using the Hamming distance by counting the number of positions where the two strings match then dividing by the total string length (64 for the linear string and 8 for each string in the parallel implementation). This can be seen in the formula below.

$$F = \frac{1}{N} \sum_{n=1}^{n} a_n \oplus b_n \qquad (1)$$

3.2.3 Swap Distance

The Hamming distance takes into account the correct positional scores between two patterns, but it does not give any indicator as to how different a pattern is in terms of the horizontal displacement. Intuitively one would think this horizontal displacement would reflect the important phenomenon of rhythmic syncopation. For example there

may exist two different patterns with equal distance but one pattern has more onsets aligned closer to the original pattern. Is there a measure that can give this "closer" pattern a higher score based on its horizontal distance? Indeed, the swap distance may be a suitable measure in this instance.

As alluded to previously, the swap distance assigns a score depending on the amount of swaps needed to convert one binary pattern from one to another. Computationally speaking, Díaz-Báñez et al. point out that actually performing the swaps to derive the score is expensive and redundant [13]. It is better to create a new vector for each of the patterns working with the offset distances of the onsets instead. For example the pattern {0, 0, 0, 1, 0, 1, 0, 0} would reduce to the offset vector {3, 5}. In the case of patterns with the same number of onsets the computation is straightforward: you simply sum the differences at each index of both vectors.

$$D = \sum_{n=1}^{N} |a_n - b_n| \tag{2}$$

The genetic algorithm can generate many possible string configurations and critically, strings with different number of onsets to the target string. This poses a problem in calculating the distance as the previous equation no longer applies: how to map one string optimally to the other? In fact, this issue has occurred in the similarity analysis of standard flamenco rhythms where the number of onsets frequently differ, as Guastavino et al. point out in [14].

To tackle this, Toussaint proposes an extension to the definition of the swap distance known as the directed-swap distance. It stipulates that a) every onset in the shorter string must receive at least one onset from the longer string and b) every onset in the longer string must go to some onset in the shorter string. Extending the genetic algorithm object for Max and Pd, we implemented an algorithm proposed by Colannino [15] for computing the distance in $O(n^2)$.

The algorithm essentially treats the problem as a minimum surjection between two sets. A weighted directed graph is constructed and the optimal distance between the two strings is extracted by gathering the shortest path, which was done using an implementation of Dijkstra's Algorithm in the Boost Graph library [16]

3.3 Evaluation Design

When evaluating this research and its resulting tools, our goals were to investigate:

1. The overall correlation of distance with perceived experience.
2. Comparing the impact of the Hamming distance versus the directed-swap distance.
3. Comparing the impact of the linear versus parallel string representation.
4. The more informal, subjective issue of the musical "interestingness" of the rhythmic patterns created.

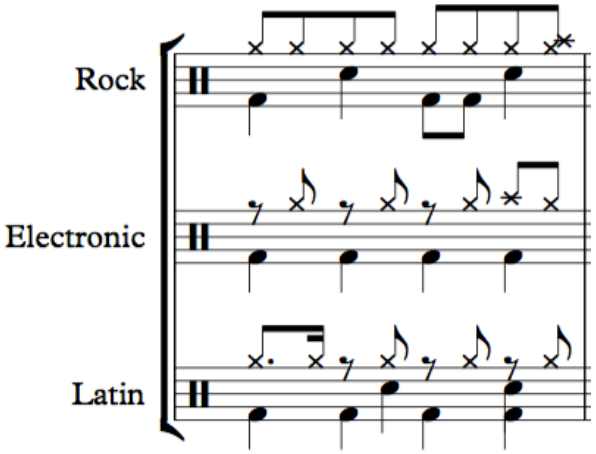


Figure 3. Target Patterns

We decided to conduct a simple listening survey to get listener feedback regarding on these aspects. The survey was web-based and unsupervised; participants were sent a link with instructions on how to complete it.

The listening portion of the survey was divided into two parts. The first part examined similarity ratings by presenting the user with the target pattern and the algorithmically generated patterns. Participants were then asked to rate the perceived similarity on a five-point Likert ranging from "Highly Dissimilar" to "Highly Simmilar".

The second part then examined the "interestingness" of generated patterns. To enable the participant to ascertain this, the patterns were arranged in a soundfile as TPx2, GPx2, TPx2, GPx2 (TP=Target Pattern, GP=Generated Pattern) i.e. a two-bar loop of the target pattern is followed by a two-bar loop of a generated pattern and the whole sequence is repeated. This choice of configuration was quite arbitrary, but it was reasoned that in order to get a sense of the interplay between the target pattern and the generated pattern it was necessary to repeat the sequence at least once.

Once again a five-point Likert scale graded the ratings, this time with labels ranging from "highly disinteresting" to "highly interesting". Regarding the subjective interpretation of "interestingness", we instructed the participants to consider how the target pattern and generated pattern "flows" from one to another, and how the generated pattern "develops" on the target pattern in terms of introducing stimulating variation.

Three target patterns were used for the purposes of the test: a standard straight 8 rock pattern, a four-on-the-floor electronic pattern and a son-based latin pattern (Fig. 3). Table 1 summarises all the variables under consideration for the evaluation. This resulted in a total of 72 WAV files with 3 fitness levels (inversely corresponding to the distance scores).

Question	Measure	String	Pattern	Fitness
Similarity	Hamming	Linear	Rock	Low
Interesting	Swap	Parallel	Latin	Med
			Elec	High

Table 1: Variable Summary

Twenty-two participants took part in the survey, mostly drawn from music students and researchers. All of the participants confirmed that they played an instrument, 7 of whom specified a percussive instrument. Eighteen out of the 22 participants reported the ability to read music. It took approximately 15 minutes to complete.

4. RESULTS

Before carrying out the statistical analysis the responses were summarised by computing the mode of the Likert scores for each stimulus. Two "interestingness" stimuli out of the total 72 (36 for similarity, 36 for interestingess) were removed due to high divergence of opinion (50% or more of the responses deviated by 2 or more Likert scale values from the mode).

4.1 Similarity Ratings

Our first task when looking at the data was to confirm whether inverse pattern distance and the fitness of the genetic algorithm correlates with the perceived similarity as determined by the participants.

Indeed the data seems to confirm this hypothesis. Table 1 presents the Spearman ranked correlation matrix of fitness, distance and the mode scores received for each stimulus. There is a clear, strong negative correlation coefficient (ρ = -0.71, $p < 0.05$) between the distance measure and the perceived similarity to the target. This correlation is not as clear with the fitness function, which can be attributed to the fact that fitness as a function of distance is evaluated differently for the two distance measures.

	Distance	Fitness	Score
Distance	1.0000000	-0.4145087	-0.7134277
Fitness	-0.4145087	1.0000000	0.4353168
Score	-0.7134277	0.4353168	1.0000000

Table 2: Overall Similarity Correlation Matrix

Fig. 4 shows the separated distance and fitness correlations against the mode scores for the Hamming and directed-swap distance measures. The fitness correlation values are 0.784 and 0.625 respectively and the distance correlation values are -0.784 and -0.716 respectively ($p < 0.05$). It can be seen that the Hamming distance has slightly better correlation.

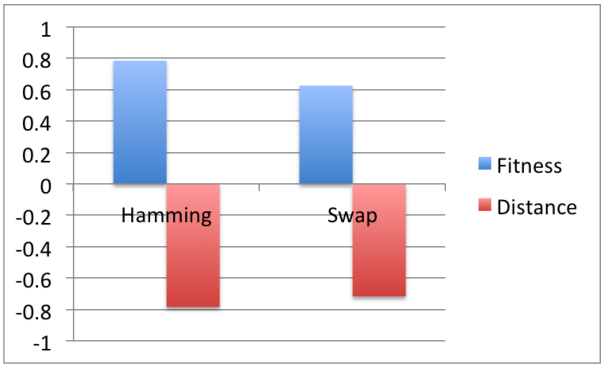


Figure 4. Distance Measure Comparison

Fig. 5 shows the separated distance and fitness correlations against the scores when we discriminate between linear and parallel pattern strings. The fitness correlation values are 0.525 and 0.480 respectively and the distance correlation values are -0.852 and -0.576 respectively ($p < 0.05$). The linear representation scheme appears to correlate better with human judgement.

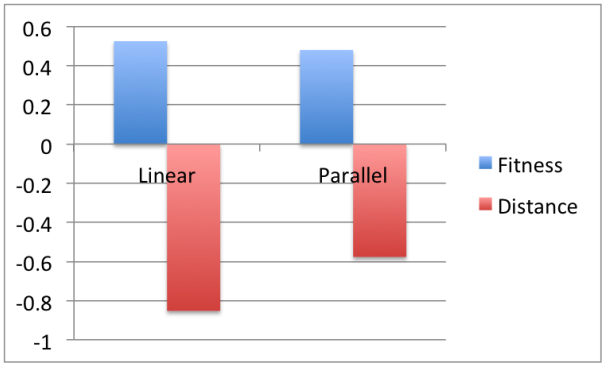


Figure 5. Representation Comparison

We can draw some tentative conclusions based on this data. Firstly, the strong correlation between the overall distance and similarity ratings suggest that even for polyphonic string representations of drum patterns, distances such as the Hamming and directed-swap are useful measures of perceived similarity. Secondly, splitting and recombining the bit strings by timbre, as carried out in the parallel operation, does not seem to offer improvement over the simplistic, long bit string representation. Finally, the more complex directed-swap algorithm, with its ability to capture the "horizontal" displacement between two drum patterns, does not appear to reflect an increase in similarity scores in our survey, confirming Toussaint's finding but also extending it to the case of polyphonic patterns.

4.2 Subjective "Interestingness" Evaluation

Table 4 presents the correlation matrix corresponding to the results of the users' impression regarding the "interestingness" of the patterns when heard in a sequence with the target. Curiously the distance and fitness correlation coefficients are both positive, despite the fact that distance is inversely proportional to the fitness of the genetic algo-

rithm but the p-values are so high (0.211 and 0.1887 respectively) this data is not reliable. It is impossible to draw some meaningful or significant conclusions based on the disparity and inconsistency across subjects.

	Distance	**Fitness**	**Score**
Distance	1.0000000	-0.5419222	0.2201162
Fitness	-0.5419222	1.0000000	0.2310225
Score	0.2201162	0.2310225	1.0000000

Table 3: "Interestingness" Correlation Matrix

Disregarding the difficulty in appraising musically subjective output, the reason for this problematic data is largely attributable to the way in which we considered the notion of "interestingness" and how the question was formulated. Asking the participants to rate two essentially diametrically opposing qualities - i.e. variation (related to dissimilarity) and repetition (related to similarity) was a flawed approach that caused confusion. This became immediately apparent from some of the user feedback at the end of each survey session. For example:

> "I noticed that I somehow prefer rhythms that are a natural evolution of the previous pattern, instead of being totally different. But if the similarity with the previous pattern is too high, the result is still uninteresting to me, because the resulting pattern is too predictable and loses every appeal."

To complete the survey then, users often reverted to their own rules to determine their ratings, as evident from these comments:

> " 'Interestingness' was hard for me to evaluate. In the end I rated with a 'good' interesting 'bad' interesting system: if it's weird but i like it, it's in the first case, if not, in the second case."

> ".. There are some times on experiment 2 that the new rhythm might not be interesting for a complete section but that might be useful as a bridge or as a temporal loop marking the end of a section."

Another participant made the point that the pattern sequence may have forced some "expectation" regarding the concept of "interestingness":

> "... I have also the feeling that having the pattern repeated (ie AB twice), and therefore, having to come back to A again after having been in B, conditions very much the results."

Despite the apparent issues with our method of evaluation, the data and informal feedback does suggest that the genetic algorithm creates "interesting" musical output. Nineteen stimuli out of the 34 analysed registered a mode score value higher than 4 as seen by the 56% green positive region in Fig. 6 (there were two responses for 'Strongly Disagree', but these were the two that were disregarded due to high divergence of opinion). The task ahead is to review the evaluation strategy in order to quantify and explain this aspect in a more coherent and predictable manner.

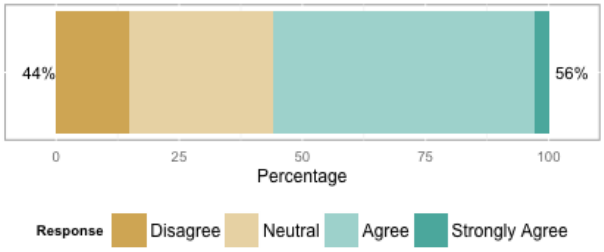


Figure 6. Distribution of Responses for "Interestingness"

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented a way of generating drum patterns automatically using genetic algorithms. Rather than rely on the commonly used interactive fitness function, our method was shown to use the notion of a "target pattern", with fitness derived from the distance of the generated patterns to the target.

Following a review of various approaches to establishing the distance between rhythms as present in the literature, we demonstrated the implementation and incorporation of two such measures - the Hamming distance and the directed-swap distance - into a genetic algorithm instrument for polyphonic drum pattern creation. We believe this paper contributes the first integration of perceptual research in rhythmic pattern generation with genetic algorithms.

To evaluate the research carried out, we conducted a listening survey to determine participants reaction to the generated patterns in terms of the similarity and "interestingess" related to the target pattern. It was shown that the distance and thus fitness correlates strongly with user perception in terms of similarity. Crucially we showed that the Hamming distance alone is a worthwhile quantifier of rhythmic similarity even in the case of polyphonic patterns. Our approach to gauging users' response to the concept of "interestingness" however, needs review and presents a complex challenge for future work.

Links

Code is available to download at:-
`http://www.github.com/carthach/GenDrum`

Acknowledgments

This research has been partially supported by the EU-funded GiantSteps project (FP7-ICT-2013-10 Grant agreement nr 610591). [2]

[2] http://www.giantsteps-project.eu/

6. REFERENCES

[1] J. A. Biles, "GenJam : A Genetic Algorithm for Generating Jazz Solos," in *International Computer Music Conference*, 1994, pp. 131 – 137.

[2] E. R. Miranda and A. J. Biles, *Evolutionary Computer Music*. Springer-Verlag New York, Inc., 2007.

[3] J. Biles, "Autonomous GenJam: eliminating the fitness bottleneck by eliminating fitness," *Proceedings of the 2001 Genetic and Evolutionary Computation Conference*, 2001. [Online]. Available: http://www.ist.rit.edu/~jab/GECCO01/

[4] A. Eigenfeldt, "Kinetic Engine: Toward an Intelligent Improvising Instrument," *Proceedings of the Sound and Music Computing Conference*, pp. 97–100, 2006.

[5] C. O. Nuanáin and L. O. Sullivan, "Real-time Algorithmic Composition with a Tabletop Musical Interface - A First Prototype and Performance," in *AM '14 Proceedings of the 9th Audio Mostly: A Conference on Interaction With Sound*, 2014.

[6] G. Bernardes, "Style Emulation of Drum Patterns by Means of Evolutionary Methods and Statistical Analysis," *Proceedings of the Sound and Music Computing Conference*, pp. 1–4, 2010. [Online]. Available: http://smcnetwork.org/files/proceedings/2010/26.pdf

[7] D. Horowitz, "Generating Rhythms with Genetic Algorithms," in *The Twelfth National Conference on Artificial Intelligence*, 2004.

[8] M. a. Kaliakatsos-Papakostas, A. Floros, and M. N. Vrahatis, "evoDrummer: Deriving rhythmic patterns through interactive genetic algorithms," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7834 LNCS, 2013, pp. 25–36.

[9] O. Post and G. Toussaint, "The Edit Distance as a Measure of Perceived Rhythmic Similarity," *Empirical Musicology Review*, vol. 6, no. 3, pp. 164–179, 2011. [Online]. Available: https://kb.osu.edu/dspace/handle/1811/52811

[10] Y. Grandvalet and D. Eck, "A Generative Model for Rhythms," *Neural Information Processing Systems, Workshop on Brain, Music and Cognition*, pp. 1–8, 2007.

[11] M. Puckette, "Pure Data : another integrated computer music environment," *Proceedings, Second Intercollege Computer Music Concerts*, pp. 37–41, 1997.

[12] T. Grill, "C++ layer for Pure Data & Max/MSP externals," in *2nd International Linux Audio Conference*, 2004.

[13] J. Díaz-Báñez and G. Farigu, "El compás flamenco: a phylogenetic analysis," *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, 2004. [Online]. Available: http://archive.bridgesmathart.org/2004/bridges2004-61.html

[14] C. Guastavino, G. Toussaint, F. Gómez, F. Marandola, and R. Absar, "Rhythmic similarity in Flamenco music: Comparing psychological and mathematical measures," *Proceedings of the fourth Conference on Interdisciplinary Musicology*, p. 76, 2008. [Online]. Available: http://mil.mcgill.ca/docs/GuastavinoToussaintCIM2008.pdfhttp://cim08.web.auth.gr/cim08_abstracts/CIM08AbstractsProceedings.pdf#page=76

[15] J. Colannino and G. Toussaint, "An algorithm for computing the restriction scaffold assignment problem in computational biology," *Information Processing Letters*, vol. 95, pp. 466–471, 2005.

[16] J. G. Siek, L.-Q. Lee, and A. Lumsdaine, *The Boost Graph Library*, 2002. [Online]. Available: http://www.informit.com/store/product.aspx?isbn=0201729148

Harmony of the Spheres: A Physics-Based Android Synthesizer and Controller with Gestural Objects and Physical Transformations

Florian Thalmann
Centre for Digital Music
Queen Mary University of London
f.thalmann@qmul.ac.uk

ABSTRACT

This paper introduces the concepts and principles behind *Harmony of the Spheres*, an Android app that investigates the sonic potential of objects in n-dimensional physical spaces with transforming properties. Using gestural multi-touch and accelerometer control, users can create musical objects in these spaces and interact with them, while they move and react to the physical conditions of the spaces. The properties of these objects can be arbitrarily mapped to sound parameters, either of an internal synthesizer or external systems, and they can be visualized in flexible ways. On a larger scale, users can make soundscapes by defining sequences of physical space conditions, each of which has an effect on the motion and properties of the physical objects.

1. INTRODUCTION

Spatial representations have gained increasing importance in interaction schemes with applications, especially since the rise of smart phones and tablets. In comparison with personal computers, these devices allow users to interact more directly and more physically with the objects represented on the screen, by touching and manipulating them with multiple fingers, moving and shaking the device, or pointing it in a direction and walking around. [1] An increasing number of musical and audio apps thus replace the traditional skeuomorphic systems containing knobs and sliders with more adventurous spatial representations. Most of these spaces directly contain what can be called musical objects, each of them producing sounds when they move around or when they interact with other objects. A few of these apps are based on physical models determining the objects' movements and interactions, which is probably directly inspired by mobile games using physical engines. However, in most of these physics apps the space itself does not seem to have any musical significance and sound seems to be almost uniquely generated when objects collide, which is of course directly analogous to the way we experience physical reality. [1]

With the app presented in this paper, *Harmony of the Spheres (HotS)*, [2] I investigate the potential of other kinds of mappings between the properties of objects in a simulated physical space and conceptual musical dimensions. I also investigate ways in which users can create and interact with objects and physical conditions of the space, using the user interface possibilities of current mobile devices. The concepts underlying the app are inspired by recent developments in mathematical music theory, specifically transformational theory. As the core of the app I developed a generalized physics model for *n-dimensional spaces*. The dimensions of such a space can be mapped, via custom mapping functions, to any musical dimensions, either of the internal synthesizer or of external systems, via OSC. Similarly, they can be mapped to any of the supported visual dimensions, such as spatial position or color. The space is populated with musical objects, called *spheres* (see Figure 1), which can have two kinds of motion. Their *inherent motion* is a multidimensional oscillating motion that is repeated infinitely, independently of physical conditions. Their *physical motion* is the motion caused by the current physical conditions of the space, which can change over time.

Users can interact with the app and create soundscapes by drawing spheres and their inherent motion in any of the n dimensions [3] directly onto the screen. Simultaneously, they can define sequences of temporary *physical conditions*, including various types of gravity, spring collisions, friction, and viscosity, each of which lasts for a specified time before it is superseded by the next conditions. In the following sections, after a brief overview of conceptually related work, I will discuss the app's fundamental concepts on a technical level and describe the way the graphical user interface works.

Copyright: ©2015 Florian Thalmann 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] In his study of all musical iOS apps existing in early 2014, Kell and Wanderley [2] identify 23 applications of a type they call *ball sims*, which all generate sound upon collisions. Examples are Balls, Boinkss, Caelestis, Catalyst, Gravitone, Gravity Beats, or Metalin. However, some apps in other categories are based on the same model, e.g. Bucephalus, Anckorage Spring, and Digital Collisions in the category *synth*, or Amos in *midi/osc*. The category *novel* possibly includes several other ones, e.g. Soundrop. Conditions are the same on the other stores, where we find apps for other platforms such as Bounce 2, or Musyc (Android).

[2] The name is inspired by the ancient Pythagorean notion of the celestial objects creating an inaudible form of music, based on the harmonic ratios of their movements. The harmony of the spheres, also called *musica mundana* or *musica universalis*, formed one of the three kinds of music, along with *musica humana* and *musica instrumentalis*.

[3] The dimensionality of the space can be varied at runtime, and visible dimensions can be switched by changing the visual mapping, in order to draw in all dimensions.



Figure 1. The main screen of *Harmony of the Spheres* with some spheres represented in five visual dimensions.

2. BACKGROUND

Recent developments in mathematical music theory emphasize the importance of musical spaces for the understanding of music. Transformational theory, for instance, models music as recursive sets of objects in logically structured spaces defined by group theory [3] or topos theory [4] and describes how different sets of objects relate to each other by finding intuitive transformations between them. The spacial representation that many musicians are still most familiar with, the score, is not sufficient to express the abstract relationships between the musical entities themselves. This is in line with recent findings in the theory of semantics, based on cognitive science. Gärdenfors, for instance, argues that conceptual spaces are fundamental learning, reasoning, and understanding and that our minds organize information in geometric and topological ways [5]. [4]

Since the early developments of computer music, applications using spatial representations have played an important role. The UPIC system [6], for instance, mapped its two main visual dimensions to the musical parameters of pitch and time, in analogy to the score. Many other apps have since built on such spatial musical representations, allowing for more freedom in mapping visual to auditive parameters and allowing various synthesis techniques, e.g. IanniX [7] or Borderlands [8]. In an earlier research project I dealt with visual and interactive representations of musical spaces and transformations where visual dimensions could be arbitrarily mapped to the dimensions of the musical objects at runtime [9, 10].

On the other hand, several projects investigated the musical potential of mapping physical models to music beyond mere collision triggering as in most mobile apps to date, as noted in the introduction. Inspired by earlier instances of sonification of scientific models, such as in Xenakis' stochastic systems [11], or by the elaboration of perceptual theories such as Gabors acoustic quanta leading to granular synthesis [12], Sturm investigated the potential of a sonification of particle systems [13]. He placed the listener in a physical space surrounded by particles, mapped particle energy to frequency, proximity to the listener to amplitude, and spatial position directly to stereo or quad spatialization position. Others have since created more interactive applications based on mappings of physical models to music, such as RedUniverse [5] or Perkins' recent work [14]. All these examples are based on either two- or three-dimensional spaces, but some of them allow for dynamic mappings between object properties such as position, velocity, or direction to sonic dimensions. Perkins' system, for instance, lets users draw an arbitrary mapping function for each mapped dimension.

[4] In later chapters of his new book, Gärdenfors stresses the importance of action and force in relation to conceptual spaces, which is directly related to the ideas behind this project.

[5] http://www.fredrikolofsson.com/f0blog/?q=node/149

3. THE UNIVERSE AND ITS VISUAL AND AUDITIVE MAPPINGS

In this early version of *Harmony of the Spheres* the spheres exist in a space or universe U, which can be one of three simple types of n-dimensional spaces. These options include $\mathbb{R}^n$ and $\mathbb{R}^n_m$, which are simply the n-tuples of real numbers and the n-tuples of real numbers modulo m, respectively, as well as the bounded n-dimensional unit interval $[0, 1]^n$. At runtime, the users can redefine the number of dimensions if needed, without the possible consequence of losing information when shrinking the space. [6]

The space in itself has an abstract existence, independent of any musical or visual properties. The users can give meaning to any of the space's dimensions by mapping it to any number of audio parameters and visual parameters using a simple dialog shown at runtime. The *audio mappings* can either reach parameters of the internal synthesizer (written in C++ and included via the Android NDK), including frequency, amplitude, phase, panning, timbre, etc, all referring to oscillators or wavetable generators, or they can be used to control parameters of external sound generators, e.g. in a Max/MSP patch, via OSC. *Visual mappings* can again map any dimension of the space to any number of the available visual parameters, currently including x- and y-positions, size, color, and opacity. These mappings determine the way all spheres in U become visible and audible, each of them represented by an independent visual and an audible object. For instance, when mapped to the internal synthesizer, each sphere corresponds to one oscillator with its parameters being set to the values corresponding to the mapped space positions.

Formally, the audio mappings form a functional graph $AC \in N \times A$ where $N = 1, \ldots, n$ is the set of dimension indices of U and A the set of audio parameters or external control parameters. The users can currently control these mappings using a set of drop down menus, one for each audio parameter, where the index of the dimension can be selected. Any such changes in mapping can be chosen to happen smoothly, using internal ramps, which is realized using interpolations between the value of the previously mapped dimensions and the newly mapped ones, a smooth parameter exchange so to speak. However, they can also be switched abruptly, if desired.

In a similar way, the visual mappings match the visual parameters V (currently including x- and y-positions, size, color, and opacity) with the elements of N, again as a functional graph $VC \in N \times V$. A simpler version of this matching principle, which directly linked visual and audio parameters, was first implemented in [9, 15]. Again, the users can control this using drop down menus, as with the audio mappings. Figure 2 shows how the mapping screen looks like in *HotS* for a five-dimensional space.

[6] The app remembers the motions' and conditions' higher-dimensional properties in case the users decide to increase the dimensionality of the space again.

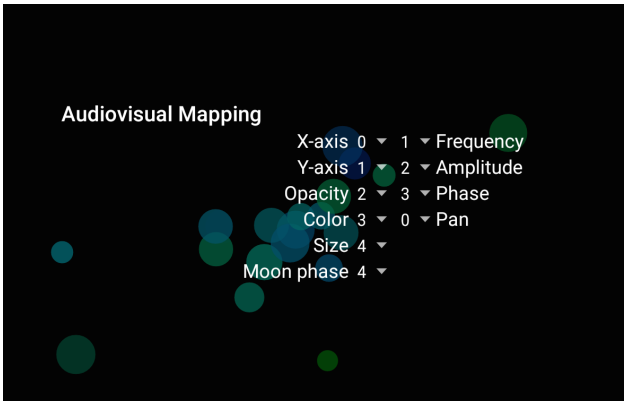


Figure 2. The mapping dialog of *HotS* showing the current visual and audio configuration.

4. THE SPHERES AND THEIR INHERENT MOTION

The focus of *HotS* are the musical objects that populate the spaces just described, the *spheres*. Apart from being moved by the physical conditions of the space, described in the next section, the spheres can have their inherent oscillatory motion as mentioned earlier. In mathematical and computational music theory, musical objects are often defined as simple points or vectors in musical spaces, with either absolute or relative positions. For example, Lewin's transformational theory deals with sets of objects in multi-dimensional spaces such as his Klang space, where all possible triads are represented as points in a two-dimensional space of $pitch \times quality$ [3]. Another example of such point objects are the Module denotators used in Rubato Composer [16], which are based on elements of Module spaces, e.g. the Module over the rational numbers $\mathbb{Q}$. In contrast to such models, the spheres in *HotS* do not consist of positions or directions, but rather trajectories within the space, and could thus be most closely compared to objects in topological spaces. These moving objects exploit the full potential of multitouch devices, with which users typically interact via (pseudo-)continuous finger motions.

Formally, I define a sphere S in the n-dimensional space U as a point $u \in U$ along with a set of n sequences of distances s_i, one for each dimension, i.e.

$$S = (u, \{s_1, \ldots, s_n\})$$

with

$$s_i = (\delta_1, \ldots, \delta_{p_i}).$$

Each of these sequences s_i describes a motion in the corresponding dimension of U, e.g. the distances of s_3 describe a motion in the third dimension, and the point u describes the sphere's reference position. These motions do not all have to be of the same length, which means that each sequence s_i has its own length $p_i \geq 0$. The simplest configuration of a sphere is one with no motion at all $S = (u, \{(), \ldots, ()\})$ with n empty sequences, which is an immobile object in U. For any other possible object, the sequences s_i define its independent motion in the respective ith dimension.

Now how are these objects put in motion in practice? Every sphere starts at its reference point u and then iterates through each sequence s_i independently, but at a synchronous pace. At each step, it adds the current δ_{k_i} to its current position in dimension i. A sphere has thus the potential to oscillate in each dimension with a different periodicity, which can lead to interesting sonic behavior.

Users can add spheres to the space by simply dragging their fingers across the screen (multitouch gestures will result in one sphere added per finger). Depending on the currently selected view configuration VC (see previous section), the motion defined by the finger gesture is recorded and added to a new sphere's dimensions currently associated with the x-axis and y-axis parameters. Then, the users can add motion in other dimensions by switching the perspective by reassigning other dimensions of U to the axis parameters, and drawing new motions from the newly selected perspectives. If they choose to leave some of the dimensions of the motion undefined, they are assigned the empty sequence $()$, which results in a stable position in those dimensions.

In order to ensure maximal usability, *HotS* currently uses a particular kind of spheres: the ones where the elements of each of the motions s_i add up to 0, so that the sphere repeatedly ends up at the starting point in each of the dimensions. [7] This is taken care of automatically at the time a sphere is defined by the user. After a defining touch gesture as just described, each s_i is completed with an additional element consisting of the inverse of the sum of all previous elements.

A special case to be mentioned here are the spheres in the bounded space $U = [0, 1]^n$. Not every conceivable motion is possible to be executed in such a space. For instance, consider a sphere

$$S = ((0.9, 0.5), \{(0.1, 0.1, -0.2), (0.3, -0.3)\})$$

in $U = [0, 1]^2$. This sphere fulfills the condition described in the previous paragraph, where the sum of the elements of each dimension is 0, and the motion is thus cyclical. However, the sphere will quickly reach the edge of the space and not be able to continue its gesture appropriately. Specifically, it starts at position $(0.9, 0.5)$, then moves to $(1, 0.8)$, where it cannot continue to the hypothetical next position $(1.1, 0.5)$. We solve this problem by ensuring that any motion past the edge of the space leaves the sphere at the edge. In this case, this leads to position $(1, 0.5)$. Our example sphere will now jump to a new position in the next step, $(0.8, 0.8)$, a position from where its cyclical motion will now be possible $((0.9, 0.5), (1, 0.8), (0.8, 0.5), \ldots)$. This solution is in line with the edge collision solution presented in the next section and corresponds with the idea of spheres being objects of mainly relative existence, with only the initial position being defined absolutely, which is crucial when working with physical models.

[7] Note that this does typically not happen at the same time for each dimension, since the s_i do not have the same length.

5. PHYSICAL TRANSFORMATIONS

In the previous section we saw how each sphere can have its inherent motion, which consists of a repetitive multidimensional oscillation. The main idea behind *HotS*, however, is a more global type of motion determined by changing physical conditions defined for the space. In the context of mathematical music theory, these physical motions can be considered a generalized form of transformations, which act upon the musical objects. Such transformations are typically expressed as functions on a set of objects. In classical transformational theory, for instance, they consist of groups of operations on sets [3], whereas in newer topos-theoretical approaches they consist of morphisms, again for instance on the category of Modules [4].

Here, our transformations mainly consist of gravity fields, which are either defined in relation to infinite planes, gravitational centers, or between the spheres themselves. Specifically, the physical transformations used in *HotS* are generalized n-dimensional adaptations of the two- or three-dimensional spatial physics commonly used in simulations or games. Currently, all types of transformations available are gravitational. However, users can also define general physical properties of space and spheres, such as collision properties and the viscosity of the material filling the space.

5.1 Gravitational Transformations

HotS currently supports three kinds of gravitational conditions, which can be combined in any way. At the time these conditions are defined using Newtonian gravitational acceleration functions $a(t)$ which assume an equal mass for all spheres. In an iterative process, all current accelerations are summed up for each sphere and added to their current velocity, thus

$$v_{t+1} = v_t + \sum_i a_i(t)$$

where i iterates through all currently active acceleration functions.

Each of the functions can be precisely configured using a linear factor σ which determines a general *gravitational strength* and takes the role of the masses in Newton's formula. The latter two types add a degree of *exponentiality* ϵ, which determines how much stronger spheres are affected by gravity the closer they are to the gravitational object. For Newtonian gravity, we choose $\epsilon = 2$. These and all other determining factors introduced later on can be changed continuously in realtime, which allows a high degree of control over the resulting soundscapes. Currently, this is realized with long exponentially mapped sliders, which ensures both a high precision and large range for each parameter.

5.1.1 Directional Gravity

The first type of gravity assumes an infinite plane generating a constant directional gravity for the entire space. Independently of any sphere's current position, we get the

constant acceleration function

$$a_{dir} = g\sigma$$

where g is a vector determining the directional gravity.

5.1.2 Central Gravity

For this type of gravity we assume a gravitational center $c \in U$. The acceleration function for the current position p_S of a sphere S is then

$$a_{cen}(p_S) = (c - p_S)\frac{\sigma}{d(p_S, c)^\epsilon}$$

where $d(S, C)$ is the Euclidean distance between the gravitational center and the sphere in question. The result is an acceleration vector resulting from the are multiplied with the distance vector between

If several simultaneous centers of gravitation are defined, we calculate the sum of all such accelerations for each sphere, as specified in the velocity equation above.

5.1.3 Cohesive Gravity

The third type of gravity comes closest to Newtonian gravity, being defined pairwise between all present spheres. We get the following formula for a sphere S and its current position p_S:

$$a_{coh}(p_S) = \sum_{S_j \neq S} (p_{S_j} - p_S)\frac{\sigma}{d(p_S, p_{S_j})^\epsilon}.$$

5.2 Physical Properties of Space and Spheres

In addition to the gravitational conditions just described, the users can also define other global physical parameters in real time.

5.2.1 Sphere Collisions

One can activate a collision mechanism based on an repulsive force that is activated as soon as two spheres overlap. It consists of an anti-gravity that is stronger, the more the spheres overlap. We can define this repulsive force as follows

$$\rho\frac{d_{min}}{d}$$

where ρ is the repulsion constant and d_{min} the minimum distance between the spheres (typically the sum of their radiuses) and d their actual distance. This results in a smooth spring-like repulsion where the spheres temporarily overlap. Users can adjust ρ in real time.

A second collision mechanism can be activated when working with $U = [0, 1]^n$. It controls how the spheres are repulsed when they reach the edge of the space. This mechanism works either the same as the sphere algorithm just described, or it can be defined to be more immediate, where the velocity is directly inverted in the respective axis where the maximum or minimum was reached. An additional friction constant ϕ can also be chosen, which defines how much of the momentum is lost due to friction.

5.2.2 Viscosity: Controlling Musical Time

Another global parameter that can be defined in real time is the viscosity υ of the space. It determines how fast all the spheres move and can thus be analogized to a global tempo control. υ is simply multiplied with the velocity vector resulting from all the calculations, which for $\upsilon < 1$ leads to a slower pace of the physics, however, not of the spheres' inherent motions.

5.3 Inherent Motion and Physical Transformations

Now what happens with the inherent motion of each sphere while it is also transformed physically in one of the ways specified above? The inherent motions are still iterated through as it is described in Section 4 and at each step of the iteration, the current distances δ_{k_i} are still traveled in each dimension regardless of the velocity resulting from the physics and before the current effect of the physical conditions are calculated. This has the consequence that spheres maintain their inherent motion relatively, but their overall absolute position varies based on the physics.

This enables users to for instance define spheres that are rapidly modulating their phase or frequency in a vibrato-like fashion, but are moving on a wider trajectory and interacting with each other. The spheres' inherent motion can also lead to favorable effects when the energy of that motion affects the way they collide with each other. Oscillating spheres may eject other spheres from their paths much more energetically than ones with no inherent motion.

5.4 Using the Accelerometer

Directional gravity as well as central gravity can also be varied in realtime using the accelerometer of the device. For directional gravity, the more the device is tilted in a direction, the stronger the gravity gets. For central gravity, the user can move the position of the gravitational center by tilting the device.

5.5 Larger Form: Sequences of Physical Conditions

Even just one physical condition can be used in an improvisatory way by varying its parameters in real time. For instance, a central gravity condition can be moved around, made stronger and weaker, etc, which immediately affects the spheres to react. However, *HotS* also allows users to plan the behavior on a larger scale by defining sequences of physical conditions, each of them lasting for a given duration. This sequence can then either be iterated through once or repeatedly, the latter resulting in a *loop* structure. The total duration of this loop can be varied in real time, by changing the individual durations of the contained physical conditions. Of course, users can also add new spheres at any time, even while physical conditions active.

Instead of looping through the sequence, user may also choose to trigger it several times, with varying musical material (a different set of spheres each time). For this, the app allows users to make the spheres return to their initial position while resetting the sequence of physical conditions to its beginning, before triggering it again.

At a later stage of implementation, the sequence may be replaced by a graph with parallel and alternative edges which can then be traveled in real time, as I implemented it in my earlier project *BigBang* [15].

6. CONCLUSION

This paper gave a brief overview of the concepts and functionality of the current state of *Harmony of the Spheres*. Even at this stage, the sonic capabilities of the app are vast, especially when used in combination with external sound generators. It may act as a counterpoint to commonly used skeuomorphic controllers such as *TouchOSC*, where the conceptual parametric dimensions are usually uncorrelated.

Apart from the potential extensions mentioned in the text, there are several other ones that I plan to work on in the future. In addition to the dimensions of the space, users will soon also be able to choose to map some of the physical quantities of the spheres, such as velocity, direction of motion, etc, to visual and auditive dimensions, just as it was done in [13] or [14]. With the current physical model, described in Section 5, useful quantities would for instance be the absolute value of a sphere's current velocity, which possibly most closely corresponds to the ancient notion of the harmony of the spheres. Other values that can be mapped are the current direction vector, or the strength of collisions.

Another extension especially well-suited for multi-touch devices are topological transformations of the space itself, or the definition of transformative force fields more complex that gravitational fields, which could be considered extensions of the common multitouch gestures.

Finally, the support of external control devices, such as MIDI controllers or the Leap Motion could further extend the degree of interactivity experienced by the users.

7. REFERENCES

[1] G. Essl and M. Rohs, "Interactivity for mobile music-making," *Organised Sound*, vol. 14, no. 2, pp. 197–207, 2009.

[2] T. Kell and M. M. Wanderley, "A high-level review of mappings in musical ios applications," in *Proceedings ICMC, SMC*, Athens, Greece, 2014.

[3] D. Lewin, *Generalized Musical Intervals and Transformations*. New York, NY: Oxford University Press, 1987/2007.

[4] G. Mazzola, *The Topos of Music. Geometric Logic of Concept, Theory, and Performance*. Basel: Birkhäuser, 2002.

[5] P. Gärdenfors, *The Geometry of Meaning: Semantics Based on Conceptual Spaces*. MIT Press, 2014.

[6] G. Marino, M.-H. Serra, and J.-M. Raczinski, "The upic system: Origins and innovations," *Perspectives of New Music*, vol. 31, no. 1, pp. 258–69, 1993.

[7] G. Jacquemin, T. Coduys, and M. Ranc, "Iannix 0.8," in *Actes des Journées d'Informatique Musicale (JIM)*, Mons, Belgium, 2012.

[8] C. Carlson and G. Wang, "Borderlands: An audiovisual interface for granular synthesis," in *Proceedings of 12th International Conference on New Interfaces for Musical Expression (NIME)*, Ann Arbor, 2012.

[9] F. Thalmann and G. Mazzola, "The bigbang rubette: Gestural music composition with rubato composer," in *Proceedings of the International Computer Music Conference*. Belfast: International Computer Music Association, 2008.

[10] ——, "Visualization and transformation in general musical and music-theoretical spaces," in *Proceedings of the Music Encoding Conference 2013*. Mainz: MEI, 2013.

[11] I. Xenakis, *Musiques Formelles*. Paris: Editions Richard-Masse, 1963.

[12] C. Roads, "Asynchronous granular synthesis," in *Representation of Music Signals*, G. D. P. et al, Ed. MIT Press, 1991.

[13] B. L. Sturm, "Composing for an ensemble of atoms: the metamorphosis of scientific experiment into music," *Organised Sound*, vol. 6, no. 2, pp. 131–45, 2001.

[14] R. Perkins, "Sonification of a real-time physics simulation within a virtual environment," in *Proceedings of the 18th International Conference on Auditory Display (ICAD)*. Atlanta: Georgia Institute of Technology, 2012.

[15] F. Thalmann and G. Mazzola, "Using the creative process for sound design based on generic sound forms," in *MUME 2013 proceedings*. Boston: AAAI Press, 2013.

[16] G. Milmeister, *The Rubato Composer Music Software: Component-Based Implementation of a Functorial Concept Architecture*. Berlin/Heidelberg: Springer, 2009.

CAPTURING AND RANKING PERSPECTIVES ON THE CONSONANCE AND DISSONANCE OF DYADS

Aidan Breen
National University of Ireland, Galway
a.breen2@nuigalway.ie

Colm O'Riordan
National University of Ireland, Galway
colm.oriordan@nuigalway.ie

ABSTRACT

In many domains we wish to gain further insight into the subjective preferences of an individual. The problem with subjective preferences is that individuals are not necessarily coherent in their responses. Often, a simple linear ranking is either not possible, or may not accurately reflect the true preferences or behaviour of the individual. The phenomenon of consonance is heavily subjective and individuals often report to perceive different levels on consonance, or indeed dissonance.

In this paper we present a thorough analysis of previous studies on the perception of consonance and dissonance of dyads. We outline a system which ranks musical intervals in terms of consonance based on pairwise comparison and we compare results obtained using the proposed system with the results of previous studies. Finally we propose future work to improve the implementation and design of the system.

Our proposed approach is robust enough to handle incoherences in subjects' responses; preventing the formation of circular rankings while maintaining the ability to express these rankings — an important factor for future work. We achieve this by representing the data gathered on a directed graph. Abstract objects are represented as nodes, and a subject's preference across any two objects is represented as a directed edge between the two corresponding nodes. We can then make use of the transitive nature of human preferences to build a ranking — or partial ranking — of objects with a minimum of pairwise comparisons.

1. INTRODUCTION

Subjectivity and the potential incoherency associated with it can be difficult to handle. Areas like psychology, sociology, aesthetics, music and computer science — collaborative filtering and recommender systems in particular — rely on the responses of individuals within a population who may or may not be expressing incoherent behaviour.

Another issue faced when dealing with human feedback is incomplete data. Often, the attention span of a subject is simply not long enough to collect sufficient information with incoherences potentially increasing over time. Some studies disregard responses from subjects where inconsistencies in responses grow too large. In one example case, discussed further in section 2.2, sample sizes were reduced by almost half [1].

Using directed graphs to rank or sort objects is not a new idea. There are numerous topological sorting, or "toposort", implementations using directed graphs. Topological sorting is achieved either by performing a reverse postorder depth first sort, or by pushing the first node found with no incoming edges onto a stack — the output — removing that node from the graph and repeating the process until all nodes have been stacked [2–5]. These approaches, however, rely on the graph to be acyclic, which is not typically guaranteed when dealing with subjective responses.

The field of graph theory is extensive, providing many tools for the representation, transformation and computation of digraphs. Further approaches within this domain have been described to rank players in a tournament [6] or candidates in an election using digraphs (both weighted and unweighted) [7, 8]. These approaches, while capable of handling cycles within the graphs, are restricted to semi-complete digraphs — digraphs where every two vertices are connected by at least one edge.

While these approaches may not be useful to us in most cases, some aspects, such as the Copeland score method [7] — see 2.4 — have been useful as a basis to form our own algorithms.

1.1 Motivations

In this paper we firstly present a review of previous studies in the area of consonance and dissonance of dyads. We then briefly identity an algorithmic approach to handle subjectivity using a digraph which addresses some of the issues faced by previous studies. The results obtained using this approach are then compared to the results of previous studies.

Our approach aims to be efficient in reducing the number of questions asked to each subject during a trial. Not only does this speed up the testing process but it also reduces the potential for incoherences which may increase over time.

Our approach also aims to be expressive enough to represent possibly contradictory answers from subjects. We believe this incoherence may prove to be useful information in terms of accurately describing the preferences of an individual. The expressiveness required to represent this information leads to an extendibility which may allow future applications to increase in efficiency and gain a deeper understanding of the true preferences of an individual.

Copyright: ©2015 Aidan Breen 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.

In order to correctly test our approach we take in in-depth look into one particularly well studied area that aims to apply a formal structure to an inherently subjective phenomenon: consonance, dissonance and roughness. This topic provides a great deal of related work. Data gathered from various studies provide a strong benchmark to which we can compare our own results.

We aim for our approach to be applicable to many domains. Even though the phenomenon of consonance, dissonance and roughness varies across cultures, it is globally understood as a core quality of musical experience. This makes it a topic that we may all have a fair opinion of — subjective as it may be — regardless of expertise.

1.2 Layout of Paper

This paper covers three main areas: the background to our test-bed, our implementation and experiment, and finally, our discussion and conclusions.

The background to our test-bed is described in section 2. In this section we introduce the concepts of consonance, dissonance and roughness followed by an outline of a number of studies carried out on the topic in relation to pairs of musical notes played in harmony (dyads). The studies described have all attempted to develop knowledge in an area that is subjective, culturally influenced and prone to inconsistencies from an individual perspective and across a population. We continue the section with a review of these studies. Section 3 describes in detail the specific challenges we face in designing our approach, based on our review in section 2.3.

Our implementation and experimentation is described in sections 4 and 5.

Finally, our discussion in section 6 provides some context for our results and contains some possible future work and improvements to our current approach, followed by our conclusions in section 7.

2. RELATED WORK

2.1 Consonance, Dissonance and Roughness

Consonance and dissonance, a notion of pleasant or unpleasant harmonies respectively, are fundamental to our understanding of musical aesthetics. The concept is greatly influenced by cultural differences and is largely subjective across individuals. It is also largely influenced by timbre which varies between sound sources and may introduce beatings between close harmonics. This makes it difficult to build a strong representational model of consonance or dissonance that holds across cultures and over time.

Since Pythagoras, consonance has been observed to correlate to the frequency ratios of notes. Galileo was the first to draw the connection between these ratios and the operation of the eardrum. He noted that pairs of musical notes with simple integer frequency ratios tended to sound more pleasant than those with more complex ratios, with the eardrum "kept in perpetual torment" [9].

The idea of "roughness" was first introduced by Helmholts [10] to describe the auditory phenomenon of harsh sounding signals due to amplitude fluctuations or a "beating" effect. It has been hypothesised that the roughness of a signal has a strong equivalence to the consonance or dissonance of that signal, at least within western music, however, the link between this measure and some notion of aesthetics is unclear [11–13].

2.2 Previous Studies

There have been numerous approaches to modelling the consonance or roughness of musical intervals both mathematically and experimentally based on trials involving human subjects.

Helmholtz, wrote at length about the topic of roughness [10]. His observations, though often referenced and strikingly similar to more modern revisions, were not much more than personal observations based on musical theory. However his model of roughness provided a starting point for further research to expand upon.

Malmberg published a study which tested 1045 subjects in three distinct groups between 1913 and 1915 for their preferences of consonance [14]. Subjects were played two pairs of tones – dyads, "or two-clangs" in Malmberg's words – on a piano or violin and asked to select which they preferred in terms of consonance. A strong effort was made to ensure all subjects understood the concept of consonance as Malmberg noted "the fundamental reason for the great divergence in the ranking by experts and the consequent disparagement of the ranking of consonance and dissonance has been due to the failure to take common ground in the definition of these terms".

Malmberg's study was conducted on a scale that has not since been reproduced. However, he did not build a ranking solely based on the responses of his subjects. Malmberg's tests aimed to compare the responses of his 1045 subjects – the "empirical ranking" – to a "norm", a ranking which had been developed based on the responses of "eight observers [who] were carefully selected on the basis of their training and fitness for the work". The eight observers were originally required to produce independent rankings, of which the averages would become the "norm", however Malmberg notes that after an initial conference, the "discussion and mutual criticism was so stimulating and interesting that all the observers agreed to sit again and continue by the same method until all should agree and a unanimous verdict could be handed in as in the case of a jury". The results of this study appear unprincipled and highlight potential difficulties in obtaining agreement in this domain.

Though the results obtained from Malmberg's observers did not directly affect the responses of his empirical study, the order of presentation and the particular dyads compared by subjects were based on his initial flawed study which may have affected his final results.

Plomp and Levelt described their approach to investigating the effect of critical bandwidth — the sensory limitations of hearing — on consonance by having subjects rate pairs of tones (generated by sine-wave oscillators) on a seven point scale, consonant to dissonant [1] with 1 corresponding to most dissonant, 7 corresponding to most consonant. They mention that some subjects had to ask for a

definition of "consonant" which they provided as "beautiful and euphonious" as it had be ascertained that "consonant, beautiful and euphonious are highly correlated for naive subjects." Obviously Plomp and Levelt conducted these experiments with a relationship between consonance and aesthetics in mind, which serves to highlight a point made by Malmberg regarding a failure to adopt a common ground on the definition of consonance and dissonance leading to disagreement in the domain.

Though the experimental set up that Plomp and Levelt used was rather inelegant — equipment had to be readjusted by hand between each test — their tests have a number of aspects in common with future studies. Firstly, as mentioned above the subjects in their tests were not all musically trained. Secondly, the tone pairs, generated for their tests were composed of simple sine waves which reduced the tonal complexity of the resulting sound presented to the subject. Finally, their subjects rated tone pairs on a graduated scale.

Plomp and Levelt generated tone pairs about a set of mean frequencies (125, 250, 500, 1000 and 2000 Hz) and used a separate sample group for each. The sample groups were reduced in size from 19, 22, 18, 11 and 18 to 11, 10, 11, 10 and 8 respectively when they removed subjects who displayed incoherent responses.

Kameoka and Kuriyagawa conducted an independent study on the absolute and relative consonance of dyads [12]. Like Plomp and Levelt, their sample groups included two groups and their sample tones were generated using sine-wave oscillators, similar to [1]. In contrast however, subjects were not asked to rank consonance and dissonance absolutely (as in [1]) but rather in a relative manner. Subjects were presented with two dyads (A and B) and asked to provide an answer on a five point scale, -2, -1, 0, 1, or 2 "according to the subjective distance in consonance between A and B. If B is more consonant than A, a plus sign and if more dissonant, a minus sign was given" [12].

The groups chosen by Kameoka and Kuriyagawa for their experiments are rather puzzling. The first, "audio engineers, who are regarded as ordinary people" and the other, "mixers of ... the Japan Broadcasting Corporation", later referred to as "specialists" [12]. It should be obvious to any reader that a person working in the field of audio engineering is far from an ordinary person in the context of their perception of musical consonance and dissonance. The authors proceed to mention that their "experiments were carried out with audio engineers, and the results in this paper should be interpreted as for ordinary people", a contradiction in itself.

It should be noted that Kameoka and Kuriyagawa used exclusively Japanese speaking subjects. This resulted in the concepts of consonance and dissonance being defined in an entirely different language, adding a further layer of complexity to Malmberg's notion of the definitions of these concepts and how that may affect a subject's response. Additionally, there may also have been a cultural influence which, as stated previously, may have a strong effect on a subject.

Hutchinson and Knopoff later produced a "formalism" for calculating the consonance of a pair of notes based on the work of Helmholtz, Malmberg and Plomp and Levelt. [15]. In their paper they discuss a number of formulas which serve to produce absolute values of consonance for different intervals. The comparison of their values to the results produced by Malmberg seem to show a large degree of correlation, however they fail to provide a direct assessment of their values by means of experiment.

2.3 Issues with Previous Studies

It has been noted that there are issues with the reproducibility of both studies carried out by Kameoka and Kuriyakawa, and Hutchinson and Knopoff [16]. This would suggest that the methods they used were simply not informative enough to accurately describe the behaviour of their subjects.

Plomp and Levelt had their subjects rank note pairs in an absolute 7 point graduated scale [1]. Malmberg based his ranking on the unanimous decisions of hand picked experts and then tested on large groups of subjects all at once, with a common test pattern [14]. The work of Plomp and Levelt, and Malmberg has been the basis of other papers attempting to produce a formulation of consonance, such as Hutchinson and Vassilakis [15] [11]. None of these studies have taken into account the contextual effects of juxtaposing particular note pairs or ranking intervals absolutely and the effects these approaches may have on results.

For example, the first note pair a subject is presented with is the only pair that does not follow a previous pair and thus has no context. The subject knows the pair must fall somewhere on the scale of dissonance provided but without any reference point, their first response is arbitrary. Alternatively, if a subject is given a number of note pairs in succession which would otherwise have been ranked on one extreme of the scale, they may be more likely to spread their ranking across the scale. Furthermore, after these similarly ranked note pairs, if a new note pair is given that, in another context would be close to the center of the scale, the subject might subconsciously rank this pair further to the other extreme of the scale. It could be argued that successive dyads may be presented to subjects with a large enough interval to ensure short term memory does not effect the response. Indeed Plomp and Levelt adopted this strategy with a 4 second interval [1], however other studies do not follow this paradigm, for example Kameoka and Kuriyakawa who mention an interval of only 0.5 seconds [12].

The problem here is context. Human beings are heavily influenced by context which can lead to incoherent responses. Studies conducted like those described in this paper depend on each test being free from contextual influences from previous tests, which is simply not possible given the experiment design in the studies mentioned earlier.

There are three main solutions to this contextual conundrum. Firstly, to present pairwise comparisons rather than an absolute ranking which reduces the effect of a user subconsciously dispersing their rankings. This approach was adopted by Kameoka and Kuriyakawa with success [12]. Secondly, to test subjects individually and present comparisons in a random order, which will reduce the effect

that the first note pair presented without context has on the result across a population. Finally, some studies like [1] use test intervals in a training session to establish a relative scale for a subject. However there are issues with this approach as different studies my train subjects to different degrees, by different methods, or indeed over train subjects and introduce fatigue.

In many earlier studies this solution was not achievable. Malmberg went to great lengths to find reproducible tones, testing pianos, organs, violins and bottles filled with wax among other instruments. If the subjects had been tested individually, it would have been arduous on the musicians and unreliable in terms of producing identical note pairs due to the fluctuations in timbre and pitch over time due to fatigue and temperature or humidity changes. With modern computer systems, it is possible to produce identical tones over and over again, as well as randomizing test orders fairly and handling the resulting data.

Another problem we have seen is how researchers may simply disregard incoherent, or contradictory responses from subjects [1]. We believe there is information to be gathered from contradictory responses and that rather than being disregarded, contradictions may actually provide a greater insight into the subjective preferences of test subjects, whether it be consonance, dissonance, aesthetics or any other subjective – or noisy – domain. Once again, it may have been the technological constraints of the time that forced researches to reduce the complexity of their analytical processes, but with modern technology it should be achievable to handle these cases.

2.4 Ranking Methods

The Copeland Score, or Copeland Method [7] is a very common and easily understood ranking method. Nodes are ranked by their out-degree, or alternatively their out-degree minus their in-degree. This method, while easily understood, often leads to ties and bears no relation to the overall structure of the graph. In our ranking approach we use this method to find or break ties.

Kameoka and Kuriyakawa asked their subjects to rank dyads on a scale between -2 and 2. These absolute rankings were then normalized across the population using a simple normalisation function described by Guilford [17].

Plomp and Levelt used a similar method of ranking dyads on an absolute scale but used a 7 point scale (1-7). The lower quartile, median, and upper quartile of subject responses were used as consonance values [1].

3. ISSUES FACED

It has been shown that the previous studies have not provided sufficient evidence to describe the preferences or behaviour of their subjects without question or doubt [16]. As we have seen, there are two main issues here: changing context, and subject incoherence. We propose a different method of ranking which allows subjects to make pairwise comparisons which removes contextual issues, and map those comparisons to a directed graph which allows us to make use of otherwise incoherent or contradicting responses.

3.1 Changing Context

We propose tackling the problem of changing context by forcing a relative context upon subjects. Subjects should only compare two objects — as in Kameoka and Kuriagawa [12] — rather than provide an absolute response (ie, grading objects on a scale) — as in Plomp and Levelt [1]. For example, rather than asking a subject to provide a value for objects A and B in succession, a subject is asked only to say which object, A or B, is more expressive of the value being tested. A subject might say that object A is more dissonant than object B. In this case, we now know that for whatever dissonance value we assign to object A, object B will always be less than or equal to that value. We refer to this as a *dominance* where A is dominant over B, or A >B.

The objects being tested, in this case A and B, can be represented as two nodes on a directed graph. A dominance is represented on the graph as an edge between two nodes. In the example above, we would produce a graph with two nodes, A and B, with an edge between them directed from A to B representing the dominance of A over B.

As currently described, the relationship between two objects A and B may have three possible values. A is dominant over B, B is dominant over A, or both A and B are dominant over each other. In this last case, we have a cycle.

This representation can be further expanded. If we extend the responses available to subjects to enable them to say how strong a dominance is (with a sliding scale perhaps), we can model a more expressive response. A more expressive response can be represented on a graph as a weight or distance associated with a directed edge.

3.2 Subject Incoherence

Incoherence refers to a subject contradicting their previous responses. As a subject becomes familiar with the boundaries of the experiment they may express more coherent responses. In our approach, we assume incoherence is affected mainly by two factors: familiarity with the experiment and fatigue over time. We have mentioned that in some studies the responses from incoherent subjects were disregarded. We believe that this is a failure of the data collection method. We now outline our approach to minimizing the introduction of incoherence, and handling incoherence that may be introduced.

To minimize the influences that increase incoherence, we adopt the following approaches. Firstly, regarding familiarity with the experiment, a subject's first responses may not be as accurate as their later responses. This effect is partly addressed by pairwise comparison, but further reduced across a population by posing initial questions in a random order. Secondly, a subject will become inconsistent over time as fatigue and boredom begin to affect their attention. To this end, we allow subjects to stop at any point if they feel bored, tired or uncomfortable. In this case we can still provide a partial ranking due to the initial random distribution of dominances throughout the graph (see 4.1).

On a graph, incoherence and contradiction manifests as a cycle. On a graph without weighting, a cycle implies a tie between all of the nodes on the cycle, where each node in the cycle is equal in value. In this way, we can now handle incoherences safely, without losing information about unrelated coherent responses.

It may also be possible to represent contradicting responses as a cycle while producing a fully ranked result by using a weighted graph. This is discussed further in 6.3.

4. APPLICATION OF THE DIGRAPH

It would be beyond the scope of this article to describe the entire implementation of our approach. However, in the following section we will briefly introduce the core concepts of our implementation for the purpose of clarity and to justify the data we have obtained.

Briefly, we generate a digraph for each subject based on their preferences. The objects we wish to rank — in this case musical intervals — are represented by nodes, and questions posed to the subject are represented by edges on the graph. The purpose of ranking a digraph is to produce a list of nodes in a ranked order. Maximally efficient ranking will produce a fully ranked list (ie. no nodes with the same ranking) with the minimum number of edges. Individual rankings may then be simply normalised by calculating the average rank of a node across the population.

By using this structure we should be able to begin with a totally unconnected graph and ask questions to the subject one by one until: **a)** we have a fully ranked list, **b)** the subject becomes fatigued and wishes to stop, or **c)** we have asked some arbitrary number of questions to prevent subjects from becoming fatigued.

To implement this approach we must provide mechanisms to answer the following questions:

- How do we decide which edges to add and in what order?
- How do we define the ranking of objects, given some set of edges?

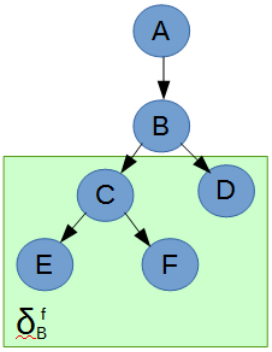
4.1 Adding Edges

To add edges to the graph we calculate a rank for each node on the graph (see 4.2), and proceed by breaking the ties between equally ranked nodes by adding an additional edge between them — *ie.* by asking a question of the subject to rank two nodes. Nodes that are explicitly tied — part of a cycle — are not considered in this process.

In the case where we have no more ties to break, we can either finish questioning the subject or over-sample (see section 6.2) by using a different method of calculating ties, such as the Copeland score method [7].

4.2 Ranking Nodes

The ranking of nodes provides a mechanism to both calculate ties in order to add more edges, and output a ranked list for a subject when testing is complete. The ranking algorithm used to calculate the final rank of nodes in a digraph must obviously remain constant across a population.



Node	Ranking
A	6
B	5
C	3
D	1
E	1
F	1

Figure 1. A sample graph and rankings. δ_B^f shows the forward nodes reachable from B.

Due to the possibility of some digraphs across a population being sparse, the ranking algorithm must accommodate for this.

The particular ranking algorithm used is not important. Although there are a number of algorithms designed to sort directed acyclic graphs [3], we are not aware of any formally defined algorithms that suit our specific needs by allowing cycles and ties. We therefore introduce our own ranking algorithm.

Though this un-weighted graph provides a limited expressiveness, Figure 3 shows that the use of this graph can provide a strong representation of the ranking of objects, especially when normalized across a population of test subjects. In section 6.3 we discuss the application of a weighted graph which may be more expressive.

Given the set of forward nodes reachable from a node n as δ_n^f, the rank $R(n) = |\delta_n^f| + 1$, as demonstrated in Figure 1. In this way we ensure that $R(n_1) \geq R(n_2)$ where $n_1 \succ n_2$. Calculating the rank in this way also means that nodes in a cycle will have an equal ranking, as they will all be reachable from each other.

The above ranking method can be calculated trivially for any node using a depth first search.

5. EXPERIMENT

We now describe an experiment conducted to show the effective implementation of the algorithm described in section 4. A small sample group of 25 subjects were presented with the experiment using recordings of piano notes to dynamically produce dyads. Dyads were produced using notes across two octaves, from A4 (440 Hz) to A6 (1760 Hz). Only intervals from 1 to 11 semitones were tested — unison and octave dyads were excluded to avoid confusion. The single sample group included both musicians and non-musicians, males and females, aged 21 to 57. All subjects were brought up within the western music culture.



Figure 2. A screen shot of the testing software. Users were asked to click on the red sound icons and select the appropriate pair using the buttons below.

5.1 Method

Each subject was tested separately. Subjects were first presented with a guideline document which described the testing software, the format of the experiment and the definition of consonance as: "Agreement or compatibility between opinions or actions", and further: "When two tones tend to blend or fuse and produce a relatively smooth and pure resulting sound, they are said to be consonant". This second definition is based on the description by Malmberg [14].

Each subject was then presented with the software as shown in Figure 2. Dyads — described as "pairs" to subjects — were played by clicking sound icons at the top of the screen, and a preference was defined by clicking the appropriate buttons below. After each preference selection, a screen with the words "Click to Continue" was displayed in order to prevent accidental preference selection.

Each preference was presented in order to break ties as describe in 4.1. Testing finished when all ties had been broken and a full ranking was achieved.

5.2 Results

Subjects provided an average of 24.8 preferences before a full ranking could be calculated and completed the test in an average of approximately 6 minutes. This is in contrast to the minimum of 55 potential preferences required with a semi-complete graph with 11 nodes (V), given as: $\frac{V^2-V}{2}$ The sample mean ranking of each interval is shown in Figure 3. This ranking is compared to previous data — we use Hutchinson and Knopoff [15], and Malmberg as examples [14] — as shown in Figure 4 and Table 1, also see [11, 12] for other comparable results. It can be seen that our ranking follows a similar pattern to previous data where the intervals of 1 and 11 semitones tend to be less consonant (more dissonant, rough) and the interval of 6 semitones is conspicuously lower than its highly ranked neighbours of 5 and 7 semitones.

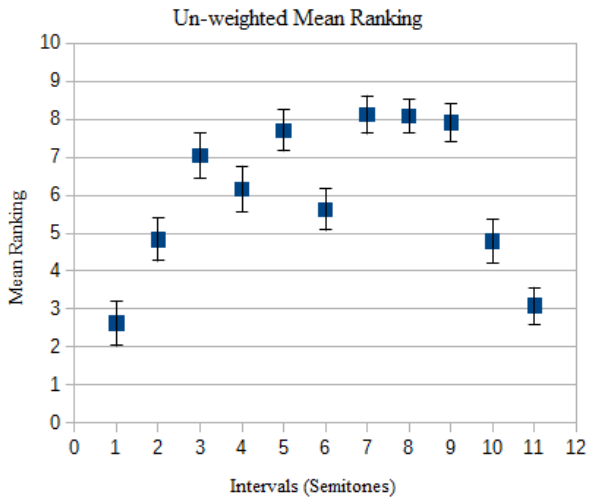


Figure 3. The mean ranking of intervals across a population of size 25 using an un-weighted digraph. Error bars show standard error of the mean.

	Correlation Coefficient	P-value
Hutchinson & Knopoff	0.788	0.004
Malmberg	0.852	0.001

Table 1. Correlation of digraph rankings with previous studies.

6. DISCUSSION

Though our experiment was carried out on a relatively small sample size, our results show a strong resemblance and correlation to previous studies. Even with a simple unweighted graph and no questions asked beyond those necessary to break all ties.

6.1 Experimental Improvements

Along with a larger sample size, a number of other improvements could be made to the experiment. Primarily, we conducted the experiment using piano notes which contain many harmonics. In-harmonic frequencies in the piano notes may have had an affect on the results produced. Piano notes were selected rather than pure sinusoidal tones in order to compare our results to the largest previous study, carried out by Malmberg [14], which used piano notes. A future study could be conducted to compare our results to those obtained from previous studies using pure tones.

The experiment outlined in this paper can be seen as the simplest implementation of the ranking digraph. We intend to further explore the possibility of building a more expressive and efficient ranking process for subjective preferences. The following sections discuss these possibilities.

6.2 Over-sampling

Our current approach of calculating ties and asking questions to break those ties does not allow subjects to specify

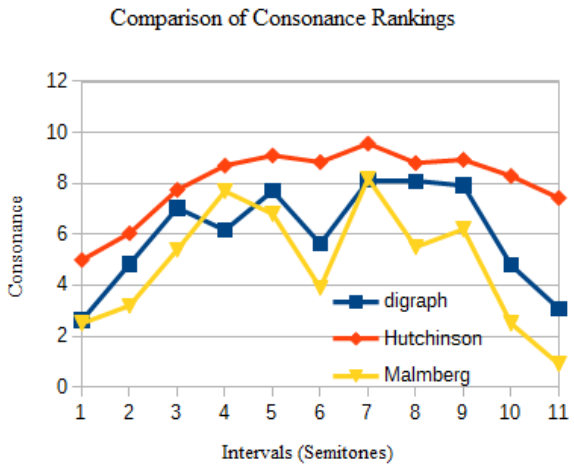


Figure 4. Comparison of results to previously published rankings [14, 15].

cycles. Subjects may express cycles if given the opportunity and we feel this information may be an important factor in the ability of any graph to accurately reflect the preferences of an individual.

To improve our current approach and allow cycles to form we propose **over-sampling**: continuing to ask preferences even after all ties have been broken. As noted in section 4.1, a different method of ranking can be used to implement this and calculate new ties.

6.3 Graph Implementation Improvements: Weighted Graph

By using a weighted graph it may be possible to both allow cycles to form, and fully rank objects, even when cycles are present in the graph. To achieve this we propose the following potential solutions.

6.3.1 Handling Cycles

In an un-weighted graph, cycles of more than 2 nodes may form, leading to many nodes in an explicit tie. In the case of a weighted graph, where a weight W $\in N$ and 0<W<1, we make the assumption that in a tie, the edge with the lowest W can be safely excluded in order to break the tie and fully rank all nodes on a graph. In a cycle where there is no singular lowest W, we cannot break the tie and proceed with all nodes in the cycle tied, and of equal value.

This solution relies on the proper ranking of nodes within a sparse weighted digraph.

6.3.2 Ranking a Weighted Graph

The weighted graph may provide a deeper expressiveness at the expense of the simplicity of implementation. A number of ranking methods exist for weighted directed acyclic graphs [8, 18] however the vast majority of research in the area assumes semi-complete or complete graphs and will not handle cycles. We therefore outline our own algorithm as one possible approach to this challenge.

Given a digraph $D = (N, E)$, a subset N_x^f (forward neighbours of x, nodes that x points to), and a subset N_x^b (backward neighbours of x, nodes that point to x) we make use of a two pass approach as described below. A two pass approach is necessary to handle ranking imbalances which occur when nodes form disconnected branches, of differing weights, sharing a common root.

First pass

Populate the working set W with all end nodes (nodes with zero outgoing edges). For each node $n \in W$, move it to the visited set V, give it a ranking $R(n) = 1$ and add all nodes pointing to n, N_n^b to W. To iterate through the entire graph, for each node $n \in W$ with forward neighbours (N^f) such that $N_n^f \subseteq V$, calculate the rank of n as

$$R(n) = \sum_{i \in N_n^f} E_{i,n} i \qquad (1)$$

where $E_{x,y}$ is the weight of the edge between nodes x and y. Continue until $|W| = 0$.

Second pass

$V = \emptyset$. Populate W with all start nodes (nodes with zero incoming edges). For each node $n \in W$, move it to V, keep its ranking the same as the first pass and add N_n^f to W. Iterate through to the entire graph, for each node $n \in W$ where $N_n^b \subseteq V$, calculate the rank of n as

$$R(n) = \sum_{i \in N_n^b} E_{i,n} i \qquad (2)$$

Continue until $|W| = 0$.

7. CONCLUSIONS

We have presented a detailed review of the literature in the area of the consonance, dissonance and roughness of dyads concerning our test bed. We outline some of the shortcomings of these studies and our findings based on this research provide the basis for the design of our ranking system.

We have identified an algorithmic approach to handling subjective preferences using a graph that can efficiently rank objects, express contradictory responses from subjects and provide a basis for extending this approach.

Our test bed experiment produced a ranking of objects within a subjective domain using a small sample group. We did so with a much smaller set of questions per subject than previous studies [12, 14, 15], in a way that minimized the effect of contextual issues and subject incoherence found in previous studies.

Finally, we have also proposed possible methods of extending our approach using over sampling and ranking subject responses using a more expressive weighted graph.

Acknowledgments

This research was funded by the Hardiman scholarship, NUIG.

8. REFERENCES

[1] R. Plomp and W. J. M. Levelt, "Tonal consonance and critical bandwidth," *The journal of the Acoustical Society of America*, vol. 38, no. 4, pp. 548–560, 1965.

[2] C. Pang, J. Wang, Y. Cheng, H. Zhang, and T. Li, "Topological sorts on DAGs," *Information Processing Letters*, vol. 115, no. 2, pp. 298–301, 2015.

[3] D. Ajwani and T. Friedrich, "Average-case analysis of incremental topological ordering," *Discrete Applied Mathematics*, vol. 158, no. 4, pp. 240–250, Feb. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166218X09003047

[4] B. Haeupler, T. Kavitha, R. Mathew, S. Sen, and R. E. Tarjan, "Incremental cycle detection, topological ordering, and strong component maintenance," *ACM Transactions on Algorithms (TALG)*, vol. 8, no. 1, p. 3, 2012.

[5] I. Katriel and H. L. Bodlaender, "Online Topological Ordering," *ACM Trans. Algorithms*, vol. 2, no. 3, pp. 364–379, Jul. 2006. [Online]. Available: http://doi.acm.org/10.1145/1159892.1159896

[6] A. Rubinstein, "Ranking the participants in a tournament," *SIAM Journal on Applied Mathematics*, vol. 38, no. 1, pp. 108–111, 1980.

[7] I. Charon and O. Hudry, "An updated survey on the linear ordering problem for weighted or unweighted tournaments," *Annals of Operations Research*, vol. 175, no. 1, pp. 107–158, 2010.

[8] J. r. Bang-Jensen and G. Z. Gutin, *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.

[9] Galilée, H. Crew, A. FAvaro, and A. de Salvio, *Dialogues Concerning Two New Sciences*. Dover Publ. Incorporated, 1967.

[10] H. Von Helmholtz, *On the Sensations of Tone as a Physiological Basis for the Theory of Music*. Longmans, Green, 1912.

[11] P. N. Vassilakis, "Auditory roughness as means of musical expression," *Selected Reports in Ethnomusicology*, vol. 12, pp. 119–144, 2005. [Online]. Available: http://www.acousticslab.org/papers/Vassilakis2005SRE.pdf

[12] A. Kameoka and M. Kuriyagawa, "Consonance theory part I: Consonance of dyads," *The Journal of the Acoustical Society of America*, vol. 45, no. 6, pp. 1451–1459, 1969.

[13] G. Von Békésy and E. G. Wever, *Experiments in hearing*. McGraw-Hill New York, 1960, vol. 8.

[14] C. F. Malmberg, "The perception of consonance and dissonance." *Psychological Monographs*, vol. 25, no. 2, pp. 93–133, 1918.

[15] W. Hutchinson and L. Knopoff, "The acoustic component of Western consonance," *Journal of New Music Research*, vol. 7, no. 1, pp. 1–29, 1978.

[16] K. Mashinter, "Calculating sensory dissonance: Some discrepancies arising from the models of Kameoka & Kuriyagawa, and Hutchinson & Knopoff," *Empirical Musicology Review*, vol. 1, no. 2, pp. 65–84, 2006.

[17] J. P. Guilford, *Psychometric Methods*. New York, NY, USA: McGraw-Hill New York, 1963.

[18] R. Van den Brink and R. P. Gilles, "The outflow ranking method for weighted directed graphs," *European Journal of Operational Research*, vol. 193, no. 2, pp. 484–491, 2009.

A Computational Model of Tonality Cognition Based on Prime Factor Representation of Frequency Ratios and Its Application

Shun Shiramatsu, Tadachika Ozono, and Toramatsu Shintani
Graduate School of Engineering, Nagoya Institute of Technology
siramatu@nitech.ac.jp

ABSTRACT

We present a computational model of tonality cognition derived from physical and cognitive principles on the frequency ratios of consonant intervals. The proposed model, which we call the Prime Factor-based Generalized Tonnetz (PFG Tonnetz), is based on the Prime Factor Representation of frequency ratios and can be regarded as a generalization of the Tonnetz. Our assumed application of the PFG Tonnetz is a system for supporting spontaneous and improvisational participation of inexpert citizens in music performance for regional promotion. For this application, the system needs to determine the pitch satisfying constraints on tonality against surrounding polyphonic music because inexpert users frequently lack music skills related to tonality. We also explore a working hypothesis on the robustness of the PFG Tonnetz against recognition errors on harmonic overtones in polyphonic audio signals. On the basis of this hypothesis, the PFG Tonnetz has a good potential as a representation of the tonality constraints of surrounding polyphonic music.

1. INTRODUCTION

Musical tonality is an important cognitive element for listening to or playing tonal music. This cognitive phenomenon depends on the perception of the consonant/dissonant interval that can be physically explained with frequency ratios and the overlap of harmonic structures between multiple tones. There are three structural properties of melodic cognition [1]:

1. **Rhythm**: Ordinal duration ratios of adjacent notes.
2. **Pitch contour**: Pattern of ups and downs of pitch changes.
3. **Tonality**: Cognitive coherence of pitch combination related to consonance, harmony, key, scale, and chord.

Understanding the tonality comparatively requires more musical knowledge or experience than the rhythm and pitch contour. Although inexpert users can intuitively input the rhythm and pitch contour with their body motion, it is comparatively difficult for them to determine pitch with tonality. Hence, a computational model of tonality cognition can help support inexpert users to play music with their intuitive body motion. We aim to formulate a computational model of tonality for enabling inexpert users to participate in playing music by inputting the rhythm and pitch contour with their body motions. In this paper, we present a model of tonality derived from only frequency ratios against tonic without musical knowledge such as key and letter notation.

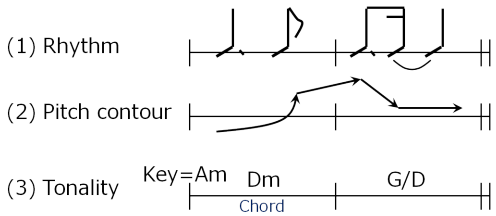


Figure 1. Three aspects for cognition of tonal melody.

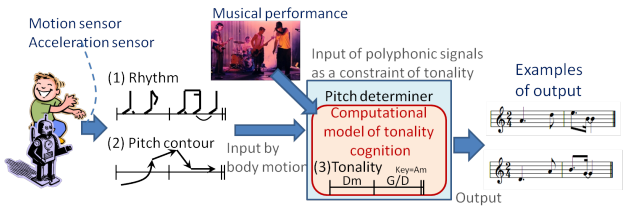


Figure 2. Application: Generating melody with tonality from only rhythm and pitch contour input by a user.

Recently, many participatory music events for regional promotion have been organized in Japan [2]. Since a broad range of participants are desired for the purpose of regional promotion, technology for supporting the participation of inexpert citizens is important. Devices or techniques that enable non-experts to play music as emotion dictates could lead to the design of novel musical interaction between citizens for regional promotion.

Clapping to the beat, swaying to the rhythm, and "call-and-response" are basic ways for participating in musical performance without IT support. We aim to provide a novel method for supporting spontaneous and improvisational participation in music performance that does not require advanced music skills or experiences. We focus particularly on spontaneous participation with sustained har-

Copyright: ©2015 Shun Shiramatsu 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.

monic sound because such participation usually requires a certain level of musical skills related to tonality. To this end, we focus on a mechanism to determine pitch having a tonality coherent with the surrounding music performance from the spontaneous rhythm and pitch contour input by the users (Figure 2). This mechanism should be helpful for encouraging spontaneous and improvisational participation in playing tonal music.

Since the (1) rhythm and (2) pitch contour depend less on musical knowledge or experience than (3) tonality, we assume that (1) and (2) can be input by an inexpert user who has less experience with music performance. Body motion is suitable for inputting (1) and (2) because they are highly relevant to body motion. The affinity between pitch contour and body motion has been described in [3]. Here, we assume the use of motion sensors or acceleration sensors for the user's input. For example, the ups and downs of a hand motion can be used to input the pitch contour.

A computational model of tonality cognition is needed to determine the pitch satisfying constraint on tonality, as shown in Figure 2. The aim to develop a tonality model not for increasing the accuracy of estimating key or chord but for controlling harmony or consonance between the surrounding polyphonic music and the system-determined pitch. Considering the recognition error on harmonic overtone in polyphonic audio signals, the model of tonality cognition for our application should be directly based on physical and cognitive principles related to the pitch frequencies of consonant intervals.

2. PRIME FACTOR-BASED GENERALIZED TONNETZ

In this section, we describe our computational model of tonality based on prime factor representation of ratios between pitch frequencies. The proposed model is derived from only the essential principles on the integer frequency ratio of consonant interval and octave equivalence.

Table 1. Correspondence between frequency ratios of just intonation intervals and the exponents z_2, z_3, and z_5.

Interval	I	#I	II	#II	III	IV
Frequency ratio	1	$\frac{16}{15}$	$\frac{9}{8}$	$\frac{6}{5}$	$\frac{5}{4}$	$\frac{4}{3}$
Cent	0.0	111.7	203.9	315.6	386.3	498.0
z_2	0	4	-3	1	-2	2
z_3	0	-1	2	1	0	-1
z_5	0	-1	0	-1	1	0

#IV	V	#V	VI	#VI	VII
$\frac{64}{45}$	$\frac{3}{2}$	$\frac{8}{5}$	$\frac{5}{3}$	$\frac{9}{5}$	$\frac{15}{8}$
609.8	702.0	813.7	884.4	1017.6	1088.3
6	-1	3	0	0	-3
-2	1	0	-1	2	1
-1	0	-1	1	-1	1

2.1 Deriving a Tonality Model from Cognitive Principles

Consonant intervals are usually formed by the frequency ratio of simple integers. When a frequency f_{tonic} of a tonic note and f form a consonant interval, the ratio of these frequencies consists of simple integers, e.g., $f = \frac{3}{2} f_{\text{tonic}}$ (perfect V) and $f = \frac{4}{3} f_{\text{tonic}}$ (perfect IV). Such a frequency ratio consisting of simple integers can be represented by the product of prime numbers, as

$$f = \Big(\prod_{p \in \mathbb{P}_n} p^{(z_p)} \Big) \cdot f_{\text{tonic}} \quad (z_p \in \mathbb{Z}), \tag{1}$$

where $\mathbb{Z}$ is the set of integers and $\mathbb{P}_n = \{2, 3, 5, \cdots, n\}$ is a set of prime numbers that are less than or equal to the upper limit n. For example, when the upper limit n of a prime number is set as 5, the perfect IV can be represented as

$$f = (2^2 \cdot 3^{-1} \cdot 5^0) \cdot f_{\text{tonic}}. \tag{2}$$

The consonant interval between f_{tonic} and f can be represented by a vector $(z_2, z_3, \cdots, z_n)$ consisting of the exponent z_p of a prime number p. This vector of the exponents is an expansion of the *prime factor representation* used in the field of number theory [4]. Although the original theory does not allow negative exponents (i.e., z_p should be a non-negative integer), we expand our representation of consonant inverval to allow negative exponents so that we can represent the integer frequency ratios of consonant intervals. For example, when the upper limit n of a prime number is 5, representations of the following consonant intervals are represented as the following vectors:

- perfect IV up: $(2, -1, 0)$
- perfect V up: $(-1, 1, 0)$
- major III up: $(-2, 0, 1)$
- perfect unison (tonic itself): $(0, 0, 0)$ (the origin)

Table 1 shows the correspondence between the frequency ratios of the pure intervals of just intonation and exponents z_p of a prime number p.

When plotting such vectors in the z_2-z_3-z_5 coordinate system, the following correspondences can be found, as shown in Figure 3. The origin point corresponds to the tonic f_{tonic}. The integer grid points close to the origin correspond to the frequency ratios of the consonant interval from f_{tonic}.

Here, *octave generalization* [5] can be applied with considering the octave equivalence between f_1 and f_2, such as $f_1 = 2^z f_2$ (z is an integer). Concretely, each point can be octave-generalized by projecting the point onto the z_3-z_5 plane (i.e., by letting $z_2 = 0$), as shown in Figure 3. In the case of the perfect IV, $(2, -1, 0)$ in the z_2-z_3-z_5 space can be projected onto $(-1, 0)$ on the z_3-z_5 plane. The pitches octave-equivalent to $(2, -1, 0)$, i.e., $(2 \pm i, -1, 0)$ where i is an integer, are also projected onto $(-1, 0)$, at the same point. In the same way, integer grid points (z_2, z_3, z_5) within an octave from a tonic (i.e., such as $1 \leq 2^{z_2} \cdot 3^{z_3} \cdot 5^{z_5} < 2$) are octave-generalized as

- perfect IV up: $(2, -1, 0) \to (-1, 0)$
- perfect V up: $(-1, 1, 0) \to (1, 0)$
- major III up: $(-2, 0, 1) \to (0, 1)$

on the z_3-z_5 plane, as shown in Figure 3.

The red triangles in Figure 4 consisting of $\big[(a, b), (a, b+1), (a+1, b)\big]$ can be regarded as representations of major triads on the root (a, b), while the blue triangles consisting of $\big[(a, b), (a+1, b-1), (a+1, b)\big]$ can be regarded as representations of minor triads on the root (a, b).

Moreover, seventh and extended chords can be formed by alternately piling up the red and blue triangles to the positive direction of the z_3 axis, i.e., the right direction in Figure 4. Major seventh and extended chords are piled up on the right of a base red triangle corresponding to the major triad. Minor seventh and extended chords are piled up on the right of a base blue triangle corresponding to the minor triad. To formulate these structures shown in Figure 4, we assume a list of integer grid points $\mathrm{chord}(a, b, \delta, m)$ on the root note (a, b).

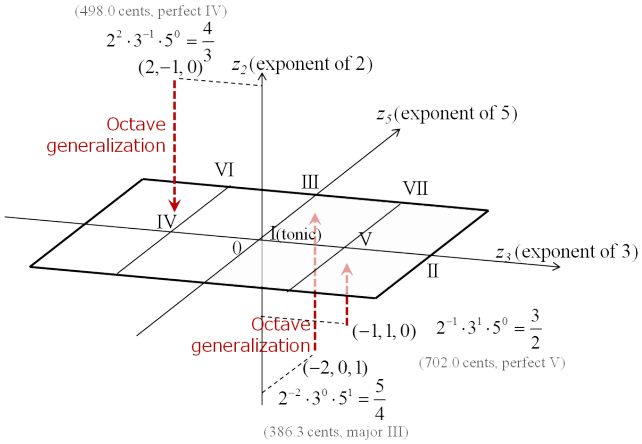


Figure 3. Octave generalization: projection onto z_3-z_5 plane by omitting z_2.

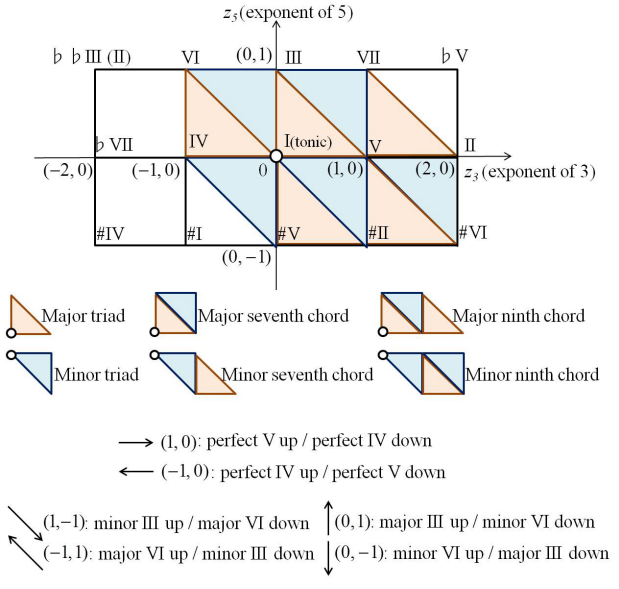


Figure 4. Proposed model: Prime Factor-based Generalized Tonnetz (5-limit).

$$\mathrm{chord}(a, b, \delta, m) = \Big[(a, b) + \sum_{i=0}^{k} \delta(i)\Big]_{k=0,1,\cdots,m} \tag{3}$$

$$\delta_{\mathrm{maj}}(i) = \begin{cases} (0, 1) & (i = 2k+1, k \in \mathbb{N}) \\ (1, -1) & (i = 2k, k \in \mathbb{N}) \end{cases} \tag{4}$$

$$\delta_{\mathrm{min}}(i) = \begin{cases} (1, -1) & (i = 2k+1, k \in \mathbb{N}) \\ (0, 1) & (i = 2k, k \in \mathbb{N}) \end{cases} \tag{5}$$

A list of integer grid points $\mathrm{chord}(a, b, \delta_{\mathrm{maj}}, m)$ represents the major triad where $m = 2$, the major seventh chord where $m = 3$, and the major ninth chord where $m = 4$. The member notes of these major chords are located in the area of $b \le z_5 \le b+1 \wedge z_3 \ge a$ at the upper right side of the root note (a, b). In contrast, $\mathrm{chord}(a, b, \delta_{\mathrm{min}}, m)$ represents the minor triad where $m = 2$, the minor seventh chord where $m = 3$, and the minor ninth chord where $m = 4$. The member notes of these minor chords are located in the area of $b - 1 \le z_5 \le b \wedge z_3 \ge a$ at the lower right side of the root note (a, b).

The positional relationships of major/minor scales against the tonic note $(0, 0)$ are similar to those of major/minor chords against a root note (a, b). Member notes of the major scale (I, II, III, IV, V, VI, VII) are distributed in the area of $0 \le z_5 \le 1$ at the upper side of the tonic $(0, 0)$, and those of the minor scale (I, II, #II, IV, V, #V, #VI) are distributed in the area of $-1 \le z_5 \le 0$ at the lower side of the tonic.

The above representation of tonality is derived only from the following two cognitive principles on frequency ratios.

1. Since the frequency ratios of a consonant interval are simple integer ratios, they can also be represented by the prime factor representation (z_2, z_3, z_5), where the integer z_p is an exponent of a prime number p (expanded to allow $z_2, z_3, z_5 < 0$).

2. Since the interval with the frequency ratio 2^z (where z is integer) is octave equivalent, (z_2, z_3, z_5) can be projected onto the z_3-z_5 plane by omitting z_2 for octave generalization.

In the other words, our tonality model is derived only from the cognitive principles on frequency ratios. Musical knowledge about the pitch notation, scale, and chord is used not for deriving our tonality model but rather for interpreting the representations appearing in the derived model. The interpretations of the representations are as follows:

- The origin $(0, 0)$: A tonic of scales
- Integer grid points (z_3, z_5) close to the origin: Candidates for scale notes
- A vector $(1, 0)$: Perfect V up
- A vector $(-1, 0)$: Perfect IV up
- A vector $(0, 1)$: Major III up

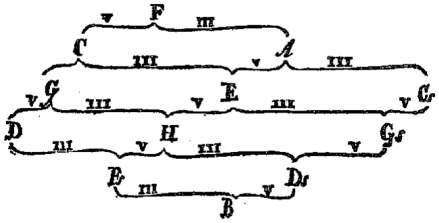


Figure 5. Euler's Tonnetz.

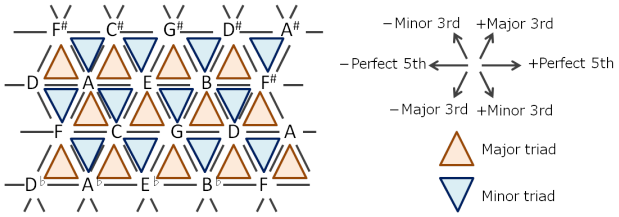


Figure 6. Riemann's Tonnetz.

- A vector $(1, -1)$: Minor III up
- A triangle $\big[(a, b), (a, b+1), (a+1, b)\big]$: Major triad on the root (a, b)
- A triangle $\big[(a, b), (a+1, b-1), (a+1, b)\big]$: Minor triad on the root (a, b)
- Integer grid points $\mathrm{chord}(a, b, \delta_{\mathrm{maj}}, m)$: Major chords on the root (a, b)
- Integer grid points $\mathrm{chord}(a, b, \delta_{\mathrm{min}}, m)$: Minor chords on the root (a, b)
- $b \leq z_5 \leq b+1 \wedge z_3 \geq a$: Distribution area of major chords on the root (a, b)
- $b-1 \leq z_5 \leq b \wedge z_3 \geq a$: Distribution area of minor chords on the root (a, b)
- $0 \leq z_5 \leq 1$: Distribution area of major scale notes on the tonic $(0, 0)$
- $-1 \leq z_5 \leq 0$: Distribution area of minor scale notes on the tonic $(0, 0)$

2.2 Comparison of Proposed Model and Tonnetz

Our derived model of tonality is topologically similar to the Tonnetz [6], which was originally proposed by Leonhard Euler in 1739 (Figure 5) and was expanded upon by Hugo Riemann in 1880 (Figure 6). In the Tonnetz, pitch notations are connected by three types of link: perfect V (opposite of perfect IV), major III (opposite of minor VI), and minor III (opposite of major VI). In our model, these links respectively correspond to the vectors $(1, 0)$ (opposite of $(-1, 0)$), $(0, 1)$ (opposite of $(0, -1)$), and $(1, -1)$ (opposite of $(-1, 1)$).

The Tonnetz was expanded as Neo-Riemannian theory and mathematically formulated in the 1980s [7, 8]. It was typically expanded to torus or spiral representations [9]

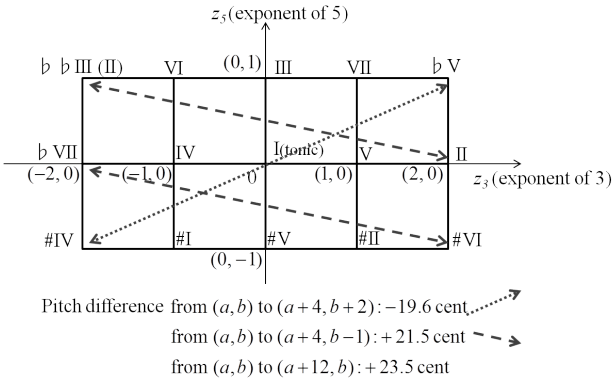


Figure 7. Pitch differences of enharmonic pairs of integer grid points.

considering circularity due to enharmonic equivalence. Enharmonic pairs of tones are also represented as vectors $(4, 2)$, $(4, -1)$, and $(12, 0)$ in our proposed model, as shown in Figure 7.

There are three key differences between our model and the conventional Tonnetz studies.

1. **Clear correspondence between the model and the physical and cognitive principles**. Although the conventional Tonnetz was originally formalized for representing the relationships among consonant intervals, the correspondence between the model and the principles on frequency ratios was not clear. Our proposed derivation process enables us to clearly understand the correspondence because it is directly derived from the principles on frequency ratios. Moreover, this feature should have a high affinity for processing polyphonic audio signals with the recognition error on harmonic overtone.

2. **Tonic representation**. In our proposed model (Figure 4), the origin $(0, 0)$ on the z_3-z_5 plane has the role of the tonic of scale. Since the tonal characteristics of each point depend on the relative position from the tonic, equivalent scales on different tonics can be represented by a same pattern on the z_3-z_5 coordination system. This feature also enables us to formulate computational representations of major/minor chords, such as Formulas (3), (4), and (5).

3. **Natural expandability to a higher dimensional space for the n-limit just intonation**. The above $n = 5$ setting for the upper limit of prime numbers can be varied to expand our tonality model. When $n = 7$, integer grid points in the z_3-z_5-z_7 space can represent the 7-limit just intonation [10], as shown in Figure 8. When $n = 11$, integer grid points in the z_3-z_5-z_7-z_{11} space can represent the 11-limit just intonation.

On the basis of the above, our proposed model can be regarded as a generalization of the Tonnetz. We call it Prime Factor-based Generalized Tonnetz (PFG Tonnetz).

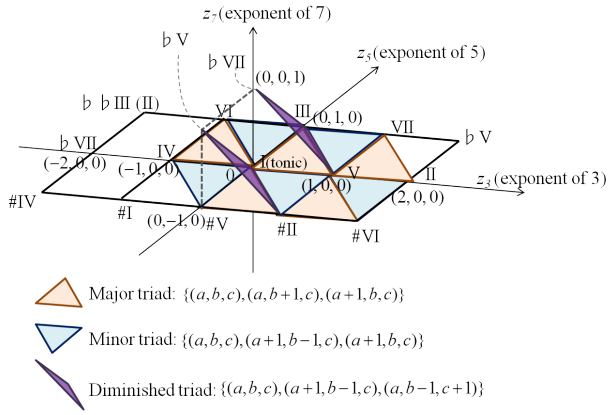


Figure 8. 7-limit PFG Tonnetz.

The model based on the z_3-z_5-$\cdots$-z_n space is called n-limit PFG Tonnetz because it represents the n-limit just intonation [10].

Hereafter, we regard PFG Tonnetz without specifying n as the 5-limit PFG Tonnetz (Figure 4) because the 5-limit PFG Tonnetz represents the usual just intonation, i.e., 5-limit just intonation. The 5-limit PFG Tonnetz can easily be visualized on the z_3-z_5 plane. The topological similarity of the 5-limit PFG Tonnetz to the conventional Tonnetz is easier to understand than that of other n-limit PFG Tonnetz.

3. APPLYING PFG TONNETZ TO DETERMINING PITCH WITH TONALITY CONSTRAINT

As discussed in Section 1, we aim to apply our model, the PFG Tonnetz, to a module to determine the pitch satisfying constraint on coherence of tonality against surrounding polyphonic music. 3D motion sensors, such as the Microsoft Kinect[1] or the Intel RealSense 3D Camera[2] , can be used for recognizing users' hand motions, e.g., the heights of hands. Our system needs to convert a recognized hand height $x(t)$ at given time t into a tonal pitch frequency $f(t)$ that satisfies a constraint on the tonality of the surrounding polyphonic music, as

$$f(t) = \underset{f(t)\ \text{satisfies}\ c(t)}{\arg\min} \Big(|f(t) - f'(t)| \Big), \qquad (6)$$

$$f'(t) = f_{\text{tonic}} \cdot \exp\big(\alpha(x(t) - x_{\text{tonic}})\big), \qquad (7)$$

where $f'(t)$ is an atonal frequency that simply corresponds to $x(t)$, $c(t)$ is a tonality constraint at the time t, $\exp(\cdot)$ is the exponential function, x_{tonic} is a basis location corresponding to f_{tonic}, and α is a parameter to adjust the ratio between the change of $x(t)$ and that of $f'(t)$.

A module for the online F0 estimation of the surrounding polyphonic music is needed to deal with the tonality constraint $c(t)$. Although F0 estimation of the polyphonic audio signals generally cannot avoid recognition errors on harmonic overtones, a representation of the tonality constraint based on the PFG Tonnetz should be robust against such errors because the frequencies of harmonic overtones, i.e., integral multiples of a true frequency, are located at grid points close to the true pitch in the PFG Tonnetz space.

In future work, we intend to empirically verify this working hypothesis on the robustness against the recognition error on harmonic overtones. We primarily need to formulate a representation of tonality constraint $c(t)$ by integrating the PFG Tonnetz and the F0 estimation of polyphonic audio signals. This representation should be learnable from training data of polyphonic music and should be empirically compared with a representation by integrating the conventional chroma vector and the F0 estimation through an experiment. For example, if the tonality constraint is represented as probabilistic prediction models over the PFG Tonnetz space or over the chroma vector, the two representations can be compared by prediction ability such as the perplexity metric.

4. CONTEXT OF THIS STUDY AND RELATED WORKS

4.1 Tonality Models

Pitch representations based on Prime Factor Representation have been proposed in other studies [11, 12]. However, these works did not consider the relationship between their model and the Tonnetz. Direct derivation of the Tonnetz on the basis of the Prime Factor Representation of frequency ratio is a viewpoint unique to the present study.

There have been many models and theories related to tonality cognition. For example, a key estimation method based on the Cycle of Fifth [13] has been proposed. The Tonnetz has also been studied by Neo-Riemannian theorists [7] and applied to instrument interfaces such as the isomorphic keyboard [14]. The PFG Tonnetz we propose in the present work has three advantages as aforementioned: (1) it clearly corresponds to physical and cognitive principles on the integer frequency ratios, (2) it has tonic representation, and (3) it is naturally expandable to higher dimensions for n-limit just intonation. If the hypothesis on the robustness against recognition errors on harmonic overtones is empirically verified in future work, it can also be our contribution.

4.2 Related Applications

Figure 9 shows a smartphone application, TonalityTouch[3] , developed in our past study. TonalityTouch can convert the user's multi-touch location into consonant pitch frequencies with tonality. The scale for converting the location to the pitch frequency can be automatically generated on the basis of the PFG Tonnetz. However, TonalityTouch does not consider the constraint on tonality against the surrounding music.

KAGURA [15] is a digital instrument with visual effects based on body motion sensing. SWARMED [16] and massMobile [17] are systems for supporting participatory mu-

[1] https://www.microsoft.com/en-us/kinectforwindows/

[2] http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-3d-camera.html

[3] https://play.google.com/store/apps/details?id=org.toralab.music.beta

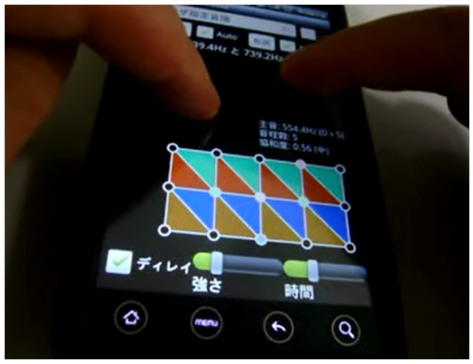


Figure 9. TonalityTouch: Smartphone application based on PFG Tonnetz.

sical performance using smartphones. Although these systems are related to our application, they do not focus on any computational model for converting the spontaneous input of pitch contour to pitch frequency satisfying the constraint of the tonality against surrounding polyphonic music, which is our focus in the current study.

4.3 Application to Music Event for Regional Promotion

We have been dealing with technologies for supporting public participation and collaboration [18, 19]. To facilitate public collaboration in local communities, building a conciliatory community through "ice-breaker activities" is important. Since music has a social functionality for enhancing positive emotions through sharing body motion [20], we aim to apply our model to such ice-breaker activities through participatory local music events. We need to investigate and verify whether the PFG Tonnetz can contribute to spontaneous and improvisational participation in music performance and whether such support can contribute to ice-breaking in local communites.

5. CONCLUSION AND FUTURE WORK

We formulated the PFG Tonnetz, a model of tonality cognition based on simple principles on frequency ratios of consonant intervals, i.e., consonant intervals can be represented by the frequency ratios of simple integers. The derivation of the proposed model is based on the Prime Factor Representation of the frequency ratios. The PFG Tonnetz can be regarded as a generalization of the conventional Tonnetz and can be applied to a representation of the tonality constraint of surrounding polyphonic music. The representation should be robust against recognition errors on harmonic overtone in polyphonic audio signals because the frequencies of harmonic overtones are located at grid points close to the true pitch in the PFG Tonnetz space. This working hypothesis should be empirically verified through experiments in future.

We are also planning to apply the PFG Tonnetz to support the spontaneous and improvisational participation of inexperts in local music events for regional promotion. We will utilize such functionality for ice-breaker activities in local communities. To do this, we will develop a system based on the PFG Tonnetz by integration with motion sensors.

Acknowledgments

This study was partially supported by a Grant-in-Aid for Young Scientists (B) (No. 25870321) from JSPS.

6. REFERENCES

[1] J. B. Prince, "Contributions of pitch contour, tonality, rhythm, and meter to melodic similarity," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 40, no. 6, pp. 2319–2337, 2014.

[2] Sakai Urban Policy Institute, "Report on investigating regional promotion by citizens' initiative through organizing music events," http://www.sakaiupi.or.jp/30.products/31.resarch/H22/H22_music.pdf, 2011, (in Japanese).

[3] M. Kan, "An audience-participatory concert emphasizing physical expression : A case study promoting an understanding of polyphonic music," *Bulletin of the Center for Educational Research and Training, Faculty of Education, Wakayama University*, vol. 18, pp. 121–129, 2008, (in Japanese).

[4] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, 1989.

[5] E. M. Burns and W. D. Ward, "Intervals, scales, and tuning," *The psychology of music*, vol. 2, pp. 215–264, 1999.

[6] R. Behringer and J. Elliot, *Linking Physical Space with the Riemann Tonnetz for Exploration of Western Tonality*. Nova Science Publishers, 2010, ch. 6, pp. 131–143.

[7] W. Hewlett, E. Selfridge-Field, and E. Correia, *Tonal Theory for the Digital Age*, ser. Computing in Musicology. Center for Computer Assisted Research in the Humanities, Stanford University, 2007, vol. 15.

[8] D. Tymoczko, "The generalized tonnetz," *Journal of Music Theory*, vol. 56, no. 1, pp. 1–52, 2012.

[9] E. Chew, *Mathematical and Computational Modeling of Tonality: Theory and Applications*, ser. International Series in Operations Research & Management Science. Springer, 2013, vol. 204.

[10] H. Partch, *Genesis of a music: an account of a creative work, its roots and its fulfilments*. Da Capo Press, 1974.

[11] J. Monzo, *JustMusic: A New Harmony Representing Pitch as Prime Series*, 4th ed. J. Monzo, 1999.

[12] D. Keislar, "History and principles of microtonal keyboards," *Computer Music Journal*, pp. 18–28, 1987.

[13] T. Inoshita and J. Katto, "Key estimation using circle of fifths," in *Advances in Multimedia Modeling*. Springer, 2009, pp. 287–297.

[14] A. Milne, W. Sethares, and J. Plamondon, "Isomorphic controllers and dynamic tuning: Invariant fingering over a tuning continuum," *Computer Music Journal*, vol. 31, no. 4, pp. 15–32, 2007.

[15] SHIKUMI DESIGN, "Kagura - the motion perform instrument," https://www.youtube.com/watch?v=SvOfu9NifyY, 2015.

[16] A. Hindle, "Swarmed: Captive portals, mobile devices, and audience participation in multi-user music performance," in *Proceedings of the 13th International Conference on New Interfaces for Musical Expression*, 2013, pp. 174–179.

[17] N. Weitzner, J. Freeman, Y.-L. Chen, and S. Garrett, "massmobile: towards a flexible framework for large-scale participatory collaborations in live performances," *Organised Sound*, vol. 18, no. 01, pp. 30–42, 2013.

[18] S. Shiramatsu, T. Ozono, and T. Shintani, "Approaches to assessing public concerns: Building linked data for public goals and criteria extracted from textual content," in *Electronic Participation. 5th IFIP WG 8.5 International Conference, ePart 2013*, ser. Lecture Notes in Computer Science, vol. 8075. Springer, 2013, pp. 109–121.

[19] S. Shiramatsu, T. Tossavainen, T. Ozono, and T. Shintani, "A goal matching service for facilitating public collaboration using linked open data," in *Electronic Participation. 6th IFIP WG 8.5 International Conference, ePart 2014*, ser. Lecture Notes in Computer Science, vol. 8654. Springer, 2014, pp. 114–127.

[20] H. Terasawa, R. Hoshi-Shiba, T. Shibayama, H. Ohmura, K. Furukawa, S. Makino, and . Okanoya, "A network model for the embodied communication of musical emotions," *Japanese Cognitive Science Society*, vol. 20, no. 1, pp. 112–129, 2013, (in Japanese).

ANALYSIS AND RESYNTHESIS OF THE HANDPAN SOUND

Eyal Alon
AudioLab, Department of Electronics,
University of York, UK
ea553@york.ac.uk

Dr. Damian T. Murphy
AudioLab, Department of Electronics,
University of York, UK
damian.murphy@york.ac.uk

ABSTRACT

Handpan is a term used to describe a group of struck metallic musical instruments, which are similar in shape and sound to the Hang [1] (developed by PANArt in 2000). The handpan is a hand played instrument, which consists of two hemispherical steel shells that are fastened together along the circumference. The instrument usually contains a minimum of eight notes and is played by delivering rapid and gentle strikes to the note areas. An experimental procedure has been designed and implemented to record, analyse, and resynthesise the handpan sound. Four instruments from three different makers were used for the analysis, giving insight into common handpan sound features, and the origin of signature amplitude modulation characteristics of the handpan. Subjective listening tests were conducted aiming to estimate the minimum number of signature partials required to sufficiently resynthesise the handpan sound.

1. INTRODUCTION

Handpan is a term used to describe a group of struck metallic percussion instruments, which are similar in shape and sound to the Hang [1] (developed by PANArt Ltd. in January 2000 [3]). The handpan is a hand played instrument, which consists of two hemispherical steel shells that are fastened together along the circumference. The instrument is played by delivering rapid and gentle finger strikes to individual notes. Similar to steel pan notes, the frequencies produced from the Hang's principal modes of vibration in each note area have a 1:2:3 ratio [4]. An additional frequency component found in the spectrum of the Hang at approximately 85 Hz is associated with the cavity (Helmholtz) resonance frequency.

In October 2014, there were approximately 80 handpan makers worldwide [5]. Some noteable makers are Pantheon Steel [6], Zen Handpans [7], CFoulke [8], and Saraz Handpans [9]. As seen in Figure 1, the handpan typically consists of eight or more notes. Amongst makers and players, the notes are commonly known as "note-fields" due to the fact that strikes delivered to different areas of the note-field will emphasize specific harmonics, resulting in a different timbre. An objective standard for classification of handpan quality does not yet exist, however some discussion of this amongst makers and enthusiasts has occurred [10]. Furthermore, no standard exists for handpan making or tuning so each maker creates instruments with different materials, tools, dimensions, and shell and note-field architectures.

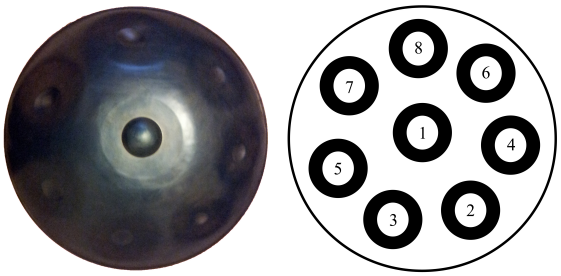


Figure 1. Top view of a handpan with eight note-fields.

This paper presents the design and implementation of an experimental procedure to measure, analyse, and resynthesise the signature handpan sound. Results from a listening test conducted in order to asses the quality of the resynthesised signals go some way towards determining the minimum number of partials required to sufficiently resynthesise the handpan sound.

2. MEASUREMENT AND ANALYSIS

A handpan frame was constructed from extruded aluminium rods and was designed to support the handpan, excitation mechanism, and microphone securely within the anechoic chamber when making measurements. This required the frame to be strong enough to provide support whilst at the same time minimise the influence of the frame itself on the measurements. It was noted from previous research on the Hang that varying the spacing of the player's knees can influence the tuning of the Helmholtz resonance frequency by effectively changing the acoustical "length" of the neck [11]. Investigation of the effects of the handpan cavity on the overall sound is beyond the scope of this paper and should be considered for future work. The size of the frame was adjusted to provide support as close to the rim of the handpan as possible, as well as to ensure no obstructions between the microphone and each of the note-fields. The Note-Field Excitation Mechanism (NFEM) was formed of a torsional spring (2.7mm wire diameter, 30mm

[1] Hang® is a registered trademark and should not be used to describe other musical instruments such as handpans [1], nor should the term handpan be used to refer to the Hang® [2].

Copyright: ©2015 Eyal Alon 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.

body length) fixed at one end and attached to a rounded rubber tip at the other. The NFEM was used by pulling the rubber tipped end of the spring back to a fixed position and then releasing to generate a strike to an individual note-field. This method of excitation was preferred over sinusoidal excitation or finger force as it allowed excitation of various positions with repeatable strikes that are similar in nature to finger strikes.

A previous study of the steel pan used sandbags to minimize radiation from surrounding notes [12]. Some steel pan makers use magnets to achieve a similar effect as the cross-talk between notes can interfere with the tuning process [13]. In order to determine the signature sound of an isolated handpan note-field, and to estimate the contribution of surrounding notes to the overall handpan sound, each note-field sound was measured in two configurations:

- Damped: Magnetic absorbing pads were placed to cover all note-fields other than the one currently being recorded, in order to suppress their vibration and contribution to the recorded signal.
- Undamped: No magnetic absorbing pads were used to dampen surrounding note-fields.

The measurement procedure for an individual handpan note-field was implemented in the following sequence:

1. Securely place the handpan inside the frame.
2. Position the microphone and NFEM appropriately.
3. Adjust recording levels to avoid clipping.
4. Deliver a strike to the note-field allowing the sound to decay to an inaudible level.
5. Place magnetic absorbing pads on all surrounding note-fields prior to delivering an additional strike.

Eight strikes were delivered to each note-field in each configuration and were allowed to decay for ten seconds prior to the following strike. Once the handpan and microphone were positioned securely, they were not moved until all notes of the handpan were recorded. Table 1 provides a key of measurements taken for all four investigated instruments. All note-fields were measured in both undamped and damped configurations.

2.1 Identification of Signature Partials

In the context of this paper, a signature partial is defined as one of a number of highest magnitude detected peaks in the spectrum of the handpan sound. The Energy Decay Relief (EDR) method is useful for smoothing transient features and amplitude modulations present in a signal [14,15], thus allowing an easier identification of the handpan's signature partial frequencies and corresponding decay rates. For each recorded handpan note, a single EDR analysis frame was used to extract the frequency values of signature partials. This frame was chosen as the first to follow the transient onset of the recorded note (approximately 4-10 ms), in order to avoid erroneous frequency selection due to the broadband nature of the note onset.

Inst. 1	Inst. 2	Inst. 3	Inst. 4
A_3	Ab_3	$A\#_3$	A_4
B_2	B_3	A_3	B_4
B_3	C_3	A_4	C_4
$C\#_4$	C_4	D_3	C_5
D_4	D_4	D_4	D_4
E_4	Eb_4	E_4	E_4
$F\#_3$	G_3	F_4	F_4
$F\#_4$	G_4	G_4	G_3
-	-	-	G_4

Table 1. A key of note measurements taken for all four instruments investigated in this project.

To improve the accuracy of the identified frequency values associated with each peak, a parabolic interpolation method was used [16]. Table 2 shows three detected partials (in order of descending magnitude from left to right) for all eight note-fields of Instrument 3, and their corresponding frequency value ratios. For seven out of eight note-fields, the three highest magnitude partials detected have an approximate 1:2:3 frequency value ratio. The third highest magnitude partial of the undamped D_3 note-field is approximately 4 times the value (i.e. the double octave) of the fundamental frequency, which produces an approximate 1:2:4 frequency ratio. The presence of this partial seems to suggest a strong coupling between the D_3 and D_4 note-fields. This emphasises the D_5 frequency (c. 592 Hz), which is the octave partial of the D_4 note-field. This suggestion is strengthened by examining the three highest magnitude partials detected from the damped D_3 signal: 152.2 Hz, 298.1 Hz, and 449.2 Hz which have a frequency ratio of approximately 1:2:3.

2.2 Decay Rate Estimation

Upon detection of the signature partials it is desirable to estimate their corresponding decay rates. Musical instrument decay times have previously been estimated in dB/sec [17] or by calculating quality factors [18, 19]. Estimation of modal decay rates can also be achieved by calculating T60 values [20] using EDR plots [15]. In the context of this paper, the PD_{60} is defined as the amount of time it takes for a partial to decay by 60 dB from its initial magnitude value. The highest magnitude partial for an individual note-field measurement was used to determine the -60 dB threshold. To implement this, MATLAB's `polyfit` function was used to calculate the coefficients of a 2nd degree polynomial that best fits a section of the corresponding decay curve (using a least-squares method [21]). In order to select the appropriate section for calculation of the polynomial, the gradient of the selected frequency bin over time was calculated. Where this gradient approaches zero represents the point where the decay curve reaches the noise floor, and this can be seen as a suitable end point for best-fit calculation.

Table 3 displays the mean PD_{60} decay times, standard deviations, and minimum and maximum values of the three highest magnitude partials for individual instruments, note

Note-field	Partial 1, freq. ratio	Partial 2, freq. ratio	Partial 3, freq. ratio
$A\#_3$	234.8 Hz 1	697.7 Hz 2.97	463.3 Hz 1.97
A_3	226.3 Hz 1	444.6 Hz 1.96	664.3 Hz 2.94
A_4	444.1 Hz 1	884.3 Hz 1.99	1325 Hz 2.98
D_3	152.1 Hz 1	297.7 Hz 1.96	592.3 Hz 3.89
D_4	591.8 Hz 1.99	297.1 Hz 1	884.1 Hz 2.98
E_4	334.6 Hz 1	664.1 Hz 1.98	993.4 Hz 2.97
F_4	700.6 Hz 1.99	352 Hz 1	1053 Hz 2.99
G_4	395.5 Hz 1	788.3 Hz 1.99	1175 Hz 2.97

Table 2. Three signature partials and corresponding frequency ratios (relative to the fundamental frequency), of all eight undamped note-fields of Instrument 3. Partials are sorted in order of descending magnitude from left to right.

groups, and all instruments. The three note groups are: low (B_2-B_3), mid (C_4-E_4), and high (F_4-C_5). Generally, the mean PD_{60} values decrease for higher register note groups. Despite this, the longest measured PD_{60} value (5.9 s) is from the mid note group. Instrument 1 and Instrument 2 have very similar results for all parameters, possibly due to the fact that they are both from the same handpan maker. Instrument 3 and Instrument 4 have relatively short average PD_{60} values, which could be due to the fact that they contain more higher register notes compared to Instrument 1 and Instrument 2.

Instrument/ note group	Mean PD_{60} (s)	Standard deviation	Min (s)	Max (s)
Instrument 1	3.3	1.1	1.7	5.9
Instrument 2	3.3	1.2	1.6	5.9
Instrument 3	2.8	0.7	1.4	4.0
Instrument 4	2.1	0.5	0.9	3.4
Low (B_2-B_3)	3.2	0.9	1.6	5.1
Mid (C_4-E_4)	3.0	1.2	1.2	5.9
High (F_4-C_5)	2.5	0.9	0.9	4.2
All instruments	2.9	1.0	0.9	5.9

Table 3. Mean PD_{60} decay times, standard deviations, and minimum and maximum values of the three highest magnitude partials for individual instruments, note groups, and all instruments.

2.3 Amplitude Modulations

Several partials in many of the measured handpan signals exhibit amplitude modulations. In order to calculate the rate of modulation, an algorithm was developed that finds the local minima in the spectrogram of a given partial and calculates the mean number of samples between the minima. The rate of modulation is then estimated by calculating the inverse of the mean number of samples. Table 4 displays the estimated amplitude modulation rates for several partials from Instrument 1.

Note-field	Frequency bin (Hz)	AM rate (Hz)
A_3	662	3.3
A_3	438	8.6
B_3	248	3.9
$F\#_3$	373	3.1
$F\#_4$	374	3.3

Table 4. Estimated amplitude modulation rates for several partials from Instrument 1.

2.4 Undamped and Damped Measurements

Upon excitation of the handpan's note-field, surrounding note-fields are also excited. This phenomenon is known as "sympathetic vibration" and has been previously investigated in other musical instruments such as the harp [22]. Comparing spectrograms of signals produced in the undamped and damped configurations provided insight regarding the origin of the handpan's signature amplitude modulation characteristics, and the contribution of surrounding note-fields to the overall handpan sound.

The significant reduction in amplitude modulation depth on partials in the damped signals, when compared to their corresponding undamped signals strengthens the following hypothesis: The signature amplitude modulation characteristics in the handpan sound are due to a slight mismatch in tuning of signature partials on separate note-fields. High-resolution methods such as ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques), can be used to identify the frequency values of these closely spaced partials [22] and this will be a subject of future work.

3. RESYNTHESIS

The time domain waveform of the handpan sound can be thought of as having two stages: attack and release. The attack is associated with a transient, broadband onset whilst the release is associated with a mostly sinusoidal steady state decay.

3.1 Steady State

The signature partials and corresponding decay rates identified in Section 2 were used to resynthesise the steady state stage of the handpan sound. The frequencies of the partials detected were used to set the oscillators for resynthesis, whilst the PD_{60} decay times were used to calculate the required exponential decay time constant. The phase, $\phi(n)$, is given by:

$$\phi(n) = 2\pi \cdot f_{sin} \cdot t(n) \quad (1)$$

where f_{sin} is the frequency of the signature partial, and $t(n)$ is the time value at sample number n (sample rate = 44.1 kHz). The initial peak magnitude, A, used for resynthesis of an individual partial is given by:

$$A = EDR_{max} \cdot 10^{\frac{A_{dB}}{20}} \tag{2}$$

where EDR_{max} is the maximum value of the EDR, and A_{dB} is the initial magnitude (in decibels) of the 2nd degree polynomial described in Section 2.2. The initial sinusoidal vector, $y_{sin}(n)$, is given by:

$$y_{sin}(n) = A \cdot \sin(\phi(n)) \tag{3}$$

The exponential decay time constant, τ, for the highest magnitude partial is given by:

$$\tau = \frac{\text{PD}_{60}}{-3} \tag{4}$$

where PD_{60} is the estimated PD_{60} decay time for the signature partial. The exponentially decaying sinusoidal vector, $y_r(n)$, is given by:

$$y_r(n) = y_{sin}(n) \cdot e^{\frac{t(n)}{\tau}} \tag{5}$$

The summed resynthesised signal, containing k desired partials, $y_{allr}(n)$, is given by:

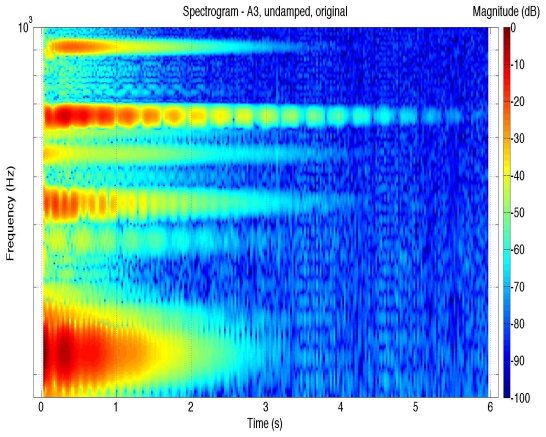
$$y_{allr}(n) = y_{r1}(n) + y_{r2}(n) + ... + y_{rk}(n) \tag{6}$$

3.2 Transient Stage

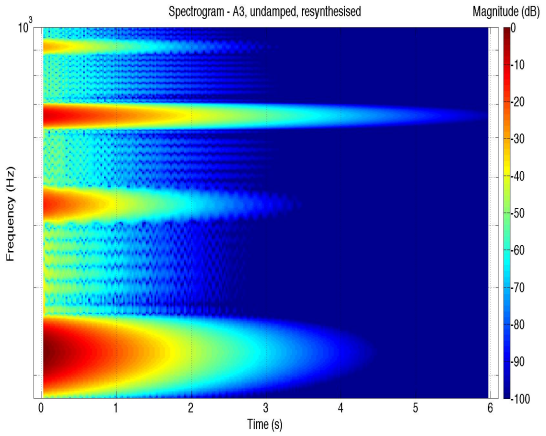
Attack transients are essential for the discrimination and identification of various musical instruments [23]. Whilst transient analysis is beyond the the scope of this paper, in order to increase the level of similarity between the resynthesised and original handpan signals, some method of transient modelling is required. In an attempt to isolate the transient portion of a struck note, all note-fields and the port hole of a handpan were covered with magnetic absorbing pads. The port hole was covered in order to reduce the presence of the Helmholtz resonance frequency in the measured signal. Then, the NFEM was used to strike the interstitial area of the handpan in between two note-fields. The signal was cropped at 10 ms, tapered and zero padded to match the length of the resynthesised steady state handpan signal. The transient and steady state signal were then convolved to produce an attack with a higher degree of similarity to the original handpan sound. The convolution of two signals can be interpreted as the multiplication of their spectrum [24], so any spectral component that is not present in both input signals, will not be present in the output signal. Convolution was preferred over simple addition of the transient and steady state signals following informal listening tests, the results of which showed that the convolved signals sounded better than their corresponding summed signals.

3.3 Amplitude Modulations

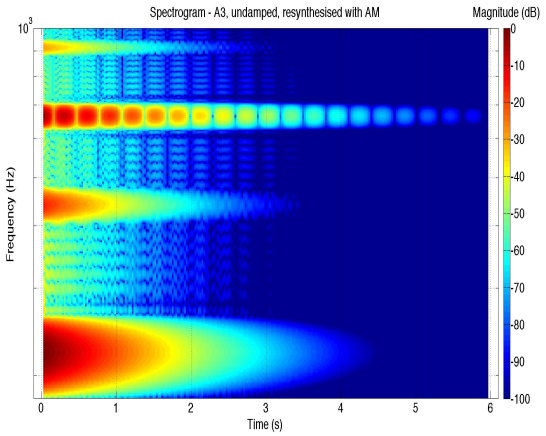
As detailed in Section 2.3, the signature handpan sound can exhibit amplitude modulations on individual or multiple partials at different modulation depths and rates. To



(a)

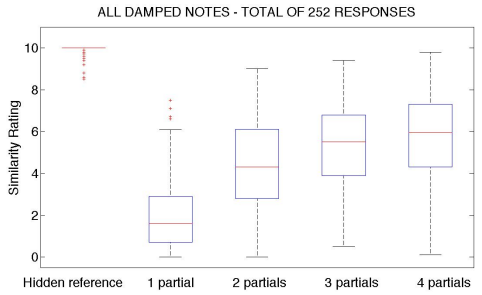


(b)

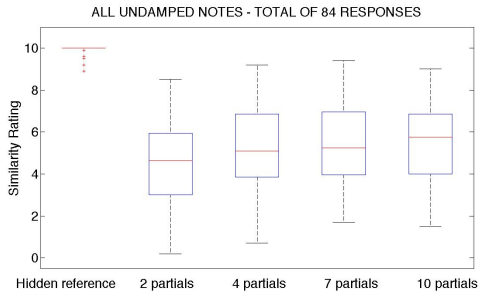


(c)

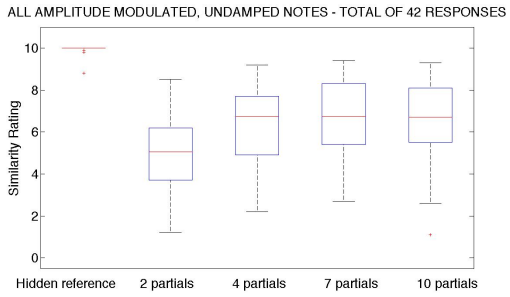
Figure 2. Spectrograms of the: (a) original; (b) resynthesised; and (c) resynthesised with AM signals. The amplitude modulated partial's frequency value shown in (c) is 662 Hz, with a modulation rate of approximately 3.3 Hz. The amplitude modulating oscillator's frequency value was set to 665.3 Hz.



(a)



(b)



(c)

Figure 3. Boxplots produced for: (a) damped (252 responses); (b) undamped (84 responses); and (c) undamped with AM (42 responses) signals respectively.

include these characteristic amplitude modulations in the resynthesised signal, it is possible to exploit the fact that the linear superposition of two simple harmonic vibrations with similar frequencies leads to periodic amplitude vibrations [19]. Figure 2 displays spectrograms of the: (a) original; (b) resynthesised; and (c) resynthesised with AM signals. The amplitude modulated partial's frequency value shown in 2(c) is 662 Hz, with a modulation rate of approximately 3.3 Hz. The amplitude modulating oscillator's frequency value was set to 665.3 Hz. Comparing the 662 Hz partial in both 2(a) and 2(c) shows a high degree of amplitude modulation rate similarity.

4. LISTENING TEST

In order to assess the quality of the resynthesised handpan sounds, a listening test was designed in order to judge the degree of similarity between the handpan recordings and resynthesised versions created using different numbers of partials. As such the results should go some way toward indicating the number of partials required for the sufficient resynthesis of the handpan sound. Three groups of resynthesised handpan sounds were investigated: damped, undamped and undamped amplitude modulated. Three note registers (low, mid, high) were tested for the damped notes of two instruments.

Each question presented to participants contained five different stimuli. The stimuli for resynthesised damped handpan signals were: 1, 2, 3, and 4 partials, whereas the stimuli for resynthesised undamped handpan signals were: 2, 4, 7, and 10 partials. The fifth stimuli for both configurations was the hidden reference, which is an identical copy of the original audio signal. The difference in number of partials used for resynthesis of the undamped and damped signals is due to the observation that the damped signals contain less signature partials then their corresponding undamped signals.

Participants were asked to rate the similarity of each of the presented audio signals to the reference audio on a scale of 0-10 (with accuracy of a single decimal point). A score of 0 indicated that the corresponding audio sample was perfectly dissimilar to the reference audio, whilst a score of 10 indicated that the audio sample was perfectly similar to the reference audio.

The resynthesised audio samples required additional processing prior to implementation of the subjective listening tests. A section of background noise was cropped immediately before or after the original handpan audio signal and added to the resynthesised signal. Additionally, normalising was also required to bring the original and resynthesised audio signals to the same loudness level. This was achieved by calculating the RMS value for the original and resynthesised signals and scaling each signal appropriately to achieve the desired global RMS level.

MATLAB's `boxplot` function was used to analyse the results of the subjective listening test. Figure 3 shows boxplots of the listening test results for: (a) damped (252 responses); (b) undamped (84 responses); and (c) undamped with AM (42 responses) signals respectively.

Examining 3(a), which contains the boxplots produced for all damped note signals, shows a clear increase in the median similarity rating with increased amount of partials. Examining 3(b), which contains the boxplots produced for all undamped note signals, also shows a slight increase in median similarity rating with increased amount of partials, however this is not as significant for the results in 3(a). Examining 3(c), which contains the boxplots produced for all undamped, amplitude modulated signals shows an increase in median similarity rating for all stimuli, compared to 3(b). For instance, the median rating for the 4 partial stimulus in 3(b) is 5.1, whereas the median value is 6.75 for 3(c). This suggests that the amplitude modulations present in some of the handpan signals is a signature component and must be included in order to sufficiently resynthesise the handpan sound. Additionally, this suggests that addition of amplitude modulation in the resynthesised signal

reduces the number of partials required to achieve higher similarity ratings.

5. CONCLUSIONS

This paper presented the results of an experimental procedure to measure and analyse the handpan sound. Analysis and comparison of undamped and damped measurements strengthened the hypothesis that the signature amplitude modulation characteristics in the handpan sound are due to a slight mismatch in tuning of signature partials on separate note-fields. Based on the analysis results, resynthesised sounds were produced and compared to measured sounds in a listening test that aimed to determine the minimum number of partials required to sufficiently resynthesise the signature handpan sound. The results showed the highest median ratings given to resynthesised signals with 4-7 partials, and an additional oscillator used to model the handpan's signature amplitude modulations. Future work should focus on accurate modelling of the attack transient, investigation of the handpan cavity acoustics, and accurate identification of closely spaced partials using high-resolution methods.

Acknowledgments

This work is part of the research carried out within the scope of an MSc (by Research). Special thanks to Dr. John Szymanski, Francis Stevens, and Andrew Chadwick.

6. REFERENCES

[1] F. Rohner, "Letter of appreciation to Samsung Electronics Ltd. Co." [Online]. Available: http://www.hang.ch/en/news/category/news

[2] F. Rohner and S. Schärer, "Newsletter PANArt, May 19th." [Online]. Available: http://www.hangblog.org/newsletter-panart-may-19th-2010/

[3] F. Rohner and S. Schärer, "History, Development and Tuning of the HANG," *ISMA 2007*.

[4] D. Wessel, A. Morrison, and T. Rossing, "Sound of the HANG," *Proceedings of Meetings on Acoustics*, vol. 4, no. 1, 2008. [Online]. Available: http://scitation.aip.org/content/asa/journal/poma/4/1/10.1121/1.3068630

[5] F. Rohner, "PANArt's offer to the metal sound sculptors." [Online]. Available: http://www.hang.ch/en/news/category/panarts-offerte-an-blechklangplastiker

[6] Pantheon Steel, "What is a Halo? What is a Genesis, Cirrus, and Stratus." [Online]. Available: http://www.pantheonsteel.com/FAQ.aspx#What_is_the_Halo__exactly_

[7] Zen Handpans, "Zen Handpans." [Online]. Available: http://zenhandpans.com

[8] C. Foulke, "CFoulke." [Online]. Available: http://www.cfoulke.com

[9] Saraz, "Saraz Handpans." [Online]. Available: http://sarazhandpans.com

[10] florianbetz, "HandPan.org - Pantam Questions about Name." [Online]. Available: http://www.handpan.org/forum/viewtopic.php?f=26&t=12382

[11] T. Rossing, A. Morrison, U. Hansen, F. Rohner, and S. Schärer, "ACOUSTICS OF THE HANG: A hand-played steel instrument," *ISMA 2007*.

[12] A. Achong, "The steelpan as system of non-linear mode-localized oscillators, I: Theory, simulations, experiments and bifurcations," *Journal of Sound and Vibration*, vol. 197, no. 4, pp. 471–487, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022460X9690543X

[13] U. Kronman, *Steel pan tuning: a handbook for steel pan making and tuning*, ser. Musikmuseets skrifter. Musikmuseet, 1992. [Online]. Available: http://books.google.co.uk/books?id=PGMJAQAAMAAJ

[14] J. Smith, "Physical Audio Signal Processing." [Online]. Available: https://ccrma.stanford.edu/~jos/pasp/

[15] J. Smith, "Using the Energy Decay Relief (EDR)." [Online]. Available: https://ccrma.stanford.edu/~jos/vguitar/Using_Energy_Decay_Relief.html

[16] J. Wells, "Reading the Sines: Sinusoidal Identification and Description using the Short Time Fourier Transform," *Music Technology Forum: Time-Frequency Analysis for Audio*, pp. 1–12, apr 2004.

[17] J. Beauchamp, "Time-variant spectra of violin tones," *The Journal of the Acoustical Society of America*, vol. 56, no. 3, pp. 995–1004, 1974.

[18] H. Suzuki, "Model analysis of a hammer-string interaction," *The Journal of the Acoustical Society of America*, 1987.

[19] N. Fletcher and T. Rossing, *The Physics of Musical Instruments*. Springer, 1998. [Online]. Available: http://books.google.co.uk/books?id=9CRSRYQlRLkC

[20] D. Martin, "Decay Rates of Piano Tones," *The Journal of the Acoustical Society of America*, 1947.

[21] MathWorks, "Matlab Documentation - polyfit." [Online]. Available: http://uk.mathworks.com/help/matlab/ref/polyfit.html

[22] J. Carrou, F. Gautier, and R. Badeau, "Theoretical and experimental investigations of harp's sympathetic modes," *19th International Congress on Acoustics*, Sep. 2007.

[23] F. Keiler, C. Karadogan, U. Zölzer, and A. Schneider, "Analysis of Transient Musical Sounds by Auto-Regressive Modeling," *Proc of 6th Int Conf on Digital Audio Effects (DAFx'03)*, 2003.

[24] U. Zölzer, *Dafx: Digital Audio Effects*. New York, NY, USA: John Wiley & Sons, Inc., 2002.

WEB AUDIO EVALUATION TOOL: A BROWSER-BASED LISTENING TEST ENVIRONMENT

Nicholas Jillings n.g.r.jillings@se14.qmul.ac.uk, **Brecht De Man** {b.deman, **David Moffat** d.j.moffat, **Joshua D. Reiss** joshua.reiss}@qmul.ac.uk
Centre for Digital Music, Queen Mary University of London

ABSTRACT

Perceptual evaluation tests where subjects assess certain qualities of different audio fragments are an integral part of audio and music research. These require specialised software, usually custom-made, to collect large amounts of data using meticulously designed interfaces with carefully formulated questions, and play back audio with rapid switching between different samples. New functionality in HTML5 included in the Web Audio API allows for increasingly powerful media applications in a platform independent environment. The advantage of a web application is easy deployment on any platform, without requiring any other application, enabling multiple tests to be easily conducted across locations. In this paper we propose a tool supporting a wide variety of easily configurable, multi-stimulus perceptual audio evaluation tests over the web with multiple test interfaces, pre- and post-test surveys, custom configuration, collection of test metrics and other features. Test design and setup doesn't require programming background, and results are gathered automatically using web friendly formats for easy storing of results on a server.

1. INTRODUCTION

Perceptual evaluation of audio plays an important role in a wide range of research on audio quality [1, 2], sound synthesis [3, 4], audio effect design [5], source separation [6, 7], music and emotion analysis [8, 9], and many others [10].

Table 1. Available audio perceptual evaluation tools

Name	Language	Ref.
APE	MATLAB	[11]
BeaqleJS	HTML5/JS	[12]
HULTI-GEN	Max	[13]
MUSHRAM	MATLAB	[6]
Scale	MATLAB	[14]
WhisPER	MATLAB	[15]

Copyright: ©2015 Nicholas Jillings 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.

Various listening test design tools are already available, see Table 1. A few other listening test tools, such as OPAQUE [16] and GuineaPig [17], are described but not available to the public at the time of writing.

Many are MATLAB-based, useful for easily processing and visualising the data produced by the listening tests, but requiring MATLAB to be installed to run or - in the case of an executable created with MATLAB - at least create the test. Furthermore, compatibility is usually limited across different versions of MATLAB. Similarly, Max requires little or no programming background but it is proprietary software as well, which is especially undesirable when tests need to be deployed at different sites. More recently, BeaqleJS [12] makes use of the HTML5 audio capabilities and comes with a number of predefined, established test interfaces such as ABX and MUSHRA [18]. BeaqleJS provides a number of similar features including saving of test data to a web server. The main difference is that with BeaqleJS, the configuration is done through writting a JavaScript file holding a JavaScript Object of the notation. Instead our presented system uses the XML document standard, which allows configuration outside of a web-centric editor. The results are also presented in XML again allowing 3rd party editors and programs to easily access. Finally, the presented system does not require web access to run, instead being deployed with a Python server script. This is particularly useful in studios where machines may not, by design, be web connected, or use in locations where web access is limited.

A browser-based perceptual evaluation tool for audio has a number of advantages. First of all, it doesn't need any other software than a browser, meaning deployment is very easy and cheap. As such, it can also run on a variety of devices and platforms. The test can be hosted on a central server with subjects all over the world, who can simply go to a webpage. This means that multiple participants can take the test simultaneously, potentially in their usual listening environment if this is beneficial for the test. Naturally, the constraints on the listening environment and other variables still need to be controlled if they are important to the experiment. Depending on the requirements a survey or a variety of tests preceding the experiment could establish whether remote participants and their environments are adequate for the experiment at hand.

The Web Audio API is a high-level JavaScript Application Programming Interface (API) designed for real-time processing of audio inside the browser through various pro-

cessing nodes [1]. Various web sites have used the Web Audio API for creative purposes, such as drum machines and score creation tools [2], others from the list show real-time captured audio processing such as room reverberation tools and a phase vocoder from the system microphone. The BBC Radiophonic Workshop shows effects used on famous TV shows such as Doctor Who, being simulated inside the browser [3]. Another example is the BBC R&D personalised compressor which applies a dynamic range compressor on a radio station that dynamically adjusts the compressor settings to match the listener's environment [19].

In contrast with the tools listed above, we aim to provide an environment in which a variety of multi-stimulus tests can be designed, with a wide range of configurability, while keeping setup and collecting results as straightforward as possible. For instance, the option to provide free-text comment fields allows for tests with individual vocabulary methods, as opposed to only allowing quantitative scales associated to a fixed set of descriptors. To make the tool accessible to a wide range of researchers, we aim to offer maximum functionality even to those with little or no programming background. The tool we present can set up a listening test without reading or adjusting any code, provided no new types of interfaces need to be created.

Specifically, we present a browser-based perceptual evaluation tool from which any kind of multiple stimulus audio evaluation tool where subjects need to rank, rate, select, or comment on different audio samples can be built. We also include an example of the multiple stimulus user interface included with the APE tool [11], which presents the subject with a number of axes on which a number of markers, corresponding to audio samples, can be moved to reflect any subjective quality, as well as corresponding comment boxes. However, other graphical user interfaces can be put on top of the engine that we provide with minimal or no modifications. Examples of this are the MUSHRA test [18], single or multiple stimulus evaluation with a two-dimensional interface (such as valence and arousal dimensions), or simple annotation (using free-form text, check boxes, radio buttons or drop-down menus) of one or more audio samples at a time. In some cases, such as method of adjustment, where the audio is processed by the user, or AB test, where the interface does not show all audio samples to be evaluated at once [20], the back end of the tool needs to be modified as well.

In the following sections, we describe the included interface in more detail, discuss the implementation, and cover considerations that were made in the design process of this tool.

2. INTERFACE

At this point, we have implemented the interface of the MATLAB-based APE (Audio Perceptual Evaluation) toolbox [11]. This shows one marker for each simultaneously evaluated audio fragment on one or more horizontal axes, that can be moved to rate or rank the respective fragments in terms of any subjective property, as well as a comment box for every marker, and any extra text boxes for extra comments. The reason for such an interface, where all stimuli are presented on a single rating axis (or multiple axes if multiple subjective qualities need to be evaluated), is that it urges the subject to consider the rating and/or ranking of the stimuli relative to one another, as opposed to comparing each individual stimulus to a given reference, as is the case with e.g. a MUSHRA test [18]. As such, it is ideal for any type of test where the goal is to carefully compare samples against each other, like perceptual evaluation of different mixes of music recordings [21] or sound synthesis models [4], as opposed to comparing results of source separation algorithms [6] or audio with lower data rate [18] to a high quality reference signal.

The markers on the slider at the top of the page are positioned randomly, to minimise the bias that may be introduced when the initial positions are near the beginning, end or middle of the slider. Another approach is to place the markers outside of the slider bar at first and have the subject drag them in, but the authors believe this doesn't encourage careful consideration and comparison of the different fragments as the implicit goal of the test becomes to audition and drag each fragment in just once, rather than to compare all fragments rigorously.

See Figure 1 for an example of the interface.

3. ARCHITECTURE

The tool uses entirely client side processing utilising the new HTML5 Web Audio API, supported by most major web browsers. The API allows for constructing audio processing elements and connecting them together to produce a high quality, real time signal process to manipulate audio streams. The API supports multichannel processing and has an accurate playback timer for precise, scheduled playback control. The API is controlled through the browser JavaScript engine and is therefore highly configurable. Processing is all performed in a low latency thread separate from the main JavaScript thread, so there is no blocking due to real time processing.

The web tool itself is split into several files to operate:

- `index.html`: The main index file to load the scripts, this is the file the browser must request to load.
- `core.js`: Contains global functions and object prototypes to define the audio playback engine, audio objects and loading media files
- `ape.js`: Parses setup files to create the interface as instructed, following the same style chain as the MATLAB APE Tool [11].

The HTML file loads the `core.js` file along with a few other ancillary files (such as the jQuery JavaScript extensions [4]), at which point the browser JavaScript begins to execute the on-page instructions, which gives the URL of the test setup XML document (outlined in Section 5). `core.js` parses this document and executes the functions in `ape.js` to build the web page. The reason for separating these two files is to allow for further interface

[1] http://webaudio.github.io/web-audio-api/
[2] http://webaudio.github.io/demo-list/
[3] http://webaudio.prototyping.bbc.co.uk/
[4] http://jquery.com/

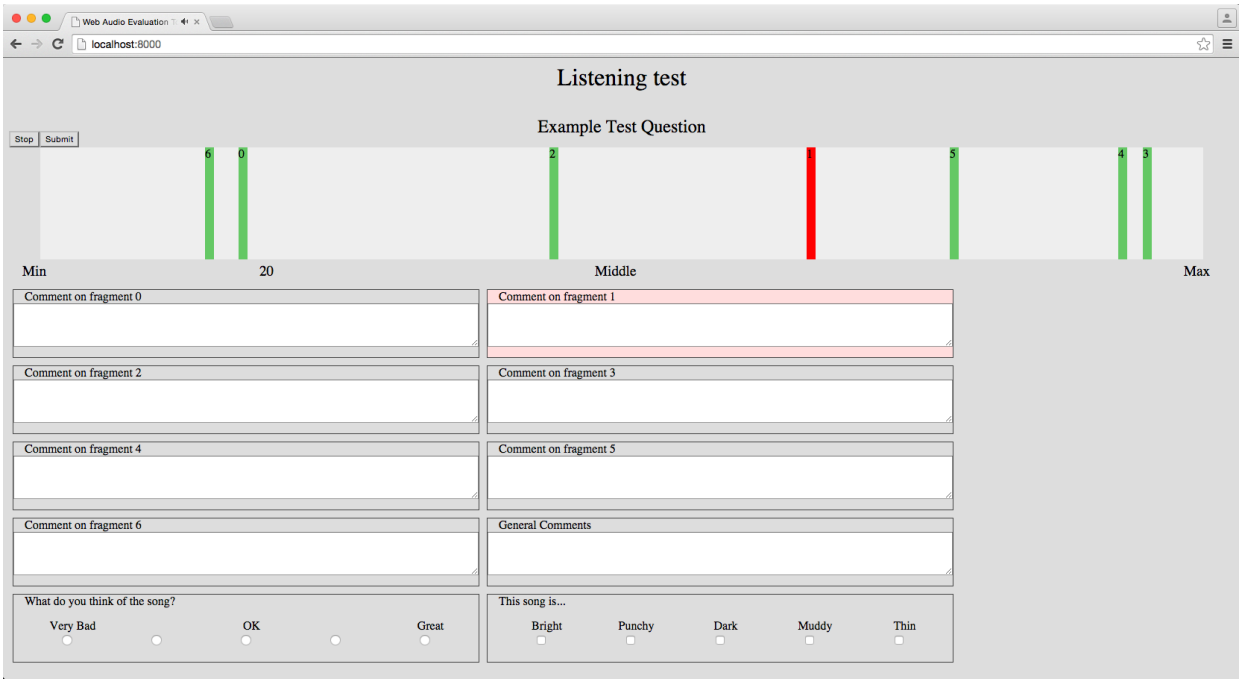


Figure 1. Example interface, with one axis, seven fragments, and text, radio button and check box style comments.

designs (such as MUSHRA [18] or 2D rating [20]) to be used, which would still require the same underlying core functions outlined in `core.js`.

The `ape.js` file has several main functions but the most important are documented here. *loadInterface(xmlDoc)* is called to decode the supplied project document in respect for the interface specified and define any global structures (such as the slider interface). It also identifies the number of pages in the test and randomises the order, if specified to do so. This is the only mandatory function in any of the interface files as this is called by `core.js` when the document is ready. `core.js` cannot 'see' any interface specific functions and therefore cannot assume any are available. Therefore *loadInterface(xmlDoc)* is essential to set up the entire test environment. Because the interface files are loaded by `core.js` and because the functions in `core.js` are global, the interface files can 'see' the `core.js` file and can therefore not only interact with it, but also modify it.

Each test page is loaded using *loadTest(id)* which performs two major tasks: to populate the interface with the slider elements and comment boxes; and secondly to instruct the *audioEngine* to load the audio fragments and construct the backend audio graph. *loadTest(id)* also instructs the audio engine in `core.js` to create the *audioObject*. These are custom audio nodes, one representing each audio element specified in each page. They consist of a *bufferSourceNode* (a node which holds a buffer of audio samples for playback) and a *gainNode*, both of which are Web Audio API Nodes. Various functions are applied, depending on which metrics are enabled, to record the interaction with the audio element. These nodes are then connected to the *audioEngine* (itself a custom web audio node) containing a *gainNode* (where the various *audioObjects* connect to) for summation before passing the output to the *destinationNode*, a permanent node of the Web Audio API created as the master output. Here, the browser then passes the audio information to the system.

When an *audioObject* is created, it is given the URL of the audio sample to load. This is downloaded into the browser asynchronously using the *XMLHttpRequest* object, which downloads any file into the JavaScript environment for further processing. This is particularly useful for the Web Audio API because it supports downloading of files in their binary form for decoding. Once downloaded the file is decoded using the Web Audio API offline decoder. This uses the browser available decoding schemes to decode the audio files into raw float32 arrays, which are in turn passed to the relevant *audioObject* for playback.

Once each page of the test is completed, identified by pressing the Submit button, the *pageXMLSave(testId)* is called to store all of the collected data until all pages of the test are completed. After the final test and any post-test questions are completed, the *interfaceXMLSave()* function is called. This function generates the final XML file for submission as outlined in Section 5.

4. SUPPORT AND LIMITATIONS

Different browsers support a different set of audio file formats and are not consistent in any format. Currently the Web Audio API is best supported in Chrome, Firefox, Opera and Safari. All of these support the use of the uncompressed WAV format. Although not a compact, web friendly format, most transport systems are of a high enough bandwidth this should not be a problem. Ogg Vorbis is another well supported format across the four supported major desktop browsers, as well as MP3 (although Firefox may not support all MP3 types [5]). One issue of the Web

[5] https://developer.mozilla.org/en-US/docs/Web/HTML/Supported_media_formats

Audio API is that the sample rate is assigned by the system sound device, rather than requested and does not have the ability to request a different one. As the sampling rate and the effect of resampling may be critical for some listening tests, the default operation when an audio file is loaded with a different sample rate to that of the system is to convert the sample rate. To provide a check for this, the desired sample rate can be supplied with the setup XML and checked against. If the sample rates do not match, a browser alert window is shown asking for the sample rate to be correctly adjusted. This happens before any loading or decoding of audio files so the browser will only be instructed to fetch files if the system sample rate meets the requirements, avoiding multiple requests for large files until they are actually needed.

5. INPUT AND RESULT FILES

The setup and result files both use the common XML document format to outline the various parameters. The setup file determines the interface to use, the location of audio files, the number of pages and other parameters to define the testing environment. Having one document to modify allows for quick manipulation in a 'human readable' form to create new tests, or adjust current ones, without needing to edit multiple web files. Furthermore, we also provide a simple web page to enter all these settings without needing to manipulate the raw XML. An example of such an XML document is presented below.

```
<?xml version="1.0" encoding="utf-8"?>
<BrowserEvalProjectDocument>
      <setup interface="APE" projectReturn="/save"
           randomiseOrder='false' collectMetrics='true'
           >
            <PreTest>
                  <question id="location" mandatory="
                       true">Please enter your location
                       .</question>
                  <number id="age" min="0">Please enter
                        your age</number>
            </PreTest>
            <PostTest>
                  <statement>Thank you for taking this
                       listening test!</statement>
            </PostTest>
            <Metric>
                  <metricEnable>testTimer</metricEnable
                       >
                  <metricEnable>elementTimer</
                       metricEnable>
                  <metricEnable>elementInitialPosition<
                       /metricEnable>
                  <metricEnable>elementTracker</
                       metricEnable>
                  <metricEnable>elementFlagListenedTo</
                       metricEnable>
                  <metricEnable>elementFlagMoved</
                       metricEnable>
            </Metric>
            <interface>
                  <anchor>20</anchor>
                  <reference>80</reference>
            </interface>
      </setup>
      <audioHolder id="test-0" hostURL="example_eval/"
           randomiseOrder='true'>
            <interface>
                  <title>Example Test Question</title>
                  <scale position="0">Min</scale>
                  <scale position="100">Max</scale>
                  <commentBoxPrefix>Comment on fragment
                       </commentBoxPrefix>
            </interface>
            <audioElements url="1.wav" id="elem1"/>
            <audioElements url="2.wav" id="elem2"/>
            <audioElements url="3.wav" id="elem3"/>
            <CommentQuestion id="generalExperience"
                 type="text">General Comments</
                 CommentQuestion>
            <PreTest/>
            <PostTest>
                  <question id="songGenre" mandatory="
                       true">Please enter the genre of
                       the song.</question>
            </PostTest>
      </audioHolder>
</BrowserEvalProjectDocument>
```

5.1 Setup and configurability

The setup document has several defined nodes and structure which are documented with the source code. For example, there is a section for general setup options where any pre-test and post-test questions and statements can be defined. Pre- and post-test dialogue boxes allow for comments or questions to be presented before or after the test, to convey listening test instructions, and gather information about the subject, listening environment, and overall experience of the test. In the example set up document above, a question box with the id 'location' is added, which is set to be mandatory to answer. The question is in the PreTest node meaning it will appear before any testing will begin. When the result for the entire test is shown, the response will appear in the PreTest node with the id 'location' allowing it to be found easily, provided the id values are meaningful.

We try to cater to a diverse audience with this toolbox, while ensuring it is simple, elegant and straightforward. To that end, we currently include the following options that can be easily switched on and off, by setting the value in the input XML file.

- **Snap to corresponding position**: When enabled and a fragment is playing, the playhead skips to the same position in the next fragment that is clicked. Otherwise, each fragment is played from the start.
- **Loop fragments**: Repeat current fragment when end is reached, until the 'Stop' or 'Submit' button is clicked.
- **Comments**: Displays a separate comment box for each fragment in the page.
- **General comment**: Create additional comment boxes to the fragment comment boxes, with a custom question and various input formats such as checkbox or radio.
- **Resampling**: When this is enabled, fragments are resampled to match the subject's system's sample rate (a default feature of the Web Audio API). When it is not, an error is shown when the system does not match the requested sample rate.
- **Randomise page order**: Randomises the order in which different 'pages' are presented.
- **Randomise fragment order**: Randomises the order and numbering of the markers and comment boxes corresponding to the fragments. Fragments are referenced to their given ID so referencing is possible (such as 'this is much brighter than fragment 4').
- **Require (full) playback**: Require that each fragment has been played at least once, partly or fully.
- **Require moving**: Require that each marker is moved (dragged) at least once.

- **Require comments**: Require the subject to write a comment for each fragment.
- **Repeat test**: Number of times each page in the test should be repeated (none by default), to allow familiarisation with the content and experiment, and to investigate consistency of user and variability due to familiarity. These are all gathered before shuffling the order so repeated tests are not back-to-back if possible.
- **Returning to previous pages**: Indicates whether it is possible to go back to a previous 'page' in the test.
- **Lowest rating below [value]**: To enforce a certain use of the rating scale, it can be required to rate at least one sample below a specified value.
- **Highest rating above [value]**: To enforce a certain use of the rating scale, it can be required to rate at least one sample above a specified value.
- **Reference**: Allows for a separate sample (outside of the axis) to be the 'reference', which the subject can play back during the test to help with the task at hand [18].
- **Hidden reference/anchor**: Whether or not an explicit 'reference' is provided, the 'hidden reference' should be rated above a certain value [18] - this can be enforced. Similarly, a 'hidden anchor' should be rated lower than a certain value [18].
- **Show scrub bar**: Display a playhead on a scrub bar to show the position in the current fragment.

When one of these options is not included in the setup file, they assume a default value. As a result, the input file can be kept very compact if default values suffice for the test.

5.2 Results

The results file is dynamically generated by the interface upon clicking the 'Submit' button. This also executes checks, depending on the setup file, to ensure that all fragments have been played back, rated and commented on. The XML output returned contains a node per fragment and contains both the corresponding marker's position and any comments written in the associated comment box. The rating returned is normalised to be a value between 0 and 1, normalising the pixel representation of different browser windows. The results also contain information collected by any defined pre/post questions. An excerpt of an output file is presented below detailing the data collected for a single audioElement.

```
<browserevaluationresult>
    <datetime>
        <date year="2015" month="5" day="28">
            2015/5/28</date>
        <time hour="13" minute="19" secs="17">
            13:19:17</time>
    </datetime>
    <pretest>
        <comment id="location">Control Room</
            comment>
    </pretest>
    <audioholder>
        <pretest></pretest>
        <posttest>
            <comment id="songGenre">Pop</comment>
        </posttest>
        <metric>
            <metricresult id="testTime">813.32</
                metricresult>
        </metric>
        <audioelement id="elem1">
            <comment>
                <question>Comment on fragment 1
                    </question>
                <response>Good, but vocals too
                    quiet.</response>
            </comment>
            <value>0.639010989010989</value>
            <metric>
                <metricresult id="elementTimer"
                    >111.05</metricresult>
                <metricresult id="
                    elementTrackerFull">
                    <timepos id="0">
                        <time>61.60</time>
                        <position>0.6390</
                            position>
                    </timepos>
                </metricresult>
                <metricresult id="
                    elementInitialPosition">
                    0.6571</metricresult>
                <metricresult id="
                    elementFlagListenedTo">
                    true</metricresult>
            </metric>
        </audioelement>
    </audioHolder>
</browserevaluationresult>
```

Each page of testing is returned with the results of the entire page included in the structure. One `audioelement` node is created per audio fragment per page, along with its ID. This includes several child nodes including the rating between 0 and 1, the comment, and any other collected metrics including how long the element was listened for, the initial position, and boolean flags showing if the element was listened to, moved and commented on. Furthermore, each user action (manipulation of any interface element, such as playback or moving a marker) can be logged along with a the corresponding time code. We also store session data such as the time the test took place and the duration of the test. We provide the option to store the results locally, and/or to have them sent to a server.

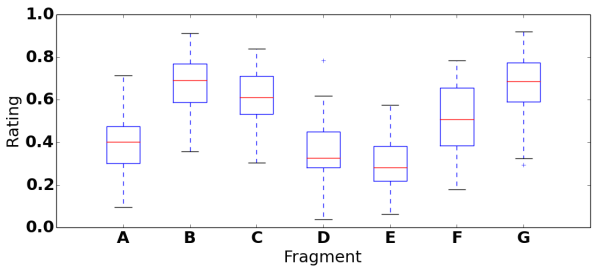


Figure 2. An example boxplot showing ratings by different subjects on fragments labeled 'A' through 'G'.

Python scripts are included to easily store ratings and comments in a CSV file, and to display graphs of numerical ratings (see Figure 2) or visualise the test's timeline. Visualisation of plots requires the free matplotlib library [6] .

6. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an approach to creating a browser-based listening test environment that can be used for a variety of types of perceptual evaluation of audio. Specifically, we discussed the use of the toolbox in the context of assessment of preference for different production practices, with identical source material. The purpose

[6] http://matplotlib.org

of this paper is to outline the design of this tool, to describe our implementation using basic HTML5 functionality, and to discuss design challenges and limitations of our approach. This tool differentiates itself from other perceptual audio tools by enabling web technologies for multiple participants to perform the test without the need for proprietary software such as MATLAB. The tool also allows for any interface to be built using HTML5 elements to create a variety of dynamic, multiple-stimulus listening test interfaces. It enables quick setup of simple tests with the ability to manage complex tests through a single file. Finally it uses the XML document format to store the results allowing for processing and analysis of results in various third party software such as MATLAB or Python.

Further work may include the development of other common test designs, such as MUSHRA [18], 2D valence and arousal/activity [9], and others. We will add functionality to assist with setting up large-scale tests with remote subjects, so this becomes straightforward and intuitive. In addition, we will keep on improving and expanding the tool, and highly welcome feedback and contributions from the community.

The source code of this tool can be found on `code.soundsoftware.ac.uk/projects/webaudioevaluationtool`.

7. REFERENCES

[1] M. Schoeffler and J. Herre, "About the impact of audio quality on overall listening experience," in *Proceedings of the 10th Sound and Music Computing Conference*, 2013, pp. 48–53.

[2] R. Repp, "Recording quality ratings by music professionals," in *Proceedings of the 2006 International Computer Music Conference*, 2006, pp. 468–474.

[3] A. de Götzen, E. Sikström, F. Grani, and S. Serafin, "Real, foley or synthetic? An evaluation of everyday walking sounds," in *Proceedings of SMC 2013 : 10th Sound and Music Computing Conference*, 2013.

[4] G. Durr, L. Peixoto, M. Souza, R. Tanoue, and J. D. Reiss, "Implementation and evaluation of dynamic level of audio detail," in *Audio Engineering Society Conference: 56th International Conference: Audio for Games*, 2015.

[5] B. De Man and J. D. Reiss, "Adaptive control of amplitude distortion effects," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, 2014.

[6] E. Vincent, M. G. Jafari, and M. D. Plumbley, "Preliminary guidelines for subjective evalutation of audio source separation algorithms," in *UK ICA Research Network Workshop*, 2006.

[7] J. D. Reiss and C. Uhle, "Determined source separation for microphone recordings using IIR filters," in *129th Convention of the Audio Engineering Society*, 2010.

[8] Y. Song, S. Dixon, M. T. Pearce, and G. Fazekas, "Using tags to select stimuli in the study of music and emotion," *Proceedings of the 3rd International Conference on Music & Emotion (ICME)*, 2013.

[9] T. Eerola, O. Lartillot, and P. Toiviainen, "Prediction of multidimensional emotional ratings in music from audio using multivariate regression models," in *Proceedings of the 10th International Society for Music Information Retrieval (ISMIR2009)*, 2009, pp. 621–626.

[10] A. Friberg and A. Hedblad, "A comparison of perceptual ratings and computed audio features," in *Proceedings of the 8th Sound and Music Computing Conference*, 2011, pp. 122–127.

[11] B. De Man and J. D. Reiss, "APE: Audio Perceptual Evaluation toolbox for MATLAB," in *136th Convention of the Audio Engineering Society*, 2014.

[12] S. Kraft and U. Zölzer, "BeaqleJS: HTML5 and JavaScript based framework for the subjective evaluation of audio quality," in *Linux Audio Conference, Karlsruhe, DE*, 2014.

[13] C. Gribben and H. Lee, "Toward the development of a universal listening test interface generator in Max," in *138th Convention of the Audio Engineering Society*, 2015.

[14] A. V. Giner, "Scale - a software tool for listening experiments," in *AIA/DAGA Conference on Acoustics, Merano (Italy)*, 2013.

[15] S. Ciba, A. Wlodarski, and H.-J. Maempel, "WhisPER – A new tool for performing listening tests," in *126th Convention of the Audio Engineering Society*, 2009.

[16] J. Berg, "OPAQUE – A tool for the elicitation and grading of audio quality attributes," in *118th Convention of the Audio Engineering Society*, 2005.

[17] J. Hynninen and N. Zacharov, "GuineaPig - A generic subjective test system for multichannel audio," in *106th Convention of the Audio Engineering Society*, 1999.

[18] *Method for the subjective assessment of intermediate quality level of coding systems*. Recommendation ITU-R BS.1534-1, 2003.

[19] A. Mason, N. Jillings, Z. Ma, J. D. Reiss, and F. Melchior, "Adaptive audio reproduction using personalized compression," in *Audio Engineering Society Conference: 57th International Conference: The Future of Audio Entertainment Technology – Cinema, Television and the Internet*, 2015.

[20] S. Bech and N. Zacharov, *Perceptual Audio Evaluation - Theory, Method and Application*. John Wiley & Sons, 2007.

[21] B. De Man, M. Boerum, B. Leonard, G. Massenburg, R. King, and J. D. Reiss, "Perceptual evaluation of music mixing practices," in *138th Convention of the Audio Engineering Society*, 2015.

AUTOMATIC SINGING VOICE TO MUSIC VIDEO GENERATION VIA MASHUP OF SINGING VIDEO CLIPS

Tatsunori Hirai
Waseda University
tatsunori_hirai@asagi.waseda.jp

Yukara Ikemiya
Kyoto University

Kazuyoshi Yoshii
Kyoto University

Tomoyasu Nakano
National Institute of Advanced Industrial Science and Technology (AIST)

Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST)

Shigeo Morishima
Waseda Research Institute for Science and Engineering / CREST, JST
shigeo@waseda.jp

ABSTRACT

This paper presents a system that takes audio signals of any song sung by a singer as the input and automatically generates a music video clip in which the singer appears to be actually singing the song. Although music video clips have gained the popularity in video streaming services, not all existing songs have corresponding video clips. Given a song sung by a singer, our system generates a singing video clip by reusing existing singing video clips featuring the singer. More specifically, the system retrieves short fragments of singing video clips that include singing voices similar to that in target song, and then concatenates these fragments using a technique of dynamic programming (DP). To achieve this, we propose a method to extract singing scenes from music video clips by combining vocal activity detection (VAD) with mouth aperture detection (MAD). The subjective experimental results demonstrate the effectiveness of our system.

1. INTRODUCTION

Many people consume music by not only listening to audio recordings, but also watching video clips via video streaming services (*e.g.*, YouTube[1]). Thus, the importance of music video clips has been increasing. Although a lot of music video clips have been created for promotional purposes, not all existing songs have their own video clips. If a video clip could be added to an arbitrary song, people could enjoy their favorite songs much more. Note that one of the most important parts of popular music is the vocal part. Thus, to enrich music listening experience, the automatic generation of "singing" video clips for arbitrary songs is a big challenge worth tackling.

Since there are a large number of music video clips available on the Web, these clips can be considered as an audio-visual dictionary covering almost all sound events. Given an audio clip, we could figure out what happens in a visual manner by searching for a video clip including similar sounds. The key idea in this paper is that we could make a music video clip for an arbitrary song by searching for video clips including singing voices that are acoustically similar to that in the target song.

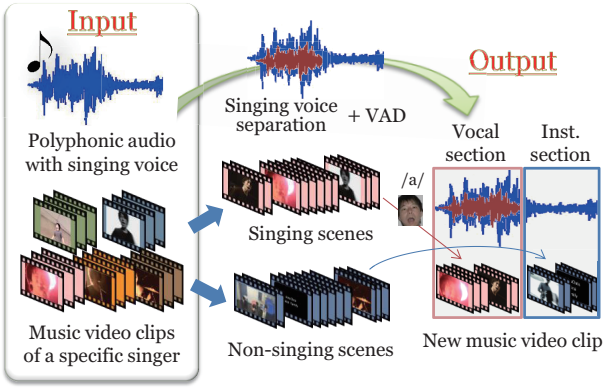


Figure 1. Conceptual image of our system.

To achieve automatic video generation, it is important to search for similar singing voices in a database of existing music video clips. If a similar singing voice can be found in an existing clip within the database, the singing actions of the singer in the clip can be expected to match the input singing voice. In this paper, we try to find multiple short video fragments that match the input singing voice and concatenate these fragments together to make a new singing video clip. As typical music video clips include a number of scene changes, the system output is allowed to contain frequent scene changes in output clips, as long as the singer remains unchanged.

Another solution to automatic singing video generation is to construct an audio-visual association model for lip sync. This approach requires clean video clips (e.g., simple background, stabled camera and target) recorded in an ideal environment for the precise analysis of audio-visual association. However, real video clips are noisy and are difficult to construct a reliable model. We aim to deal

Copyright: ©2015 Tatsunori Hirai 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] http://www.youtube.com/

with real video clips rather than video clips recorded in an ideal environment. It is extremely difficult to precisely detect objects in real video clips. Therefore, constructing a model from such clips is an unreasonable approach. We achieve automatic "singing voice to singing video" generation by focusing on singing voices and acoustic similarity between these voices. Fig. 1 shows a conceptual image of our system. Given an input song sung by an arbitrary singer and existing music video clips in which the singer appears, our system automatically generates both singing and non-singing scenes by mashing up[2] existing singing and non-singing scenes. This paper has two main contributions: audio-visual singing scene detection for music video clips, and singing video generation based on singing voice similarity and dynamic programming (DP).

2. RELATED WORK

The research topic of automatic music video generation has recently become popular, in fact a competition was held at the ACM International Conference on Multimedia 2012. Several methods have been proposed for the automatic generation of music video clips by focusing on *shallow* audio-visual association [1–4]. Foote *et al.* [1] proposed an audio-visual synchronization method based on audio novelty and video unsuitability obtained from camera motion and exposure. Hua *et al.* [2] proposed a system of automatic music video generation based on audio-visual structures obtained by temporal pattern analysis. Liao *et al.* [4] proposed a method to generate music video by extracting temporal features from the input clips and a piece of music, and casts the synthesis problem as an optimization. Although these methods consider audio-visual suitability [1], temporal patterns [2], or synchronization [4], higher-level information (*e.g.*, the semantics of video clips) was not taken into account. Nakano *et al.* [3] proposed a system called *DanceReProducer* which automatically generates dance video clips using existing dance video clips. For audio-visual association, the system uses an audio-to-visual regression method trained using a database of music video clips. Since they use low-level audio and visual features for regression, higher-level information (*e.g.*, dance choreography) was not taken into account.

In this paper, we tackle the problem of audio-visual synchronization between a singing voice and a singer's singing action (*e.g.*, lip motion). This enables more *semantic* synchronization than conventional methods. Yamamoto *et al.* [5] proposed a method that automatically synchronizes band sounds with video clips in which musical instruments are being played. Although this method can be considered as semantic synchronization, it does not mention synchronization of the vocal part, and requires manual input for sound source separation. In contrast, we focus on the vocal parts of a song and automate all processes.

In terms of synchronizing voice and lip motion, lip sync animation has been intensively studied in the field of computer graphics (CG). Various lip sync methods have been

[2] The term "mashup" refers to the mixture of multiple existing video clips.

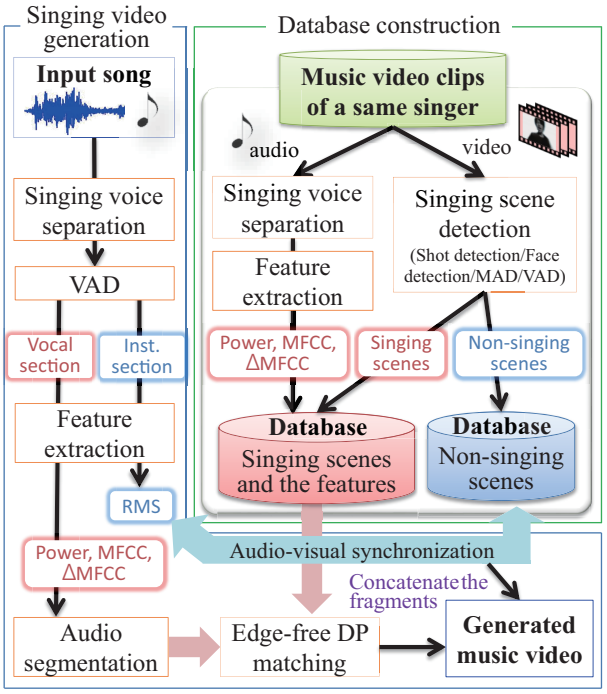


Figure 2. Overview of our system.

applied to 3DCG characters, including image-based photo-realistic human talking heads [6–9]. Basically, these talking heads are obtained by 3D face reconstruction or 3D face capture, methods that are not easily applicable to general video clips. Therefore, to make a video clip of a specific singer, users must prepare an ideal frontal face image or a sufficient amount of ideal video sequences of the singer. Our goal is to use the abundance of existing video clips so that users do not need to record new data to generate a video clip.

To mash up existing music video clips for synthesizing new singing video clips, we require a method to automatically detect singing scenes. Video event detection is a popular research topic in both the multimedia and pattern recognition communities. Many promising video analysis methods have been presented at the International Workshop on Video Retrieval called TRECVid [10]. The test data for the semantic indexing task in TRECVid include video clips with the label "Singing." To distinguish singing events from other events, most teams used acoustic features such as the mel frequency cepstral coefficient (MFCC) in addition to video features. These methods were designed for general-purpose event detection, not for the specific detection such as singing scene detection. To extract a target activity (*e.g.*, singing) from music video clips, we propose an automatic singing scene detection method that constructs a database of such scenes from existing music video clips.

3. SYSTEM IMPLEMENTATION

Our system consists of two processes: database construction and singing video generation. The system flow is shown in Fig. 2. The only data required by the system are the input song and music video clips for a database.

The database video clips must include singing scenes of the singer of the input song. A larger number of database clips will result in better output video quality.

To construct the database, singing scene detection will be applied to music video clips. Specifically, we employ an algorithm that combines vocal activity detection (VAD) from polyphonic musical signal and mouth aperture detection (MAD) based on facial recognition in a video clip. At the same time, singing voice separation is applied to the audio part of the database music video clips, and the singing voice feature is extracted. As a result, the singing scenes in the database clips and the singing voice and the features will be stored as a database for the system.

The singing video generation starts with singing voice separation and VAD. At this point, an input singing voice and the singer's singing scenes with the singing voices are available. For the vocal section of an input song, the system searches for acoustically similar singing voices from a database of singing scenes. For example, if part of the input singing phrase (query) is "oh," the system searches for a singing voice with a similar sound, such as the "o" from the word "over" or "old," on the basis of the similarity of singing voice features. Note that the system does not consider lyrics. It is difficult to find good matches between longer queries and the database. Therefore, the output singing video will be a mashup of small fragments of singing scenes. The length of each fragment will be automatically determined on the basis of the automatic singing voice segmentation. For the instrumental section of an input song, the system automatically adds the best synchronized video fragments. These are calculated on the basis of the matching between the accents of both the input song and the database clips. In this case, reference video scenes will be narrowed down to the non-singing scenes in the database clips. By mixing separately generated singing scenes and non-singing scenes, system generates a new music video clip. Further details of each process will be described in a later section.

4. SINGING SCENE DETECTION

We apply singing scene detection to music video clips for database construction. The singing scenes in a music video clip are one of the highlights of the clip. We define a singing scene as one in which a singer's mouth is moving, and the corresponding singing voice is audible. Therefore, not only audio analysis but also video analysis is necessary to detect such scenes. Our approach is to combine the existing VAD method [11] with a new MAD method that is customized for handling faces in a video clip by using continuity of video frames.

4.1 Vocal activity detection (VAD)

VAD is applied to the polyphonic audio signal of a music video clip. We apply the HMM-based VAD method proposed by Fujihara *et al.* [11]. This method trains models for both vocal and non-vocal states by GMM. Using HMM to express the audio signal as a transition of both states, the method detects vocal sections on the basis of probability.

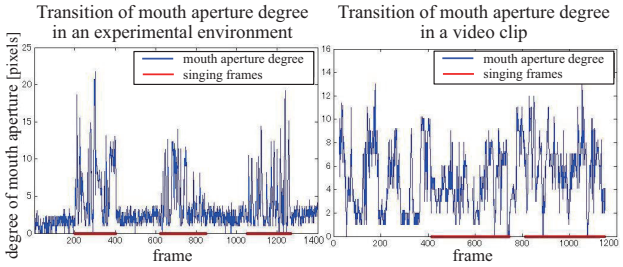


Figure 3. Transition of the mouth aperture degree in an experimental environment and a real video clip. Manually labeled singing scenes are shown in red.

4.2 Mouth aperture detection (MAD)

To detect mouth activity, we require a face recognition method for video clips. For facial detection, we apply the method proposed by Irie *et al.* [12]. This method uses a global fitting of the active structure appearance model (ASAM) to find face areas and a local fitting model to detect each facial part and facial feature points. Therefore, the feature points of the mouth can be detected.

This face detection method is comparatively robust to facial expressions and transitions in the direction of the face. However, the detection of facial feature points in a music video clip is difficult, as there is considerable noise in the detected results. Fig. 3 illustrates the appearance of noise in a real music video clip by comparing with the mouth aperture degree in a video recorded in an experimental environment. Hence, we cannot directly use the mouth aperture degree based on the mouth feature points to detect singing scenes in a real video clip. Instead of directly using the mouth aperture degree, we use the standard deviation of the distance between the upper and lower lip in each consecutive sequence of the same person's face as a mouth feature.

To acquire suitably consecutive face sequences, the system detects shot boundaries of a video clip. A shot is a consecutive video sequence that has no scene changes or camera switches. According to the continuity of video frames, a person in one shot is always the same when there is no movement of the person or camera. However, even if there is a movement of the person or camera, the same person can be captured by tracking their movement. To detect the boundary of a shot, the system subtracts consecutive luminance histograms, and uses their summation as a shot detection feature. When the value of the feature is higher than other values, the frame is considered to be a shot boundary. It is possible to detect the same person's face by tracking the spatial trajectory of the detected face across one shot. If the mouth feature is greater than a threshold[3] , the system classifies the shot with the entire consecutive face sequence as a singing scene.

[3] The value is experimentally fixed to 10 when the face size is normalized to 512×512 pixels.

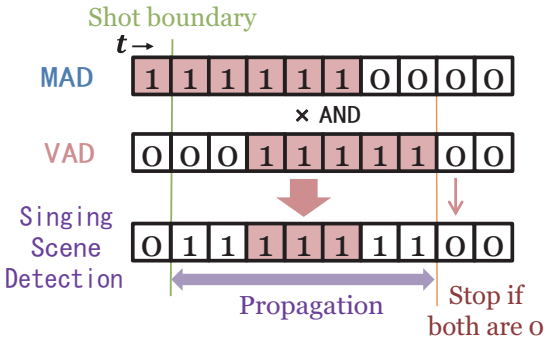


Figure 4. Combination of detected results.

	Precision	Recall	F score
VAD	0.632	0.732	0.672
MAD	0.609	**0.823**	0.677
VAD+MAD	**0.662**	0.759	**0.690**

Table 1. The accuracy of singing scene detection.

4.3 Combination of VAD and MAD

Our singing scene detection method combines the results of both VAD and MAD. Fig. 4 illustrates our method of combining the results. Both VAD and MAD results can be expressed in binary (1 denotes singing, and 0 denotes not singing). By taking the logical AND of both results, we can classify singing scenes with high reliability, however, only part of them can be detected. To detect more singing scenes, we use the continuity of video frames. The method propagates the results to consecutive frames until both the VAD and MAD results of the frame are 0.

To evaluate the accuracy of our singing scene detection method, we performed an experiment to detect singing scenes in real music video clips. We manually add annotations to 10 music video clips of professional musicians, and compared the detection accuracy with VAD and MAD. Table 1 lists the average accuracy of singing scene detection with each method. These results show that our combined algorithm gave the best performance in terms of accuracy (F score).

5. SINGING VIDEO GENERATION

To generate singing video, users prepare an arbitrary input song for singing video generation and the same singer's existing music video clips for database construction. Here, singing scenes should be included in database clips. The system calculates the similarities between the input singing voice and the database singing voices to search for well-synchronized singing sequences. Previous studies on talking heads ([13], [14]) have used both audio and visual features to construct an audio-visual association model from which a talking head is generated. In contrast, we use audio information alone to retrieve video fragments. This is because the extraction accuracy of visual features, especially the mouth feature, is not high enough to construct an audio-visual model from real music video clips. Most talking head research uses video captured in an experimental environment which is different from real music video clips. Therefore, we do not use visual features in generation part but focus on an acoustic similarity of singing voices.

5.1 Database construction

Our system constructs a database from the user-prepared music video clips. The database includes the singing scenes from the clips and the singing voice features extracted from the audio. The singing scenes are extracted with our singing scene detection method, and singing voice separation is performed on the audio part of the clips.

5.1.1 Singing voice separation

To extract singing voice features, a singing voice separation method is required. We apply the singing voice separation method proposed by Ikemiya *et al.* [15], which achieved the best separation performance in the Music Information Retrieval Evaluation eXchange (MIREX2014), a singing voice separation task.

This method uses a robust principal component analysis (RPCA) to separate non-repeating components, such as a singing voice, from a polyphonic spectrogram. By estimating the F0 contour from the separated components including the singing voice, we can obtain a binary mask that passes only the harmonic partials of the F0 contours. This method further improves the singing voice separation accuracy by combining the binary masks obtained using RPCA and F0 harmonics.

5.1.2 Singing voice feature extraction

From a separated singing voice, the system extracts the singing voice features that represent the characteristics of a singer's voice and prosody. Our goal is to generate a singing video that is well-synchronized to an arbitrary input song. The lyrics are an important factor in synchronizing a singing voice and a singing video, especially with regard to the motion of the mouth. However, it is difficult to obtain lyrics that are aligned with the audio signal for all existing songs. Therefore, we employ the MFCC which is related to prosody, and the power which is related to the dynamics of singing voice, as the singing voice feature.

The system extracts the 12 dimensional MFCC (excluding zeroth order which corresponds to the power), ΔMFCC, and the one dimensional power of the audio signal from the singing voice (25 dimensions in total).To realize better lip-sync, we handle power feature separated from MFCC features. The length of audio analysis frame and the analysis time step is 1/29.97 seconds in order to synchronize audio and video analysis time step. At this point, the singing voice feature values are normalized to have a mean value of 0 and a variance of 1.

Thus, singing scenes from the user-prepared music video clips and the singing voice features can be stored in the database.

5.2 Video fragment retrieval

From the input song sung by the same singer as database clips, the system retrieves singing video fragments that synchronize well with the input song. Because the input and database clip are of the same singer, we expect the mouth shape to be alike for similar singing voice features. To search for singing video fragments with similar singing voices, we extract the same 25 dimensional singing voice features from the singing voice of the input song as in the database. After the feature extraction, VAD is carried out to determine the section to which the singing video should be added.

Because our system will be used to generate new music video clips that do not exist in the database, there is no chance of finding a video clip with perfect synchronization. Therefore, we search for small fragments of singing video clips, and concatenate these fragments to achieve good synchronization. The intensity of synchronization is a trade-off between the temporal consistencies in the output clip. As we imagine, the output clip will be visually inconsistent, because it is a mixture of multiple video clips. However, many music video clips consist of more than one scene and the occurrence of frequent scene changes is not unusual in case of music video clips.

The length of each fragment is automatically determined by the singing voice segmentation. By manually specifying the minimum and maximum fragment lengths, the system automatically finds segmentation boundaries of an input singing voice on the basis of the power of the singing voice. This segmentation is performed by searching for the minimum power point within a user-specified range. Thus, singing voice can be segmented based on the phrases.

To retrieve the best-synchronized singing scene fragments, we employ edge-free DP. Though the normal DP matching require two sequences to be the same length, edge-free DP searches for the shortest path with arbitrary length. We fixed the length of the input singing voice feature and made the length of database features variable with edge-free DP. The Euclidean distance between the input and database singing voice features is used as the cost of DP. A weight is added to the cost in order to change the priority of the MFCC and the power. Adding larger weight to the power realizes better synchronization in terms of timing. Whereas the MFCC features affect prosody (shape of mouth), the power feature affects the onset and offset (timing) of a voice which is important in lip syncing. We assign a weight of 0.2-0.5 (20-50%) to the power feature, and spread the rest across the MFCC features (sum of the weight will be 1.0 in total). The system searches for a database singing scene with the minimum cost for each input singing voice fragment by shifting the DP start point in the database clips frame by frame. Edge-free DP makes it possible to adopt the end point of DP on the basis of the cost, so that the start point is fixed but the end point of DP depends on the cost. By concatenating all the retrieved fragments, the system generates the singing video output. Although we do not consider the mouth shapes of a singer, the mouth shape in a retrieved fragment tends to correspond to the phoneme of the input singing voice.

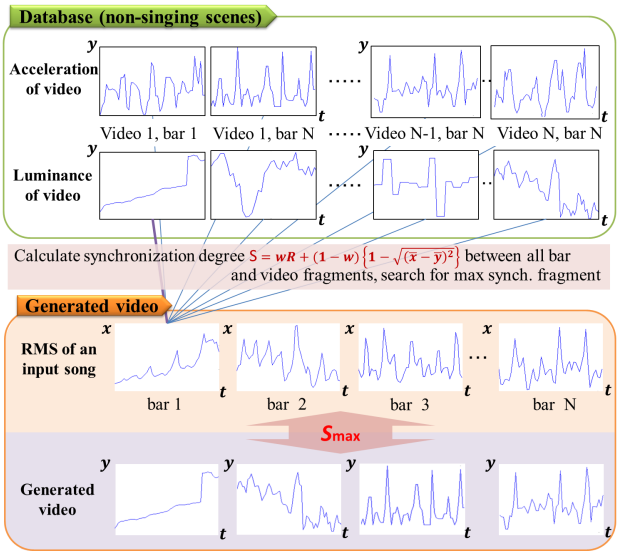


Figure 5. Generation of non-singing scenes.

5.3 Non-singing scene generation

The generation of singing scenes alone is not sufficient for music video generation. We therefore implement an automatic music video generation method for non-singing scenes to complement that for singing scenes. We employ the metric proposed by Hirai *et al.* [16], which considers the synchronization between the music and the video based on a subjective evaluation. This synchronization method considers accents in the video, such as the acceleration or transition in luminance, and the audio accent of the input music, which is calculated by the root mean square (RMS) of the audio signal. By applying this method to instrumental sections detected by VAD, non-singing scenes can be generated. Here, the database for this part only contains non-singing scenes from the database video clips, and the length of each fragment is fixed to one musical bar of the input song. Here, the length of one musical bar is extracted based on the same method as Hirai *et al.* [16].

Fig. 5 shows how these non-singing scenes are added to the instrumental sections of the input song. The synchronization degree S represents a measure for calculating the synchronization, as proposed in [16].

5.4 Mixing singing and non-singing scenes

To generate the final output music video clip, we mix the singing and non-singing scenes. However, the generation of singing scenes relies on the detection of singing scenes in database clips and the separation of the singing voice, which are not perfect. Thus, some mistakes may occur in these processes. For example, the system may detect an instrumental section as a vocal section in the VAD process, leading to the generation of singing scenes for instrumental sections. Therefore, we only use reliable singing scenes.

To improve the output clip, the system replaces unreliable singing scenes with non-singing scenes. This reliability may be either the DP cost, which represents the synchronization degree, or the VAD likelihood, which represents the reliability of VAD. Because there is no rule that

singing scenes must be accompanied by a vocal section in a music video clip, we can instead use better synchronized fragments. This process makes it possible to generate only well-synchronized singing scenes, and replace other scenes with acceptable (non-singing) scenes.

6. EVALUATION

We performed a subjective evaluation experiment to compare the generated results with music video produced by another method. Twenty subjects were asked to watch the music video clips automatically generated by our proposed method and the comparison method by Hirai *et al.*([16]), and to determine which was better in terms of audio-visual synchronization. All subjects are not the professional of music video editing. Since we used the comparison method to generate non-singing scenes, the generated video clips using the same song are the exactly same in non-singing scenes. The differences between the clips generated by each method are therefore in the singing scenes. As input, we used five songs by one singer, and constructed the database from clips of this singer's music video clips. The subjects watched a total of ten 30-second video clips, and scored them from 1 (Hirai *et al.* 2012 is better) to 5 (proposed method is better) by comparing the clips generated with each method in an aspect of audio-visual synchronization. The clips show the beginning of the songs, and all include an instrumental section.

Fig. 6 shows the evaluation score for each song. The baseline is 3.00, and higher scores indicate that proposed method is better than the comparison method. The evaluation score for each song is the average of all subjects' evaluation scores, and the average of the five songs is 3.66. Seventeen out of twenty subjects pointed out that the synchronization between singing voice and the singer's mouth is one of the factors for the audio-visual synchronization. From this result, we can say that our method is better with regard to singing scene generation. However, some subjects mentioned that the frequent scene changes and unnatural scene transitions were distracting. It is difficult to evaluate music video clips because people focus on many factors within a single clip. Taking this into account, the evaluation results suggest that our method exhibits respectable audio-visual synchronization.

7. CONCLUSION

This paper presented a system that can automatically generate a "singing" video clip for an arbitrary song. The main contribution of this work is our proposal for an automatic generation method that employs singing voice similarity and edge free DP to semantically synchronize singing scenes from existing video clips with a singing voice. Our automatic detection technique for singing scenes in music video clips, especially the MAD technique based on the standard deviation of the consecutive face sequences, represents another contribution.

Our future work is to handle inter-singer generation. The current singing voice features limit our method, as the

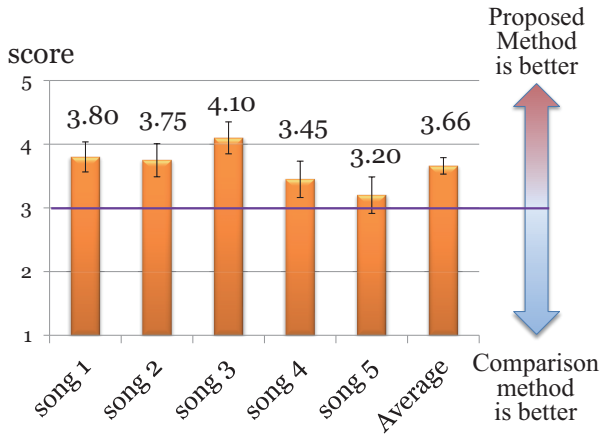


Figure 6. Subjective evaluation results comparing proposed method and comparison method by Hirai *et al.* 2012 [16].

singer's voice of the database video clips must sound similar to the voice of the input song, because the feature values tend to change with each individual. Initial tests suggest that inter-singer generation is feasible between two singers with a similar voice, but this is not the case with voices that are not similar. These problems may be solved by applying a voice conversion technique. In the future, we aim to generate a music video clip in which an arbitrary singer sings an arbitrary song. It might be possible to create a video clip in which a deceased singer sings a new song, which we can sometimes see in a film concert.

Our future work will also include semantic audio-visual synchronization that is not limited to a singing voice, but instead considers other audio-visual objects (events), using existing video clips. Because we have access to large numbers of video clips, the style of watching might change if we can use them to create new experiences. Extending our framework, it might be possible to add video for unknown sounds to help us understand their sound source. We are investigating how music video clips can be reconstructed using existing clips, and are on the way to achieving this. The time may come when people will enjoy automatically generated digital content as well as human created content.

Acknowledgments

This work was supported by OngaCREST, CREST, JST and partially supported by JSPS Grant-in-Aid for JSPS Fellows.

8. REFERENCES

[1] J. Foote, M. Cooper, and A. Girgensohn, "Creating music videos using automatic media analysis," in *Proc. ACMMM*, 2002, pp. 553–560.

[2] X.-S. Hua, L. Lu, and H.-J. Zhang, "Automatic music video generation based on temporal pattern analysis," in *Proc. ACMMM*, 2004, pp. 472–475.

[3] T. Nakano, S. Murofushi, M. Goto, and S. Mor-

ishima, "DanceReProducer: An automatic mashup music video generation system by reusing video clips on the web," in *Proc. SMC*, 2011, pp. 183–189.

[4] Z. Liao, Y. Yu, B. Gong, and L. Cheng, "audeosynth: Music-driven video montage," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 34, no. 4, 2015.

[5] T. Yamamoto, M. Okabe, Y. Hijikata, and R. Onai, "Semi-automatic synthesis of videos of performers appearing to play user-specified music," in *Proc. WSCG*, 2013, pp. 179–186.

[6] C. Bregler, M. Covell, and M. Slaney, "Video rewrite: Driving visual speech with audio," in *Proc. SIGGRAPH*, 1997, pp. 353–360.

[7] S. Kawamoto *et al.*, "Galatea: Open-source software for developing anthropomorphic spoken dialog agents," in *Life-Like Characters*, ser. Cognitive Technologies. Springer Berlin Heidelberg, 2004, pp. 187–211.

[8] R. Anderson, B. Stenger, V. Wan, and R. Cipolla, "Expressive visual text-to-speech using active appearance models," in *Proc. CVPR*, 2013, pp. 3382–3389.

[9] L. Wang and F. Soong, "Hmm trajectory-guided sample selection for photo-realistic talking head," *Multimedia Tools and Applications*, pp. 1–21, 2014.

[10] A. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *Proc. MIR*, 2006, pp. 321–330.

[11] H. Fujihara, M. Goto, J. Ogata, and H. Okuno, "Lyricsynchronizer: Automatic synchronization system between musical audio signals and lyrics," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 6, pp. 1252–1261, 2011.

[12] A. Irie, M. Takagiwa, K. Moriyama, and T. Yamashita, "Improvements to facial contour detection by hierarchical fitting and regression," in *Proc. ACPR*, 2011, pp. 273–277.

[13] B.Theobald and N. Wilkinson, "On evaluating synthesised visual speech," in *Proc. Interspeech*, 2008, pp. 2310–2313.

[14] S. Deena, S. Hou, and A. Galata, "Visual speech synthesis using a variable-order switching shared gaussian process dynamical model," *Multimedia, IEEE Transactions on*, vol. 15, no. 8, pp. 1755–1768, 2013.

[15] Y. Ikemiya, K. Yoshii, and K. Itoyama, "Singing voice analysis and editing based on mutually dependent f0 estimation and source separation," in *Proc. ICASSP*, 2015.

[16] T. Hirai, H. Ohya, and S. Morishima, "Automatic mash up music video generation system by perceptual synchronization of music and video features," in *Proc. SIGGRAPH (poster)*, 2012.

RENDERING AND SUBJECTIVE EVALUATION OF REAL VS. SYNTHETIC VIBROTACTILE CUES ON A DIGITAL PIANO KEYBOARD

Federico Fontana
Università di Udine
Dept. of Mathematics and Computer Science
via delle Scienze 206, Udine 33100, Italy
federico.fontana@uniud.it

Federico Avanzini
Università di Padova
Dept. of Information Engineering
Via G. Gradenigo 6/b, Padova 35131, Italy
avanzini@dei.unipd.it

Hanna Järveläinen, Stefano Papetti
Zürcher Hochschule der Künste
Inst. for Computer Music and Sound Technology
Pfingstweidstrasse 96, Zurich 8031, Switzerland
name.surname@zhdk.ch

Giorgio Klauer, Lorenzo Malavolta
Conservatorio di Musica "Cesare Pollini"
Via Eremitani 18, Padova 35121, Italy
name.surname@gmail.com

ABSTRACT

The perceived properties of a digital piano keyboard were studied in two experiments involving different types of vibrotactile cues in connection with sonic feedback. The first experiment implemented a free playing task in which subjects had to rate the perceived quality of the instrument according to five attributes: Dynamic control, Richness, Engagement, Naturalness, and General preference. The second experiment measured performance in timing and dynamic control in a scale playing task. While the vibrating condition was preferred over the standard non-vibrating setup in terms of perceived quality, no significant differences were observed in timing and dynamics accuracy. Overall, these results must be considered preliminary to an extension of the experiment involving repeated measurements with more subjects.

1. INTRODUCTION

Research on musical haptic perception is constantly growing, aiming at connecting measurable effects originated by tactile properties of an instrument to subjective preference judgments and, ultimately, to the musician's experience and specific aspects of his or her performance. Such research considers both traditional instruments such as pianos and violins [1, 2], and extends to augmentations spanning the broader area of new instrument design with applications to musical interaction and education [3, 4].

Specifically concerning the piano, the reproduction of the tactile properties of the keyboard has been first approached from a kinematic perspective with the aim of reproducing the mechanical response of the keys [5, 6], also in light of experiments emphasizing the sensitivity of pianists to the keyboard mechanics [7]. Only recently, and in parallel to industrial outcomes [8], did researchers start to analyze the role of vibrotactile feedback as a potential conveyor of salient cues: an early attempt claimed possible qualitative relevance of these cues [9]. Later along the same line, ground for a substantial step forward was set when some of the present authors not only found significant sensitivity to such cues [10], but also hypothesized that pianists are sensitive to key vibrations also when their amplitude is below the standard subjective thresholds originally estimated by stimulating subjects' fingertips with purely sinusoidal stimuli [11].

This conclusion gives rise to an interesting discussion, since it contradicts previous experiments [1] only apparently. Indisputably, those experiments did not take into account the complex perceptual effects due to vibrotactile temporal, spatial and spectral summations resulting from playing single or multiple keys. More importantly they did not address the issue of interactivity, reflecting an inherent lack of (also non-musical) studies addressing vibrotactile perception under active touch. Especially for this reason, authors of this paper have recently studied vibrotactile sensitivity measured under this condition, obtaining thresholds that are significantly lower than what reported in the previous literature [12].

In light of such unexpected differences found in the pianists' sensitivity thresholds, this work focuses on the ability of subjects to make a distinction in perceived *quality* between different types of vibrotactile feedback. In other words, we hypothesize that pianists appreciate the reproduction of real as opposed to simplified synthetic key vibrations. The experiment required to disassemble a digital piano keyboard, and instrument it so as to convey vibratory signals to the user; then, to record key vibrations on an acoustic piano and to synthesize simplified counterparts, which were organized in two respective sample banks.

Two experiments were planned making use of this setup: One studying subjective quality perception, and one measuring timing and dynamic performance. Results show that the setups augmented with vibrations were generally preferred over the non-vibrating standards, with a slight pref-

Copyright: ©2015 Federico Fontana 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. Experimental setup.



Figure 2. One of the transducers used to convey vibration at the keyboard.

erence towards amplified vibrations as opposed to vibrations of realistic amplitude. On the other hand, no effect was observed on timing or dynamics accuracy in the performance experiment, suggesting either that the difference is more of a subjective nature, or that vibrotactile feedback is not relevant for the specific performative task considered here. However, in the present pilot experiment, low concordance was observed between subjects, which suggests that intra- and inter-individual consistency is likely an important issue. As a future task, the experimental design needs some revision until more significant conclusions can be claimed.

2. SETUP

The keyboard of a Viscount Galileo VP-91 digital piano was detached from its metal casing, containing also the electric and electronic hardware, and then screwed to a thick plywood board (see Fig. 1).

Two Clark Synthesis TST239 Silver Tactile Transducers were attached to the bottom of the wooden board as shown in Fig. 2, respectively in correspondence of the lower and middle octaves, in this way enabling to convey vibrations at the most relevant areas of the keyboard [10]. Once equipped in this way, the keyboard was laid on a X-shaped keyboard stand, interposing foam rubber at the contact points.

The transducers were driven by a Yamaha P2700 amplifier in dual mono configuration, fed with a monophonic signal. The input was provided by a RME Fireface 800 audio interface communicating with an Apple MacBook Pro via Firewire. Sound and vibrotactile feedback were generated via software using Reaper 4 digital audio workstation, [1] which hosted the following plug-ins: the Pianoteq 4.5 physical modelling piano was used to synthesize audio feedback, delivered to the performer via headphones; the Native Instruments Kontakt 5 sampler [2] in series with MeldaProduction MEqualizer parametric equalizer [3] were used for vibration playback. The piano synthesizer was configured to match the sound of the grand piano used for recording vibration samples, as described below.

A schematic of the setup is shown in Fig. 3. The computer was also used to conduct the tests and collect experimental data. For this, programs were implemented as patches for the Pure Data real-time environment. [4] More details are given below in the description of each experiment.

2.1 Spectral equalization

Even if the setup was assembled in a way to avoid resonances due to nonlinearities, evidently the vibratory frequency response of the keyboard-plywood board system was not flat. Additionally, the transducers exhibit a prominent notch around 300 Hz. The overall frequency response of the transduction-transmission chain was measured in correspondence of all the A keys and led to an average magnitude spectrum that, once inverted, provided the spectral flattening equalization characteristics shown in Fig. 4. It can be noticed that the 300 Hz notch of the transducers is compensated along with resonances and anti-resonances of the mechanical system.

To avoid the generation of resonance peaks along the keyboard, we approximated this characteristics using the parametric equalizer plug-in, namely with a shelving filter providing a ramp climbing by 18 dB in the range [100–600] Hz, in series with a 2nd-order filter block approximating the peak around 180 Hz.

2.2 Vibration signals

Real vibration recordings were acquired at the keyboard of a Yamaha DC3 M4 Disklavier, using a Wilcoxon Research 736 piezoelectric accelerometer and iT100M Intelligent Transmitter connected to the audio interface. By triggering each of the 88 actuated keys of the Disklavier via MIDI control, vibration samples were recorded on every key at velocities 12, 23, 34, 45, 56, 67, 78, 89, 100,

[1] www.reaper.fm
[2] www.native-instruments.com
[3] www.meldaproduction.com
[4] puredata.info

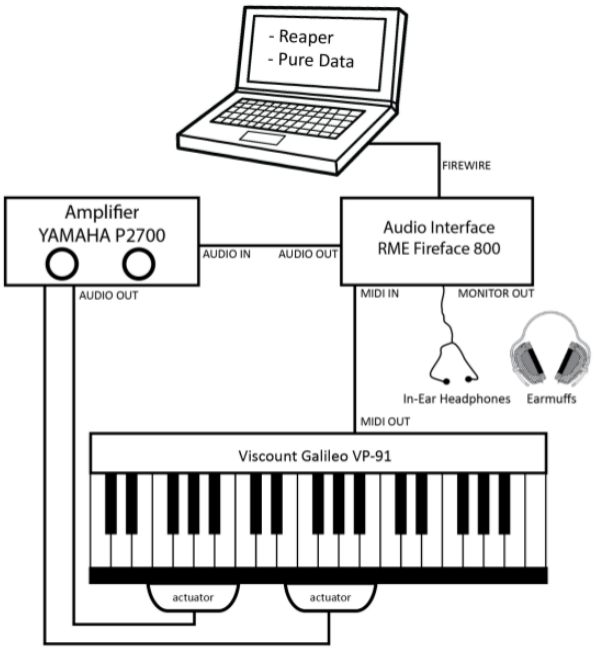


Figure 3. Schematic of the setup.

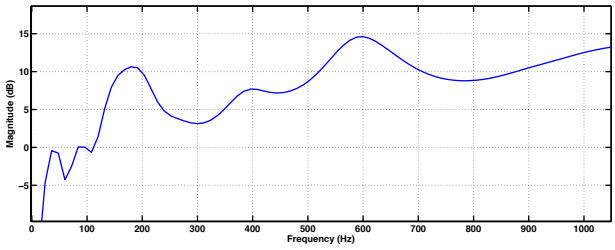


Figure 4. Spectral flattening: average equalization curve.

111. The accelerometer was secured to each measured key with Pongo. [5]

In addition to these recorded samples, a second bank of vibration signals was synthetically generated, with the purpose of reproducing the same amplitude envelope of the real signals while changing the spectral content ony. To this end, synthetic signals for each key and each velocity value were constructed as follows. First, white noise was was generated and then bandlimited in the range [20–500] Hz (corresponding to the vibrotactile bandwidth [11]). Then the noise was passed through a 2nd-order resonant filter centered at the fundamental frequency of the key. The resulting signal was modulated by the amplitude envelope of the corresponding recorded vibration sample, which in turn was estimated from the energy decay curve of the sample via the Schroeder integral [13]. Finally, the energy of the synthetic sample was equalized to that of the corresponding real sample.

The two sets of recorded and synthetic vibration samples were loaded into two distinct instances of the sampler plug-in, which managed their interpolation across velocities, based on the messages of MIDI note and key velocity coming from the digital keyboard.

[5] http://www.fila.it/en/pongo/history/

2.3 Key velocity calibration

The keys of the Disklavier and the Galileo digital piano have different response dynamics because of their mechanics and mass. Since pianists adapt their style in consequence of these differences, the digital keyboard had to be subjectively calibrated aiming at equalizing its dynamics with that of the Disklavier.

The keyboard response was set using the velocity calibration routine included in Pianoteq, which was performed by an experienced pianist first on the Disklavier and then on the digital keyboard. As expected, two fairly different velocity maps were obtained. Then, by making use of a MIDI filter plug-in in Reaper, each point of the digital keyboard velocity map was projected onto the corresponding point of the Disklavier velocity map. The resulting key velocity transfer characteristics was then independently checked by two more pianists, to validate its reliability and neutrality. In this way we ensured that when a pianist played the digital keyboard at a desired dynamics, the corresponding vibration samples recorded on the Disklavier would be triggered.

2.4 Loudness matching

As a final calibration step, the loudness of the piano synthesizer at the performer's ear was matched to that of the Disklavier grand piano. In order to do this, the sound produced by the A keys of the Disklavier at various velocities was recorded using a KEMAR mannequin positioned at the pianists location [10].

Then, additional measurements were taken with the KEMAR mannequin now wearing the same equipment that would be later used by experimental subjects, i.e. a pair of Sennheiser CX 300-II earphones and, on top of them, a pair of 3M Peltor X5 ear-muffs. In this case, A notes generated at the corresponding velocities by the Pianoteq engine were played back. Finally, the loudness of the piano synthesizer was matched to that of the Disklavier, by using the volume mapping feature of Pianoteq, which allows one to set independently the volume of each key across the keyboard.

3. EXPERIMENTS

Eleven subjects participated in the experiment, five females and six males. Their average age was 26 years, and their average piano playing experience was 8 years after reaching conservatory level. Two of the subjects were jazz pianists, the rest played classical piano. All of them signed an informed consent form. A session including the two experiments lasted about one hour.

Audio-tactile stimuli were produced at runtime: the digital keyboard played by the participants sent MIDI messages to the computer, where the piano synthesizer plug-in generated the related sounds and, in parallel, the sampler plug-in played back the corresponding vibration samples then processed by the equalizer plug-in (see again Fig. 3).

Subjects wore earphones and ear-muffs on top of them, in the same fashion as the KEMAR mannequin did during the loudness matching procedure described above. In

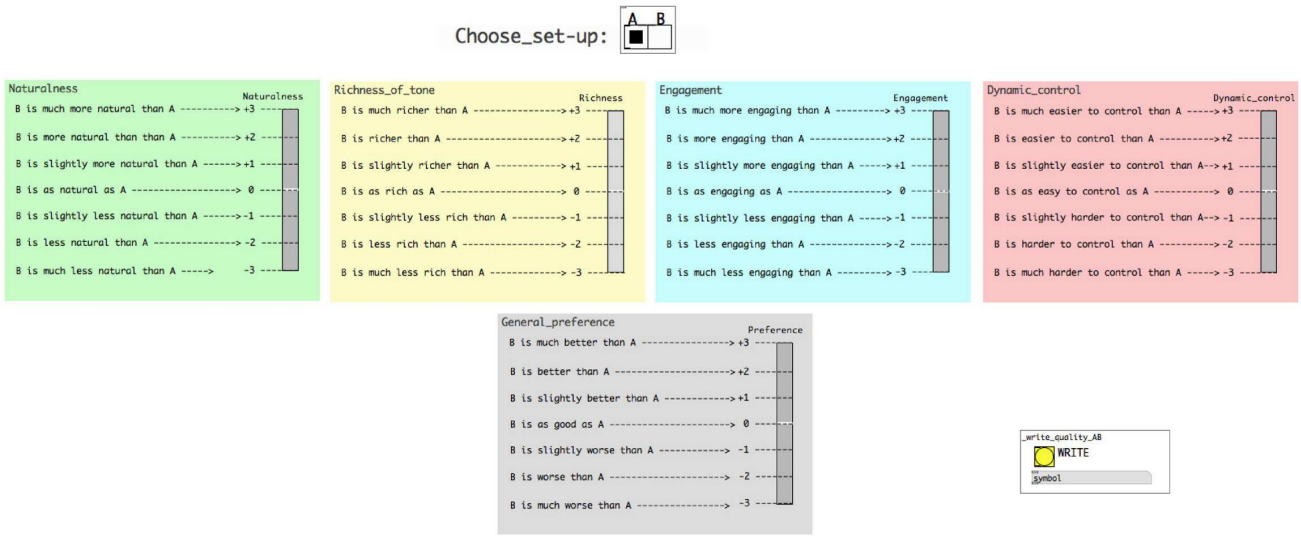


Figure 5. The graphical user interface used by participants to switch between conditions (A,B) and to rate the five attributes.

this way they were not exposed to the sound coming by air conduction from the transducers, as a by-product of their vibration.

3.1 Experiment 1: Quality

3.1.1 Stimuli and conditions

Three vibration conditions were assessed, always relative to a non-vibrating standard stimulus A:

B: recorded real vibrations;
C: recorded real vibrations with 9 dB boost;
D: synthetic vibrations.

Conversely, sound feedback was generated by the same piano synthesizer configuration throughout the experiment.

3.1.2 Design and procedure

The task was to play freely on the digital keyboard and assess the playing experience on five attribute rating scales: Dynamic control, Richness, Engagement, Naturalness, and General preference. The dynamics and range of playing were not restricted in any way.

Subjects could switch freely among setups α and β: Setup α was always the non-vibrating standard, while setup β was one of the three vibration conditions (B, C, D). The rating of β was given in comparison to α. The presentation order of the conditions was randomized. Also, participants were not aware of what could actually change in the different setups, and in particular they did not know that sound feedback would not be altered. The free playing time was 10 minutes per couple of conditions (A, B), and participants were allowed to rate the five attributes at any time during the session by means of a point & click graphical user interface (GUI), depicted in Fig. 5. In the end, each subject gave one rating in each attribute scale for each vibration condition.

Ratings were given on a continuous Comparison Category Rating scale (CCR), ranging from -3 to +3, which is widely used in subjective quality determination in communications technology (recommendation ITU-T P.800). Subjects moved a slider on the continuous scale, to the position which best reflected their opinion. The scale had the following tick marks:

+3: "β much better than α"
+2: "β better than α"
+1: "β slightly better than α"
0: "β equal to α"
-1: "β slightly worse than α"
-2: "β worse than α"
-3: "β much worse than α"

The five attributes were selected based on previous experiments with the Disklavier [10] and recent research on violin evaluation [2].

The GUI for the participants and the software for controlling the conditions and recording data were realized in Pure Data, running on a laptop placed at the subjects' reach.

3.2 Experiment 2: Timing and dynamic stability

The technical setup was the same as in Experiment 1. Additionally, a metronome sound at 120 BPM was delivered through the earphones.

Only conditions A and B were used in the test, alternating realistic vibrations with no vibrotactile feedback.

3.2.1 Design and procedure

Subjects were asked to play an ascending and then a descending D-major scale at pace with the metronome (every second beat), at a fixed given dynamics. Only the three leftmost octaves were considered, so as to maximize the tactile feedback, requiring the subjects to play with their left hand only. Each subject repeated the task for three dynamic levels (*pp*, *mf*, *ff*) three times each, in each condition (i.e., with and without vibrations), for a total of 18 randomized trials. MIDI data consisting of note ON, note length and key velocity messages were recorded across the test for subsequent analysis.

A program made with Pure Data was used to carry the test under the experimenter's supervision, and record MIDI data.

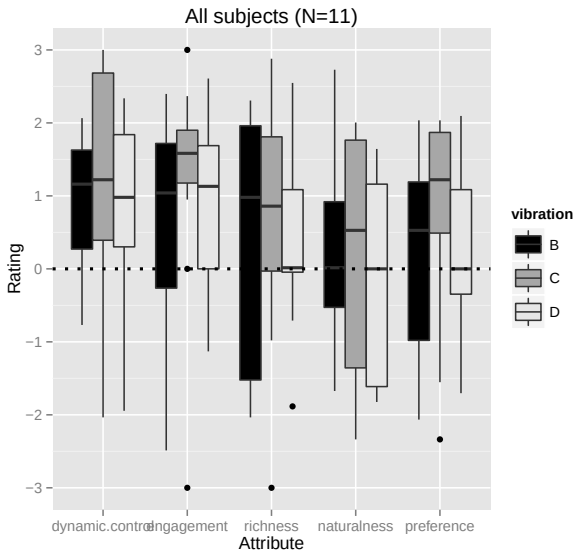


Figure 6. Results of the quality experiment. Boxplot presenting median and quartiles for each attribute scale and vibration condition.

4. RESULTS

4.1 Perceived quality

Inter-individual consistency was assessed for each attribute scale by computing the Lin concordance correlations ρ_c for each pair of subjects [14]. The average ρ_c were 0.018 for general preference, 0.006 for dynamic control, -0.04 for richness, -0.02 for engagement, and -0.04 for naturalness. In all scales, a few subjects either agreed or disagreed almost completely and, due to this large variability, ρ_c was not significantly different from 0 for any of the scales ($t(54) < 0.77, p > 0.05$). The low concordance scores indicate a high degree of disagreement between subjects.

Responses were positively correlated between all attribute scales. The weakest correlation was observed between richness and dynamic control, (Spearman correlation $\rho_s = 0.18$), and the highest between general preference and engagement ($\rho_s = 0.75$). The partial correlations between general preference and the other attribute scales were as follows: $\rho_s = 0.39$ for dynamic control, $\rho_s = 0.72$ for richness, and $\rho_s = 0.57$ for naturalness.

Results are plotted in Fig. 6, and the mean ratings for each scale and vibration condition are given in Table 1. On average, each of the vibrating modes was preferred to the non-vibrating standard, the only exception being condition D for Naturalness. For conditions B and C Naturalness received faintly positive scores. The strongest preferences were for Dynamic range and Engagement. General preference and Richness had very similar mean scores though somewhat lower than Engagement and Dynamic control. Generally, C was the most preferred of the vibration conditions: it scored highest on four of the five scales, although B was considered the most natural. Interesting enough, B scored lowest in all other scales.

As the normality rule for Analysis of Variance was violated, a non-parametric Friedman test of differences among

Vibration	Dyn.	Rich.	Eng.	Nat.	Pref.
B	0.92	0.30	0.50	0.26	0.24
C	1.28	0.67	1.21	0.17	0.81
D	0.87	0.42	1.00	-0.23	0.29

Table 1. Mean ratings over all subjects for each attribute and vibration condition.

repeated measures was conducted for the Preference ratings. It rendered a Chi-square value of 21.9 which was significant ($p < 0.05$), suggesting a significant difference between vibration conditions. However, Wilcoxon signed ranks tests, performed on the hypothesis that the median is positive, were insignificant for all conditions (B: $V = 37.5, p > 0.05$; C: $V = 41, p > 0.05$; D: $V = 28, p > 0.05$).

Heterogeneity was observed in the data, as might be expected due to the high degree of variability in the inter-individual agreement scores ρ_c. A k-means clustering algorithm was used to divide the subjects *a posteriori* into two classes according to their opinion on General preference. Eight subjects were classified into a "positive" group and the remaining three into a "negative" group. The results of the respective groups are presented in Fig. 7. A difference of opinion is evident: The median ratings for the most preferred setup C are nearly +2 in the positive group and -1.5 in the negative group for General preference. In the positive group, the median was > 0 in all cases except one (Naturalness, D), whereas in the negative group, the median was positive in only one case (Dynamic control, B).

4.2 Timing and dynamic stability

The hypothesis was that, if the subjects' timing and dynamic behaviour is affected by key vibrations, differences should be seen in means and standard deviations of key velocities and inter-onset intervals (IOI's).

Mean key velocities were computed for each subject as the average over the three repeated runs for each condition. Results are presented in Fig. 8. At the ff condition, subjects played just slightly louder with vibrations than without, while at the mf condition they played slightly softer in presence of vibrations. However, a repeated measures ANOVA was insignificant for both vibrations ($F(1, 2826) = 2.27, p > 0.05$) and the interaction between vibrations and dynamic condition ($F(2, 2826) = 0.83$, $p > 0.05$). No effect was observed either by studying the lowest octave alone, where the vibrations would be felt strongest. Nor was there a significant difference in the standard deviations between vibrations ON and OFF conditions (95% CI's obtained from paired t-tests (df=10) included $\mu(sd_B) - \mu(sd_A) = 0$ at all dynamic conditions).

IOIs were likewise stable across the two vibration conditions. Generally they were slightly more scattered at the pp dynamic, but no effect of vibrations was observed (see Fig. 9). Note durations were also stable across regardless of vibrations, suggesting that there was no significant difference in articulation or note overlap.

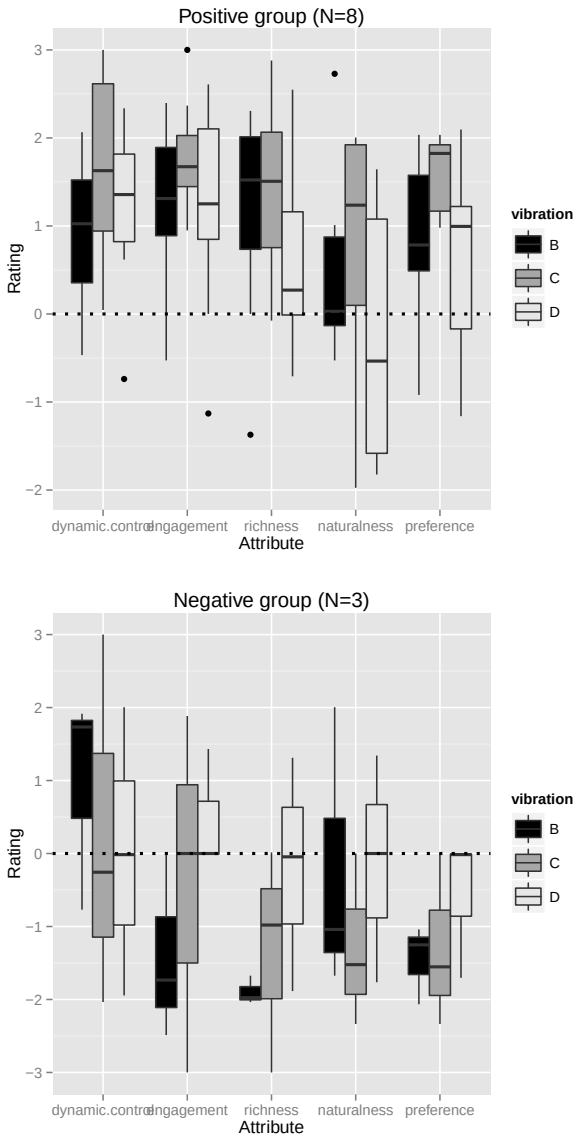


Figure 7. Quality results for the positive and negative groups.

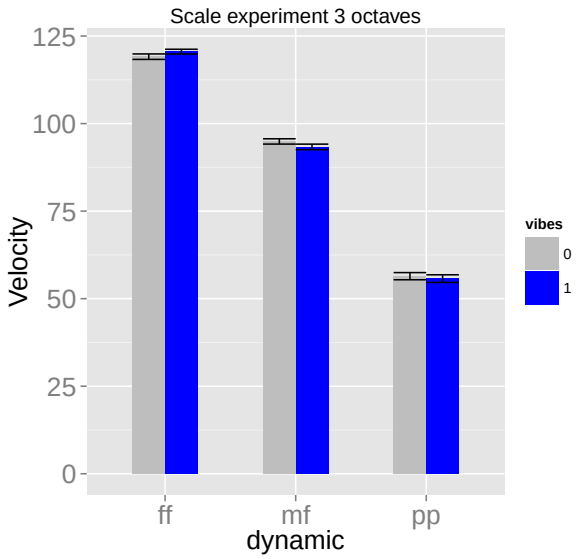


Figure 8. Mean key velocities in Experiment 2, with 95% CI error bars as given in [15].

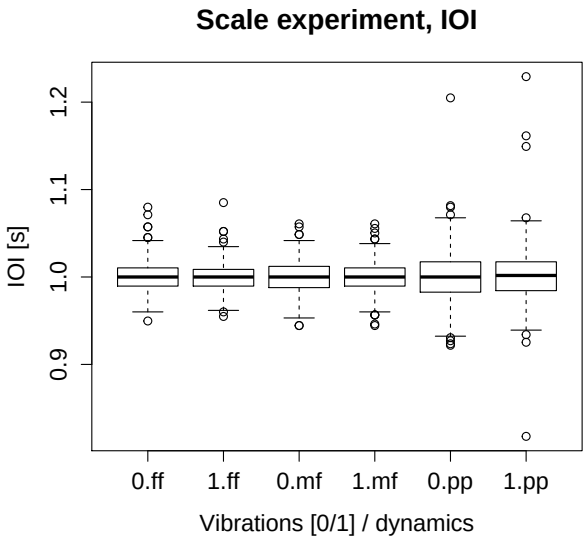


Figure 9. Boxplot of IOI's in the scale experiment.

5. DISCUSSION AND CONCLUSIONS

It is concluded that key vibrations increase the perceived quality of a digital piano. Although the recorded vibrations were perceived as the most natural, amplified natural vibrations were overall preferred and received highest scores on all other scales as well. The other interesting outcome is that the vibrating setup was considered inferior to the non-vibrating standard only in Naturalness for synthetic vibrations. This suggests that pianists are indeed sensitive to the match between the auditory and vibrotactile feedback.

The attribute scales with the highest correlation to General preference were Engagement ($\rho_s = 0.75$) and Richness ($\rho_s = 0.72$). A similar result was obtained in a recent study on violin evaluation, where richness was significantly associated with preference [2].

The high degree of disagreement between subjects suggests that intra- and inter-individual consistency is an important issue in instrument evaluation experiments. Due to only one attribute rating per subject and condition, intra-individual consistency could not be assessed in the present study and will be left for a future revision.

However, the heterogeneity in the data was similar across all attributes and conditions, making it hard to believe it was caused by inconsistency alone. Roughly two thirds of the subjects clearly preferred the vibrating setup, perhaps less rewarded by the synthetic vibrations, while the remaining one third had quite the opposite opinion. It is interesting that both the jazz pianists, having probably more experience of digital pianos than the classical pianists, were in the "negative" minority: would a vibrating digital keyboard be perceived as less pleasant than a neutral one, reflecting a preference of those pianists to the digital piano's traditional tactile response? In the next phase of the experiment, jazz and classical pianists will be studied in two *a priori* groups. Also, subjects will be asked to describe their opinion in a short qualitative interview.

No differences were observed in timing performance and dynamic stability. The task, three octaves of D-major scale in relatively slow pace, was probably easy to perform even without the possible aid of key vibrations. However, recent research shows that pianists use tactile information as a means of timing regulation [16, 17], even though the role of key vibrations remains unknown. There is also evidence that vibrotactile feedback helps force accuracy in finger pressing tasks [18, 19]. Whether vibrations caused by the currently depressed key(s) might help with velocity planning of the upcoming key press, is an interesting question that the present experiment cannot answer. Future experiments will investigate different performative tasks, in which the information provided by vibrotactile feedback is more salient than in the present one. As an example, a different task may involve repeated sustained chords where the player has to maintain the dynamics as constant as possible: in this case key vibrations and their time evolution are perceived more clearly and it may be hypotesized that they aid the control of dynamics.

It may well be that the effect of the key vibrations is purely subjective and simply makes playing the digital piano more engaging. However, subjects also gave high ratings for the perceived dynamic control for all vibrating setups. A future experiment will further investigate this effect on performance level.

Acknowledgments

This research is pursued as part of the project AHMI (Audio-Haptic modalities in Musical Interfaces), funded by the Swiss National Science Foundation (SNSF).

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] C. Saitis, B. L. Giordano, C. Fritz, and G. P. Scavone, "Perceptual evaluation of violins: A quantitative analysis of preference judgments by experienced players," *The Journal of the Acoustical Society of America*, vol. 132, no. 6, pp. 4002–4012, 2012. [Online]. Available: http://scitation.aip.org/content/asa/journal/jasa/132/6/10.1121/1.4765081

[3] M. Marshall and M. Wanderley, "Examining the effects of embedded vibrotactile feedback on the feel of a digital musical instrument," in *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, Oslo, Norway, May 30 - June 1 2011, pp. 399–404.

[4] M. Giordano and M. M. Wanderley, "Perceptual and technological issues in the design of vibrotactile-augmented interfaces for music technology and media," in *Haptic and Audio Interaction Design*, ser. Lecture Notes in Computer Science, I. Oakley and S. Brewster, Eds. Springer Berlin Heidelberg, 2013, vol. 7989, pp. 89–98. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-41068-0_10

[5] C. Cadoz, L. Lisowski, and J.-L. Florens, "A Modular Feedback Keyboard Design," *Comput. Music J.*, vol. 14, no. 2, pp. 47–51, 1990.

[6] R. Oboe and G. De Poli, "A Multi-Instrument, Force-Feedback Keyboard," *Comput. Music J.*, vol. 30, no. 3, pp. 38–52, Sep. 2006.

[7] A. Galembo and A. Askenfelt, "Quality assessment of musical instruments - Effects of multimodality," in *Proc. 5th Triennial Conf. of the European Society for the Cognitive Sciences of Music (ESCOM5)*, Hannover, Germany, Sep. 8-13 2003.

[8] E. Guizzo, "Keyboard maestro," *IEEE Spectrum*, vol. 47, no. 2, pp. 32–33, Feb. 2010.

[9] F. Fontana, S. Papetti, M. Civolani, V. dal Bello, and B. Bank, "An exploration on the influence of vibrotactile cues during digital piano playing," in *Proc. Int. Conf. on Sound Music Computing (SMC2011)*, Padua, Italy, 2011, pp. 273–278.

[10] F. Fontana, F. Avanzini, H. Järveläinen, S. Papetti, F. Zanini, and V. Zanini, "Perception of interactive vibrotactile cues on the acoustic grand and upright piano," in *Proc. Joint ICMC/SMC Conf.*, 2014.

[11] T. Verrillo, "Vibrotactile thresholds measured at the finger," *Perception and Psychophysics*, vol. 9, no. 4, pp. 329–330, 1971.

[12] S. Papetti, H. Järveläinen, and G.-M. Schmid, "Vibrotactile sensitivity in active finger pressing," in *World Haptics Conf.*, 2015, to appear.

[13] M. R. Schroeder, "New method of measuring reverberation time," *J. Acoust. Soc. Am.*, vol. 37, no. 6, pp. 1187–1188, June 1965.

[14] L. Lin, "A concordance correlation coefficient to evaluate reproducibility," *Biometrics*, vol. 45, pp. 255–268, 1989.

[15] R. Morey, "Confidence intervals from normalized data: A correction to Cousineau (2005)," *Tutorial in Quantitative Methods for Psychology*, vol. 4, no. 2, pp. 61–64, 2008.

[16] W. Goebl and C. Palmer, "Tactile feedback and timing accuracy in piano performance," *Exp. Brain. Res.*, vol. 186, pp. 471–479, 2008.

[17] ——, "Finger motion in piano performance: Touch and tempo," in *International symposium for performance science*, 2009.

[18] H. Järveläinen, S. Papetti, S. Schiesser, and T. Grosshauser, "Audio-tactile feedback in musical gesture primitives: Finger pressing," in *Proceedings of the Sound and Music Computing Conference (SMC2013)*, Stockholm, 2013.

[19] T. Ahmaniemi, "Effect of dynamic vibrotactile feedback on the control of isometric finger force," *IEEE Trans. on Haptics*, 2012.

Design and Implementation of a Whole-Body Haptic Suit for "Ilinx", a Multisensory Art Installation

M. Giordano[1], I. Hattwick[1], I. Franco[1], D. Egloff[1], E. Frid[2], V. Lamontagne[3], TeZ[4], C. Salter[3], M. Wanderley[1]
[1]Input Devices and Music Interaction Laboratory (IDMIL)
Centre for Interdisciplinary Research in Music Media and Technology
McGill University, Montréal, Québec, Canada
[2] Sound and Music Computing Group, KTH, Stockholm, Sweden
[3] Concordia University, Montréal, Québec, Canada
[4] Maurizio Martinucci - www.tez.it
marcello.giordano@mail.mcgill.ca

ABSTRACT

Ilinx is a multidisciplinary art/science research project focusing on the development of a multisensory art installation involving sound, visuals and haptics. In this paper we describe design choices and technical challenges behind the development of the haptic technology embedded into six augment garments. Starting from perceptual experiments, conducted to characterize the thirty vibrating actuators used in the garments, we describe hardware and software design, and the development of several haptic effects. The garments have successfully been used by over 300 people during the premiere of the installation in the *TodaysArt 2014* festival in The Hague.

1. INTRODUCTION

Since E. H. Weber's experiments on the sense of touch in the early 19th century (see e.g. [1]) various new fields of sensory research have been established, and principles of human haptic perception have been implemented in virtual scenes, electro-mechanical interfaces, as well as in robotic and bio-mechatronic systems. Nevertheless, prevailing studies remain mostly in engineering or psychology contexts where artists have little access to neither the research nor the tools developed. As a consequence many of these techniques that could have major artistic impacts are confined to technical academic conferences and papers exclusively, as they are not implemented in practical applications, and not incorporated into the vocabulary of artistic expression. The use of haptic technology in a new media art context is a promising area of artistic exploration that lies at the crossroads of engineering, info communications, neuroscience, and art.

The work described in this paper is a successful attempt at exploring this area. It is the result of a multidisciplinary collaboration of haptic researchers, fashion designers and interactive artists, with the goal of creating a vibrotactile augmented garment to be used in "Ilinx", a multi-sensory art installation blending sound, visuals and whole-body vibrations. The design process and technical challenges behind the development of the haptic technology embedded in the garments will be described, together with the implementation of a vocabulary of haptic effects used during the installation.

The garments designed for "Ilinx" [1] illustrate a novel performance system that is able to convert exterior information and translate it into corporeal sensations. The mental manifestations and ideas that arise from the uncanny sensation of shifted proprioception can help to increase the personal awareness of the perceptual space that we occupy in our everyday life, and thus generate a sense of re-embodied presence, reminding us that not everybody feels the same in his or her own skin.

2. ACTUATOR CHARACTERIZATION

The initial phase of the project was dedicated to the choice and characterization of the vibrating actuators to be embedded in the garments. Several factors were clear since the early stages of the project: garments had to be wirelessly controlled (hence battery powered), light, robust and relatively inexpensive but capable at the same time of displaying interesting haptic effects.

We evaluated several different kinds of actuators for use in this project, including eccentric mass (ERM) rotating motors, linear resonant actuators, and tactile transducers. Our primary concerns for choosing an actuator were ease of implementation, price, and size. In particular we looked closely utilizing tactile transducers, which consist of a voice-coil driven by an audio signal (or any AC signal). In our experience the ability of this kind of actuator to respond to a broad spectrum of frequencies provides more flexible tactile stimuli. We chose to utilize ERM motors instead, however, for several key reasons. The first is that we knew in this project that we would be driving large arrays of actuators and we recognized the difficulty of working with a large number of audio signals. ERM motors have the

[1] http://phonomena.net/ilinx/

Copyright: ©2015 M. Giordano[1], I. Hattwick[1], I. Franco[1], D. Egloff[1], E. Frid[2], V. Lamontagne[3], TeZ[4], C. Salter[3], M. Wanderley[1] 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.

benefits of being driven by DC (and hence able to be controlled by PWM signals), available in small form factors, and inexpensive due to their ubiquity.

2.1 Physical Characterization

In order to be able to proficiently design tactile effects to be displayed through the augmented garments, a full characterization of the actuators had initially to be performed, both from a physical and perceptual point of view. This characterization had been conducted in a previous work by some of the authors (Frid et al. [2]). The results are briefly summarized in the next sections.

We performed several measurements to provide an amplitude and frequency characterization of the motors. An Arduino Uno board connected to an IC unit was used to generate the PWM signal needed to drive the motor; a PCB 352C23 [2] 1-axis accelerometer was fixed on the top face of an accelerometer using petro-wax (Fig. 1). We recorded actuator vibrations at 192 *kHz* for 10 distinct duty cycle values of the PWM signal (ranging from 0.2 to 1.0). For each duty cycle step, we measured amplitude and average peak frequencey (Fig. 2) as well as ramp-up- and ramp-down time, i.e. the time need for the motor to go from a full-stop to target vibration amplitude and vice versa (Table 1).

As seen in Table 1, ramp-down times were measured to range from 400 to 610 ms, while ramp-up times were constantly below 15 ms for all PWM duty cycles. Fig. 2 shows a clear correlation of both amplitude and frequency to duty cycle value; as a result, these two parameters can not be separately controlled.

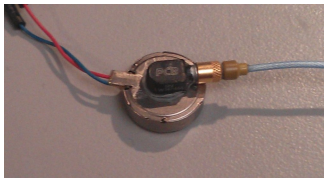


Figure 1: PCB 352C23 1-axis accelerometer fixed to the actuator.

Table 1: Ramp-down times for different duty cycles

Duty cycle	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
t (ms)	400	490	540	580	580	600	600	610	610

2.2 Perceptual Characterization

We conducted [2] two pilot experiments with a total of 8 participants (4 male & female, aged from 21 to 31 years old) for Experiment 1, and 10 participants (5 male & female, aged 21 to 31) for Experiment 2. In Experiment 1 we investigated vibrotactile absolute threshold for 5 discrete duty cycle steps (0.1 to 0.5). For this test, the actuators were placed on the back of the torso, symmetrically about the spine. An elastic Velcro® band was used to guarantee

[2] `http://www.pcb.com/Products.aspx?m=352C23`

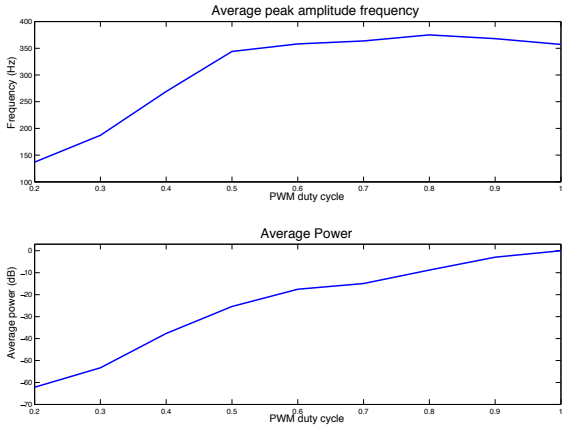


Figure 2: Average peak amplitude frequency (top) and RMS amplitude (bottom) at each discrete PWM duty cycle step from 0.2 to 1. Both these analyses were performed up to 1000 Hz in the original spectrum, which is the upper limit for tactile perception. The average peak amplitude is a weighted average of the most significant frequency peaks found in the spectrum. The frequency range varies from 140 to 380 Hz. Average power is expressed in dB, with maximum amplitude used as reference power.

constant contact between actuators and skin. Participants had to wear headphones presenting pink noise, and were asked to report if they could perceive a 500 ms stimulus at a random duty cycle value.

In Experiment 2, we investigated the ability of participants to discriminate 500 ms long stimuli at different duty cycles, using a two-alternative forced-choice (2AFC) paradigm. With the same apparatus as in Experiment 1, participants were asked to perform "same" or "different" judgements on 81 stimuli pairs of various intensity levels presented in a randomized order.

Results from Experiment 1 indicated that stimuli with a duty cycle equal to or greater than 0.2 can be perceived more than 50 % of the times. Stimuli at 0.1 duty cycle were only perceived 4.2 % of the times. Table 2 summarizes the results from Experiment 2. The required difference in duty cycle for discrimination between two stimuli was found to be a function of the absolute value with reference to the duty cycle scale (0.2-1.0); a larger duty cycle difference is required for a stimulus in the upper duty cycle range than for a stimulus in the lower range. Overall, to ensure robust discrimination, only pairs with duty cycle differences greater than or equal to 0.3 should be used; as seen in 2, such pairs can be perceived as different above chance level.

Table 2: Correctness for different duty cycles. For duty cycle differences greater than 0.3, correctness is above chance.

Duty cycle difference	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Percentage	26	38	55	63	81	83	90	95

3. GARMENT DESIGN

3.1 Preliminary Tests

Early ideas of the wearable prototype were informed by previous work on vibrotactile augmented garment ([3], [4]) and by modular designs of vibrotactile systems developed by the authors and collaborators ([5], [6]). Moreover, a thorough review of literature on perceptual acuity and spatial resolution of the sense of touch at different loci (i.e. [7]–[11]) and of emergence of tactile illusions ([12], [13]) was carried on. This allowed us to have clearer picture of the physiological limitation inherent to the skin the body-parts we investigated during our tests, and at the same time to leverage tactile illusion in order to achieve a wider range of effects.

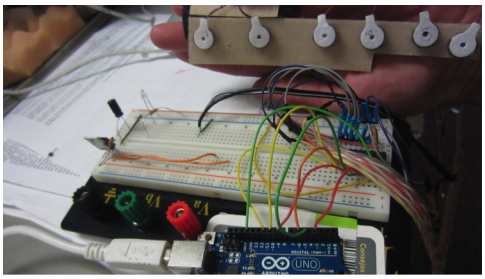


Figure 3: Early prototype using a Dual-Lock Velcro strip.

The first wearable prototype consisted of six pager motors mounted on a 3M Dual Lock velcro strip (Fig. 3). Prototyping with the velcro tape proved rather useful for defining salient distances betweeen individual actuators and receptive fields of the body, as the motors could be easily mounted and rearranged along the tape. During this early test phase, the critical distance between two actuators was evaluated by applying Weber's 2-point discrimination threshold technique: actuators were placed close together at first and then rearranged to increase the distance in-between them until two distinct vibrotactile sensations could be felt.

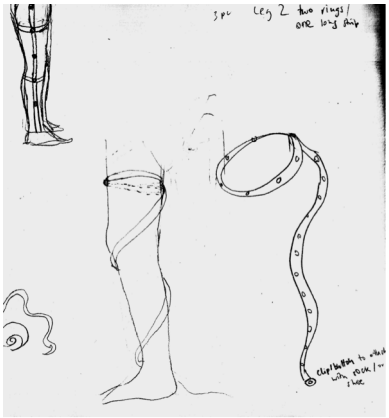


Figure 4: Sketches of possible actuator arrangement using detachable fabric strips.

In order to find the most appropriate receptive fields for stimulation, the actuator strip was applied to various different body parts, such as legs, stomach, back, inner and outer arms, neck and torso. The most salient sensations were found to be when the strip was either placed like a belt around the stomach, put along the outside of the leg, or twirled around the leg. These findings inspired the early sketching shown in Fig. 4.

Vibrations on the throat and neck felt very intense and almost uncomfortable, but spatial acuity was very high at these loci. By the means of these explorative studies, we have found that six actuators were enough to induce a continuously sensation that mimics movement along the arms and legs. Initially, sets of 2x4 actuators were believed to be necessary to convey specific activation patterns, however, the more economic solution of six actuators proved to be just as efficient. We believe that for certain receptive fields on the body as few as three actuators could induce the illusion of a continuous sensation along the skin.

3.2 Garment

Our preliminary tests led us to choose to place actuators in strips of six down the length of both arms and legs as well as in a circle around the torso. After several iterations garments were developed consisting of two chap-like leggings and a single garment with sleeves which are open down the length of the arm. Velcro straps were used to secure the sleeves to the arm in three locations and the leggings to the legs in four locations. In addition a wider velcro strap was used to secure the main body of the jacket to the torso. The open sleeves and leggings came with the advantage of being easy to put on and take off while holding actuators tightly to the body.

3.3 Hardware

The electronics for each garment consist of five circuit boards on each limb segment (two arms, two legs, and torso) and a single central processing unit. Each limb segment's circuit board contains power regulation, a 9DoF movement sensor consisting of 3-axis accelerometers, gyroscopes, and magnetometers, and a microcontroller for generating control signals for that segment's six motors. The central processing unit, as seen in figure 6, consists of a BeagleBone Black (BBB) microprocessor running an embedded distribution of Linux and a WiFi dongle. The BBB is responsible for transmitting and receiving messages over WiFi and routing incoming messages to the appropriate motor driver board.

As noted above the ERM motors have a prolonged ramp-down time. In order to compensate for this motor driver circuits were implemented that feature a 'braking' function which creates a short between the two motor terminals. This braking function is called for 100ms every time the PWM value of a motor's control signals transitions from a non-zero to zero value.

One key concern of the costume designers was that the motors be attached to the garments in such a way as to minimize the impact of the wiring on the garment's flexibility. The ethernet cables connecting the driver boards to the BBB had their external covers removed for this reason, and we chose to use conductive thread for connecting the motors to the motor driver boards on the limbs segments. It



Figure 5: The final version of the garment. The actuators are clearly visible on the two leg modules and on the jacket (sleeves and waist). The green labels show the name of the modules as they are referred to in Sec. 4.2.

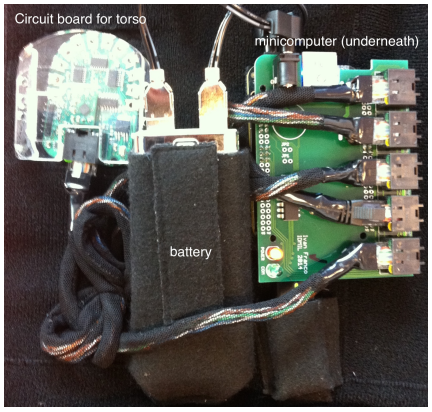


Figure 6: A driver board (left side) connected to the custom cape designed for the Beaglebone Black (BBB) central unit.

was also necessary to find a way to securely attach the motors to the garment which would also provide for a close fit between the motors and the body when the garment was worn. We created a 3D-printed housing for the motor (shown in figure 7) which contained three circular mounting points for sewing to the garment. The wires connecting to the motor were soldered to ring terminals which fit into cutouts in the housing, and then conductive thread was embroidered around both the holes in the housing and the ring terminal, fastening the housing to the garment and making an electrical connection at the same time.



Figure 7: 3D printed housing for connecting the motors to the garments.

4. CONTROL SYSTEM

4.1 System Architecture

The control of the system is relayed through a central processing unit based on the Beaglebone Black (BBB), a popular single-board computer with a 1GHz ARM Cortex-A8 processor. The BBB controls each of the individual driver boards through a custom PCB add-on, which implements a standard SPI bus. This SPI custom board is mounted directly on the BBB board and provides connectivity and power to each of the five driver boards through standard RJ45 connectors. Power is provided by a battery with two independent outputs, one plugged to the BBB and another to the SPI board (which also powers the driver boards), in order to share load and avoid possible power spikes in each of the subsystems.

The BBB is also connected to a local wireless network through a small WiFi USB dongle, so that each garment can be controlled by any other device on the same network via a messaging system based on the Open Sound Control protocol (OSC). In practice each garment can be understood as an individual OSC server, to which commands can be sent through an unique IP address. Additionally this self-contained system can be monitored and controlled through a *ssh* connection via standard Unix shell, through which it is possible to check execution results, manage running processes or audit the system's processing load.

4.2 Message Namespace

As shown in figure 8 the BBB is constantly running a python script that processes the incoming OSC messages and relays the respective command to each of the five driver boards through the SPI bus. The messaging system implements an abstraction layer through a namespace in which driver boards are associated to particular body segments according to the following convention (see Fig. 5):

- /ar: Right Arm
- /al: Left Arm
- /tf: Torso
- /ll: Left Leg
- /lr: Right Leg

Individual motors are addressed through a normalized value which represents the motor's location on the body segment, which can be discretized with an approximation to

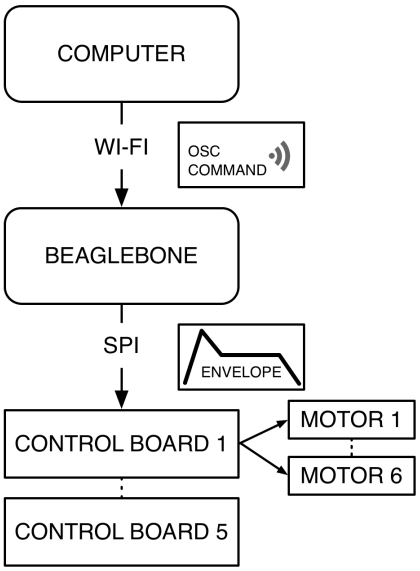


Figure 8: Signal flow of the system from the mainframe computer to the individual motors on each garment.

the nearest motor. This approach also allow for an arbitrary number of motors on each body segment, which might be scaled up or down according to specific application needs. The rest of the message is composed of an amplitude envelope with the intended response for the triggered event over time. A pseudo-message should contain the following data:

```
[limb, normalized motor position, attack time, decay time, sustain level, release time]
```

A practical example of sending an envelope to the middle motor of the left leg would be similar to:

```
/ll 0.5 250 500 0.7 250
```

This protocol allows for the generation of the control signals for the motor to be located on each driver board while the higher-level definition of haptic effects takes place on a remote computer. While it could have been possible to define and embed higher-level effects directly on the hardware system, this simple and effective control protocol permits users to design their own effects and control the system with many different types of software. In this project these effects were programmed in *Max/MSP*, a modular programming language oriented to music and media, and integrated into the performance control mainframe. This allowed for an easy and accurate synchronization between haptics and audiovisual events happening throughout the installation, which were also controlled though a Max-based software.

5. HAPTIC EFFECTS

We were interested in discovering and defining specific haptic effects to utilize during the composition process. We identified two main categories of effects to implement using this system: discrete and continuous. *Pokes*, *Buzzes* and *Sparkles* fall into the first category. The first two effects are achieved through a simple activation of one or more spatially close motors. A *Poke* is implemented sending a sharper envelope message, while a *Buzz*-envelope has longer attack and decay times. The *Sparkles* effects consist of random actuation of actuator all over the body, or limited to one specified limb. The key differentiator of a discrete effect is that each instance of the effect is perceived as occurring at a single location on the body.

Continuous effects, on the other hand, use a combination of motors to create sensations that are perceived as moving on the body, and they rely on a precise pattern of actuation. The *Snake* effect requires the definition of a starting and ending point, duration, intensity and overlap factor (i.e. overlap between subsequent motor activations). An illustration of the effect is depicted in Fig. 9. Several other continuous effects were created. A wave effect reproduces the effect of a wave traveling horizontally or vertically across the body and effected by a sequential, overlapping activation of contiguous motors on a body segment. A variant of this is a spin effect in which the motors on the torso are activated in a continuous loop.

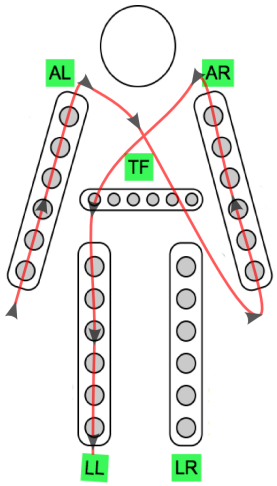


Figure 9: The representation of the garment in *Max*. The red path shows the actuation pattern of the *Snake* effect: a wave of vibrations traveling along the limbs, following a specified order.

6. PERFORMANCE

A brief description of the installation is provided in the following paragraphs, together with participants' feedback collected through short, informal interviews at the end of the installation.

More information regarding the garments' use in the installation as well as an overview of the installation itself will be provided in forthcoming publications.

Here we will instead focus on a few details pertaining to the perception of haptic effects.

The initial presentation of "Ilinx" was from September 25-28 at *TodaysArt 2014* festival in The Hague. Over four

days more than 300 visitors experienced the immersive environment while wearing the garments. The installation was divided into two sections. In the first section, participants enter the pitch-black room hosting the installation and are instructed to seat on the ground. The suits get activated and produce a vibration pulse effect, which is synchronized to a bell-like sound produced by quadraphonic speakers. The duration of this section is of roughly 10 minutes, and new sonic material is progressively introduced throughout the section. The second section starts with the appearance of faint lights, and at this point participants are free to stand and explore the room, while more visual and sonic effects appear. Vibration pattern matching sound and light effects in the room continue to be delivered through the suit.

Our experience using the system during the installation demonstrated vividly that the fit of the suit determined how effectively the actuators' vibrations were transferred to the body. In particular, getting the jacket tight enough around the waist for the vibrations to be as perceptible as they were on the limbs was difficult.

6.1 Participants' Feedback

Six informal interviews involving volunteer participants were conducted right after the installation. From the feedback we collected we could extrapolate the following main points about participants' perception of the suits and the haptic effects:

- Participants experienced different degrees of satisfaction concerning the tightness of the suit. Some of them found it too loose, while others judged the tightness to be good enough to guarantee constant actuator-skin contact.
- Even when the suit size perfectly matched participants' body-size, the actuators on the back were still too lose for participants to perceive them clearly;
- Participants consistently underestimated the number of actuators embedded in the suit (responses varied from 10 to 20). This might be due to the lower spatial resolution of the skin in targeted areas;
- Vibrations were felt more clearly in the first section of the piece. In the second section, when participants were standing and walking in an environment full of rich auditory and visual stimuli, the focus shifted from the tactile sense to the other senses.

Overall, participant enjoyed the installation, judging it surprising and engaging. They agreed that, in the body parts for which a sufficient level of tightness was reached, vibration effects could clearly be perceived. They perceived haptic effects such as the *Snake* as continuous vibrations travelling across the body. This suggests that the haptic effects we designed were accurately rendered through the suits.



Figure 10: One visitor at *TodaysArt 2014* wearing the final version of suit.

7. CONCLUSIONS

We presented the outcome of a collaborative project which resulted in the creation of a six tactile-enhanced garments to be used in "Ilinx", a multi sensory art installation. The creation of these garments was driven by a perceptual research-based methodology as well as by artistic and functional considerations.

In this paper, we focused on the haptic research which motivated the choice of actuator placement and effect design, and on development of custom hardware and software solution that were embedded in the suits. The technology we developed proved to be reliable and robust, and capable of allowing the creation of a variety of haptic effects. Given the time constraints, and the practical demands due to the needs of the artistic project, only a small fraction of the expressive potential of the suit could be explored. Several forthcoming projects based on the use of the suits will enable us to take full advantage of this potential and expand the vocabulary of available haptic effects.

Acknowledgment

This research was funded by a grant from the Canada Arts Council.

References

[1] E. H. Weber, D. J. Murray, and H. E. Ross, *The sense of touch*. Academic Press for Experimental Psychology Society, 1978.

[2] E. Frid, M. Giordano, M. M. Schumacher, and M. M. Wanderley, "Physical and perceptual characterization of a tactile display for a live-electronics notification system", in *Proceedings of the joint International Computer Music Conference (ICMC) and Sound and Music Computing Conference (SMC)*, 2014.

[3] E. Gunther and S. O'Modhrain, "Cutaneous grooves: composing for the sense of touch", *Journal of New Music Research*, vol. 32, no. 4, pp. 369–381, Dec. 2003.

[4] J. B. F. van Erp, H. A. H. C. van Veen, C. Jansen, and T. Dobbins, "Waypoint navigation with a vibrotactile waist belt", *ACM Transactions on Applied Perception*, vol. 2, no. 2, pp. 106–117, Apr. 2005.

[5] H. Knutzen, T. Kvifte, and M. Wanderley, "Vibrotactile feedback for an open air music controller", English, in *Sound, Music, and Motion - Lecture Notes in Computer Science (LNCS)*, ser. Lecture Notes in Computer Science, M. Aramaki, O. Derrien, R. Kronland-Martinet, and S. Ystad, Eds., Springer International Publishing, 2014, pp. 41–57.

[6] D. Egloff, J. Braasch, P. Robinson, D. Van Nort, and T. Krueger, "A vibrotactile music system based on sensory substitution.", *The Journal of the Acoustical Society of America*, vol. 129, no. 4, 2011.

[7] R. W. Cholewiak and C. McGrath, "Vibrotactile targeting in multimodal systems: accuracy and interaction", in *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Ieee, 2006, pp. 413–420.

[8] R. W. Cholewiak, A. A. Collins, and J. C. Brill, "Spatial factors in vibrotactile pattern perception", in *Proceedings of Eurohaptics*, 2001.

[9] E. Piateski and L. Jones, "Vibrotactile pattern recognition on the arm and torso", in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Ieee, 2005, pp. 90–95.

[10] L. A. Jones, J. Kunkel, and E. Piateski, "Vibrotactile pattern recognition on the arm and back", *Perception*, vol. 38, no. 1, pp. 52–68, 2009.

[11] J. B. F. van Erp, "Vibrotactile spatial acuity on the torso: effects of location and timing parameters", in *Proceedings of Eurohaptics*, IEEE, 2005, pp. 80–85.

[12] F. A. Geldard and C. E. Sherrick, "The cutaneous "rabbit": a perceptual illusion", *Science*, vol. 178, no. 4057, pp. 178–179, 1972.

[13] V. Hayward, "A brief taxonomy of tactile illusions and demonstrations that can be done in a hardware store.", *Brain research bulletin*, vol. 75, no. 6, pp. 742–52, Apr. 2008.

MUSIC CONTENT DRIVEN AUTOMATED CHOREOGRAPHY WITH BEAT-WISE MOTION CONNECTIVITY CONSTRAINTS

Satoru Fukayama, Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{s.fukayama,m.goto}@aist.go.jp

ABSTRACT

We propose a novel method for generating choreographies driven by music content analysis. Although a considerable amount of research has been conducted in this field, a way to leverage various music features or music content in automated choreography has not been proposed. Previous methods suffer from a limitation in which they often generate motions giving the impression of randomness and lacking context. In this research, we first discuss what types of music content information can be used in automated choreography and then argue that creating choreography that reflects this music content requires novel beat-wise motion connectivity constraints. Finally, we propose a probabilistic framework for generating choreography that satisfies both music content and motion connectivity constraints. The evaluation indicates that the choreographies generated by our proposed method were chosen as having more realistic dance motion than those generated without the constraints.

1. INTRODUCTION

Motion capture systems are widely used to create choreographies for dancing robots or computer animated characters. However, this methodology does not provide flexibility in creating choreographies with various types of music since the choreography needs to be manually created from scratch for every change in the accompanying music. Motion capture systems are often unavailable to those who create dance motion video clips and upload them to video sharing services on the Internet. They usually design choreographies by setting each pose on key frames, which requires a considerable amount of time. We aim to achieve automated choreography to generate dance motions of computer animated characters accompanied by an arbitrary music.

We define automated choreography as a task to automatically generate choreography by leveraging the music content. Previous approaches to generating choreography tried to find dance motion that mostly match the music segment from the viewpoint of various music features. Music features such as tempo [1, 2], beats [3–5], combinations of acoustic features [6], music structures [7], pitches [8], and melodic contours [9] have been used to analyze the relationships between music and dance.

Copyright: ©2015 Satoru Fukayama, Masataka Goto 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.

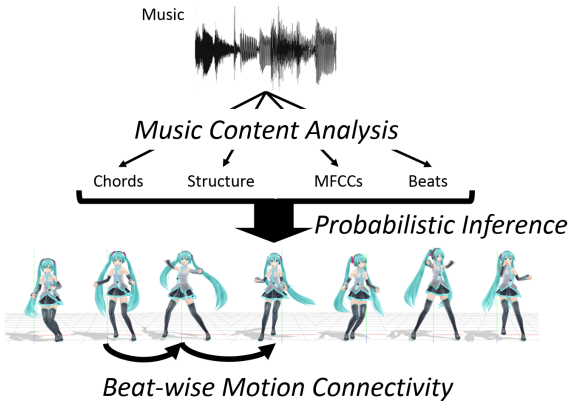


Figure 1. Overview of this research. Various music features are leveraged to generate choreography by concatenating the dance motions. To maintain quality of choreography, motion connectivity constraints are introduced. Generating process is implemented in probabilistic framework. Generated sample of choreographies can be found at https://staff.aist.go.jp/s.fukayama/SMC2015/.

However, the following three issues have not yet been addressed. First, which music features are useful in generating choreography has not been investigated. Second, the connectivity constraints of dance motions have not been considered when the length of the motions are short to reflect the constraints based on the music. Finally, the way to combine music constraints and motion connectivity constraints by using the limited amount of data is not clear.

We propose a novel framework for solving these problems. First, we investigate which music content gives the most useful constraints for choreography based on a data driven approach. Second, we propose a novel method for considering the physical constraints of choreography, such as avoiding unnatural motions and encouraging repetitions. Third, we discuss a probabilistic framework that can simultaneously consider both music constraints and motion connectivity constraints even when there is a limited amount of motion data.

In our probabilistic framework, there are two technical novelties. First, training the probabilistic model that represents the relationship between the music content and dance motion often suffers from data sparseness. We solve this by

using the linear combination of probabilistic models where every model holds information about the relationship between each musical feature and the dance motion. Second, calculating the probability of concatenating the dance motion is difficult, since most of the dance motions only appears once in the data and it is impossible to observe various transitions from the same dance motion. We therefore perform the interpolation of probabilistic values by leveraging the distance between dance motions and calculating the transition probabilities.

2. MUSICAL CONSTRAINTS

What kind of musical features are useful for choreography? Our research aims to give a tentative answer in a data driven approach.

Our method is leveraged by various music analysis techniques. Although there has been previous research in automated choreography that is leveraged by acoustic analysis [6] and structural analysis [7] of pieces of music, we are not aware of research that tried to utilize musical features such as chord labels or up/down-beats, which is one of our research contributions.

2.1 Acoustic feature (MFCCs)

Mel-frequency cepstral coefficients (MFCCs) and their first- and second-order frame-to-frame differences (delta MFCCs and delta-delta MFCCs, respectively) are used to change the choreography through the auditory differences of the music. MFCCs and the deltas are widely used in acoustical analysis of music as they are said to approximate the human auditory system's response. We chose 16 as the dimension of the coefficients, which led us to calculate vectors with 48 dimensions consisting of 16 dimensions for each MFCCs, delta MFCCs, and delta-delta MFCCs vector.

As the supply of choreographies and music training samples are limited, and to avoid overfitting between choreography and the musical features, we do not use the values of MFCCs themselves but instead use an index of a feature cluster. The feature clusters are obtained by conducting k-means clustering with a fixed number of clusters (500). The clustering is done after dimension reduction by performing principal component analysis (PCA) on the data to avoid insufficient clustering caused by the high dimensionality of the data. It reduced the number of dimensions from 48 to 16.

2.2 Musical structure

Analysis results of music structural segmentation are used to create structure and highlight segments in choreography. Structural segmentation detects and labels similar segments in a piece of music. It can be used in choreography to generate similar dances among segments with the same label. Segments labeled as chorus sections, which are the most highlighted segments, can be used to generate relatively active motions compared to the other parts.

Structural segmentation can be conducted by analyzing the self-similarity matrix (SSM) of frame-by-frame acoustic features such as MFCCs or chroma vectors. We used an SSM-based approach, analyzed the hierarchical structure of the music, and simultaneously detected the chorus section.

The results of the hierarchical structural segmentation were encoded into vectors, for example as $[1, 0, 0, 1, 0]$, containing binary values that each indicated whether the segment belonged to the nth hierarchical structure. The dimension of this encoded vector was set to the maximum number of hierarchical structures observed in the music we used. When the segment was detected as a chorus section, the first component of the vector was replace with 2 as $[2, 0, 0, 1, 0]$.

2.3 Beat locations and measure boundaries

The information regarding beat locations and measure boundaries is useful for aligning choreography to music. Since choreography is usually described for every beat or "count", it is natural to consider beats when creating choreography. Furthermore, the up-beat and down-beat information that can be obtained from the measure boundaries and the beat locations is useful in differentiating the moves depending on the strength of each beat.

The beat locations and measure boundaries were first analyzed with the beat detection module based on the calculation of the beat salience function. The analysis results obtained from the module were manually corrected afterwards. All the beats were labeled with integers indicating the beat order in a measure and the number of beats in a measure, such as "1/4", "2/4", "3/4", and "4/4" for a measure with 4 beats.

2.4 Chord sequence

The chord sequences bring us similarity information for the beats while the hierarchical structure gives us more global similarity information for the sections. Although the chords, especially the chord labels, do not seem to be helpful in creating dances, their information can be used as supplementary queues for creating local structure in choreography. Even though we can not uniquely determine what kind of choreography should be aligned to a specific chord label, we can generate similar motions for segments with the same chord labels.

Chord sequences were analyzed with the automatic chord recognition module based on chroma features and Hidden Markov Models (HMM). The information described in a chord label included the root note, chord type, and base note if the root note was not the base note.

3. MOTION CONNECTIVITY CONSTRAINTS

When we try concatenating the fragments of dance motions (motion fragments) to generate choreography, concatenating fragments with long lengths seems to be a reasonable strategy to ensure the quality. This is because the generated results contain more motions which match those in the choreography database. Setting a shorter segment length increases the risk of generating motion that seems to be random and lacking context.

However, because of the limited size of the choreography database, there is a trade-off between finding longer segments and satisfying more musical constraints. A longer segment contains more beats than shorter segments, so the number of beat-wise musical constraints increase, and this make it difficult to find a long segment that satisfies those constraints.

Thus, we concatenate the fragments with a short length that can satisfy the beat-wise musical constraints. The motion connectivity constraints are simultaneously considered to avoid randomness in the choreography.

3.1 Smoothness of fragment transition

To avoid generating discrete moves when concatenating the fragments, the smoothness between two adjacent fragments should be considered. Previous research into choreography with concatenation approach has tended to check the smoothness by calculating only the similarity between the end of the first fragment and the beginning of the following fragment [3]. As this approach does not take smoothness between the connection points into account, the degree of success largely depends on what kind of interpolation (linear, spline and so forth) is used. Therefore, we calculate the distance between two fragments by summing up the distances among all the points between the connection points.

3.2 Repetitions

Repetitive moves are often observed in choreography. We created a hypothesis for preferred and not-preferred types of repetitions and imposed constraints on the concatenation of motion fragments.

Too much repetition affects the naturalness of the dance especially when the repetitions are within a few beats. To avoid this, we set a constraint to prohibit using fragments that appeared in the past 4 beats.

On the other hand, repetition of segments of 4 beats or 8 beats is popular. Thus, we impose constraints on the motion to encourage this kind of repetition. The way to constrain the motion in this manner is described in the next section.

3.3 Phrasing of dances

Without proper constraints, concatenation of motion fragments tends to generate motions without phrasing. Here, the phrasing is the segmented structure of continuous movements, not having a sudden halt in the middle of a segment.

Therefore, we monitor the "activeness" value of each motion fragment, which is calculated by taking the squared sum of the frame-to-frame differential of the body movements. Constraints are imposed on the sequence of fragments to prevent a drastic change in activeness between adjacent fragments.

3.4 Parallel shift

Even though we impose constraints to ensure a smooth change between fragments as described above, smooth movements, such as the parallel shift of the dancer, look strange and are hard to recognize as human movements. This is because the legs are usually used to move a body horizontally; however, some concatenation that considers only the smoothness between the fragments may generate motions of shifting horizontally without moving the dancers legs.

From our observation, these strange moves often occur when there is a position change of the body center without changes in the rotation angles of the legs. We impose constraints to avoid these kinds of movement.

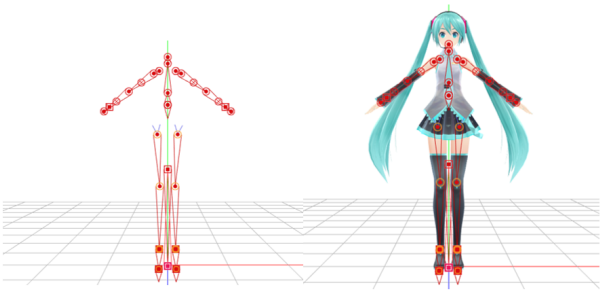


Figure 2. 28 bones (on the left) are used in our formulation to simulate the movements of a dancer (on the right). 5 bones are inverse kinematic bones (IK bones), which jointly move other bones, and movements of each bone are described with 7 values (3 values for position, and 4 values for rotation). Movements of other 21 bones are described with 4 values for rotation.

4. MATHEMATICAL FORMULATION OF AUTOMATED CHOREOGRAPHY

4.1 Data structure for poses

Choreography can be represented with frame-by-frame sequential values of positions and rotations of "bones". Here, bones are the structures embedded in the 3D model of a dancer that approximately correspond to the real bones in a human. Each bone is connected to the other bone to construct a human body. We chose 26 bones to simulate the dancer's movements. The chosen bones are shown in Fig. 2.

The chosen bones consist of 5 inverse kinematic (IK) bones and 21 ordinary bones. The IK bones jointly move the other bones that are connected to the IK bones to avoid unrealistic gestures such as disjointed toes. The IK bones consist of "body center", "left toe", "right toe", "left leg", and "right leg". The moves of these IK bones are represented with position and rotation from the original position, which is shown in Fig. 2. The bones are described with 3 values for the three dimensional position of a dancer on the stage and 4 values for the rotation represented with a quaternion. To summarize, 7 values represent the state of an IK bone. The other ordinary bones are represented with only rotation, which requires 4 values since the position of these bones are calculated from the position of the IK bones. In total, 119 values describe a pose at each frame, although our proposed framework can cope with different

settings of bones and values.

4.2 Concatenation approach

We aim to use the relationship between the musical constraints and the choreography to generate dance motion from music. We chose the concatenation approach that firstly extracts motion fragments from the dance motion database, then analyzes the relationships between the fragments and the corresponding music constraints, and finally concatenates them to generate a new choreography.

Since the musical constraints can change at every beat, motion fragments should include the pose at (or closest to) the time of the beat onset. The fragment also needs to include the motion behind the current beat and the one towards the next beat since the connectivity between the fragments should be analyzed.

The fragments are cut out in lengths of 2 beats, locating the beat onset in the center of each fragment. Let b_i be the frame index of the ith beat onset. Let $\mathbf{x}[n]$ be the pose vector consisting of 119 values for the positions and rotations of the bones at frame index n. The motion fragment extracted from the neighborhood of b_i is:

$$\mathbf{X} = \{\mathbf{x}[b_{i-1}], \cdots, \mathbf{x}[b_i], \cdots, \mathbf{x}[b_{i+1}]\} \tag{1}$$

where $\mathbf{X}$ denotes the set of pose vectors of the fragment.

We can concatenate the adjacent motion fragments through linear interpolation. For instance, the concatenation of $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ is

$$\mathbf{x}[n] = \begin{cases} \mathbf{x}^{(i)}[n] & b_{i-1} \leq n \leq b_i \\ \frac{b_j - n}{b_j - b_i}\mathbf{x}^{(i)}[n] + \frac{n - b_i}{b_j - b_i}\mathbf{x}^{(j)}[n] & b_i \leq n \leq b_j. \\ \mathbf{x}^{(j)}[n] & b_j \leq n \leq b_{j+1} \end{cases} \tag{2}$$

Following the discussion in Section 3.1, the smoothness $\mathscr{S}$ for concatenation of $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ can be defined with the distance between portions of two motion fragments where these two are interpolated:

$$\mathscr{S}\left(\mathbf{X}^{(i)}, \mathbf{X}^{(j)}\right) = \sum_{n=b_i}^{b_j} \left|\mathbf{x}^{(i)}[n] - \mathbf{x}^{(j)}[n]\right|^2 \tag{3}$$

4.3 Probabilistic models for choreography

To generate choreography that is as human-like as possible, we use the tendencies of how the motion fragments appear corresponding to the musical constraints in the motion database. In our method, we capture these tendencies by using probabilistic modeling.

Let A_k, S_k, B_k, and C_k be the labels of musical constraints (acoustic feature, musical structure, beat, and chord, respectively) described in Section 2, which are aligned to the kth motion fragment in the database. The probability for observing $\mathbf{X}^{(i)}$ at the kth frame is the conditional probability, which is represented as $P\left(\mathbf{X}^{(i)}|A_k, S_k, B_k, C_k\right)$.

When we try to train this model, it is difficult to exhaustively observe all the combinations of the musical constraints. Therefore we factorize the probability into sub-models holding information for each musical constraint as:

$$\begin{aligned} &P\left(\mathbf{X}^{(i)}|A_k, S_k, B_k, C_k\right) \\ &= \lambda_0 P\left(\mathbf{X}^{(i)}\right) + \lambda_1 P\left(\mathbf{X}^{(i)}|A_k\right) + \lambda_2 P\left(\mathbf{X}^{(i)}|S_k\right) \\ &\quad + \lambda_3 P\left(\mathbf{X}^{(i)}|B_k\right) + \lambda_4 P\left(\mathbf{X}^{(i)}|C_k\right) + \lambda_5 U \end{aligned} \tag{4}$$

where λ_m $(m = 0, \ldots, 5)$ are the interpolation coefficients satisfying $\sum_m \lambda_m = 1$ and $^{\forall} m, \lambda_m > 0$, and U is the uniform distribution to conduct smoothing. These coefficients are tuned by splitting the training data into two portions, training the sub-models with the first portion, and then maximizing the log-likelihood of the second portion with respect to λ_m.

Since the frequency of the appearance of fragment $\mathbf{X}$ given the condition $Y \in \{A, S, B, C\}$ is sparse, we revise the frequencies using a kernel function and then calculate the conditional probabilities using the revised frequencies. This method introduce kernel functions $\phi_m(\mathbf{X})$ $(m = 1, \cdots, M)$, which returns the similarity between an arbitrary $\mathbf{X}$ and $\mathbf{X}^{(m)}$ in the training data, where M is the number of fragments extracted from the database. Let $c(\mathbf{X}, Y)$ be the frequency of the appearance of fragment $\mathbf{X}$ when the condition value is Y, and let $\hat{c}(\mathbf{X}, Y)$ be the revised frequency. The revised frequency and the conditional probability can be obtained by

$$\hat{c}(\mathbf{X}, Y) = \sum_{m=1}^{M} \phi_m(\mathbf{X})\, c\left(\mathbf{X}^{(m)}, Y\right), \tag{5}$$

$$P\left(\mathbf{X}^{(i)}|Y\right) = \frac{\hat{c}\left(\mathbf{X}^{(i)}, Y\right)}{\sum_{m=1}^{M} \hat{c}\left(\mathbf{X}^{(m)}, Y\right)}. \tag{6}$$

$\mathrm{P}\left(\mathbf{X}^{(i)}\right)$ can also be inferred in this manner. We set the kernel function to be the Gaussian distribution as $\phi_m(\mathbf{X}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\mathscr{D}\left(\mathbf{X}, \mathbf{X}^{(m)}\right)\right)$ where $\mathscr{D}\left(\mathbf{X}, \mathbf{X}^{(m)}\right) = \sum_n \left|\mathbf{x}[n] - \mathbf{x}^{(m)}[n]\right|^2$.

The transition probability between fragments can be calculated by using the smoothness measure $\mathscr{S}$ defined in Equation (3). The probability for transitioning from $\mathbf{X}^{(i)}$ to $\mathbf{X}^{(j)}$ is calculated by

$$P\left(\mathbf{X}^{(j)}|\mathbf{X}^{(i)}\right) = \frac{\exp\left(-\frac{1}{2}\mathscr{S}\left(\mathbf{X}^{(i)}, \mathbf{X}^{(j)}\right)\right)}{\sum_{m=1}^{M} \exp\left(-\frac{1}{2}\mathscr{S}\left(\mathbf{X}^{(i)}, \mathbf{X}^{(m)}\right)\right)}. \tag{7}$$

Now we can define the automated choreography in a probabilistic formulation. Given the musical constraints on beats $(k = 1, \cdots, K)$, generating the concatenation of fragments is performed by maximizing the probability $P\left(\mathbf{X}_1 \cdots \mathbf{X}_K | \{A_k\}_{k=1}^K, \{S_k\}_{k=1}^K, \{B_k\}_{k=1}^K, \{C_k\}_{k=1}^K\right)$ with respect to $\mathbf{X}_1 \cdots \mathbf{X}_K$. By taking the logarithm of this probability with first-order Markov assumption, we can derive that this is equivalent to maximizing the objective function

$$\begin{aligned} J(\mathbf{X}_1 \cdots \mathbf{X}_K) &= \sum_{k=1}^{K} \ln P(\mathbf{X}_k | A_k, S_k, B_k, C_k) \\ &\quad + \sum_{k=1}^{K} \ln P(\mathbf{X}_k | \mathbf{X}_{k-1}) \end{aligned} \tag{8}$$

with respect to $\mathbf{X}_1 \cdots \mathbf{X}_K$. Note that we calculated $P(\mathbf{X}_1|\mathbf{X}_0)$ as $P(\mathbf{X}_1)$. The motion fragments that maximize J can be calculated by using dynamic programming. Since the search space for concatenating fragments is huge (M^K possibilities), we used pruning methods to limit the search space and to make the problem computationally feasible.

4.4 Applying motion connectivity constraints

To impose motion connectivity constraints (described in Section 3), the probability distributions and the search space for generating choreographies are revised. Note that the smoothness constraints are already considered in the transition probability of fragments as described in Section 4.3.

Generating repetition of fragments can be implemented by sharing the musical constraints and revising the probability. For instance, if we expect similar fragments at k and k', then probabilities $P(\mathbf{X}|A_k, S_k, B_k, C_k)$ and $P(\mathbf{X}|A_{k'}, S_{k'}, B_{k'}, C_{k'})$ are both renewed to the linear interpolation of these distributions *i.e.* in accordance with $\frac{1}{2}\{P(\mathbf{X}|A_k, S_k, B_k, C_k) + P(\mathbf{X}|A_{k'}, S_{k'}, B_{k'}, C_{k'})\}$.

Phrasings of choreography are generated by monitoring the "activeness" $\mathscr{E}(\mathbf{X})$, which is the squared sum of the frame-to-frame differential of the motion fragment. We can calculate this measure as $\mathscr{E}(\mathbf{X}) = \sum_n |\mathbf{x}[n] - \mathbf{x}[n-1]|^2$. In particular, we reject concatenating $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$ when $|\ln \mathscr{E}(\mathbf{X}_{k+1}) - \ln \mathscr{E}(\mathbf{X}_k)| > 2.0$.

Parallel shift of body center can be checked by monitoring the difference of the "center bone" position per beat and the "activeness measure" with respect to only the "leg bones". We prohibit parallel shift when the difference of the "center bone" position is large but the small "activeness measure" of the "leg bones" is small, which means the dancer is moving without using his/her legs.

5. EVALUATIONS

5.1 Effect of each musical constraint

We conducted an evaluation to verify which musical constraint (among acoustic feature, musical structure, beat, and chord) was "useful" in automated choreography. The verification was performed with an information theoretical method. That is, the "usefulness" of the music constraint Y in choreography was verified when the choreography became more predictable with the probabilistic model using Y than the model without using Y.

The predictability can be compared with the values of cross-entropy between various combinations of musical constraints. In our situation the cross-entropy can be obtained with

$$H(\mathbf{X}|Y) = -\frac{1}{K}\sum_{k=1}^{K} \log_2 P(\mathbf{X}_k|Y_k), \qquad (9)$$

where $k = 1, \ldots, K$ are the indices of motion fragments in the database. Y_k is the musical constraint at the kth fragment. The predictability is high when the value of cross-entropy is low.

Evaluator	1	2	3	4	5
Accuracy	8 /10	10/10	10/10	10 /10	9 /10

Table 1. Results of subjective evaluation. Five evaluators were asked to choose more natural choreography out of two choreographies: one generated with motion connectivity constraints and other generated without them accompanied by 10 different pieces of music. The number of chosen choreographies which were generated with the proposed method is shown above (Accuracy). Mean accuracy for choosing the motion-connectivity constrained choreography was 0.94 with 95% confidence interval ± 0.11 (Student's t-test).

In our experiment, we prepared 20 different combinations of musical constraints. For every combination of constraints, first we split the motion database into three portions and then trained the five sub-models $P(\mathbf{X}), P(\mathbf{X}|A), P(\mathbf{X}|B), P(\mathbf{X}|C), P(\mathbf{X}|S)$ by using the first portion of the database. Second, we optimized the combination weights λ_m in Eq. (4) using the second portion of the database. The optimized λ_ms are the contribution ratio of musical constraints in predicting the motion fragments. Finally, we calculated the cross-entropy by using the third portion of the database. The motion database consisted of $24,527$ motion fragments accompanied with music. $22,527$ motion fragments were used to train the sub-models, $1,000$ fragments were used to optimize the combination weights, and $1,000$ fragments were used to calculate the cross-entropy.

To obtain the music constraints, the music tracks were first automatically analyzed by using our web service called Songle (http://songle.jp) [10] and then corrected manually using the Songle's error correction interface. Songle leverages various music content analysis techniques to automatically analyze songs publicly available on the web and is open to the public.

The result of the evaluation is shown in Figure 3. We confirmed that the cross-entropy decreased when several musical constraints were taken into account ($H(\mathbf{X}) = 7.643 > H(\mathbf{X}|A, B, C, S) = 7.383$). This indicated that the predictability of dance motion had been increased by using the combination of several music constraints. Optimized results of λ_m $(m = 0, \ldots 4)$ are represented as stacked bars in Figure 3. The length of each color in a bar is calculated by $H \times \frac{\lambda_m}{\sum_{m=0}^{4} \lambda_m}$. Note that we did not use λ_5 for calculating the ratio, since the uniform distribution did not hold information from the motion dataset or the musical constraints. The optimized λ_m indicated that the structure label was the most valuable information for predicting a motion fragment.

5.2 Effect of motion connectivity constraints

We conducted a subjective evaluation to confirm that the motion connectivity constraints were effective in maintaining the naturalness of the choreography. The excerpts of choreography we used in this experiment are uploaded at https://staff.aist.go.jp/s.fukayama/SMC2015/. The screen-

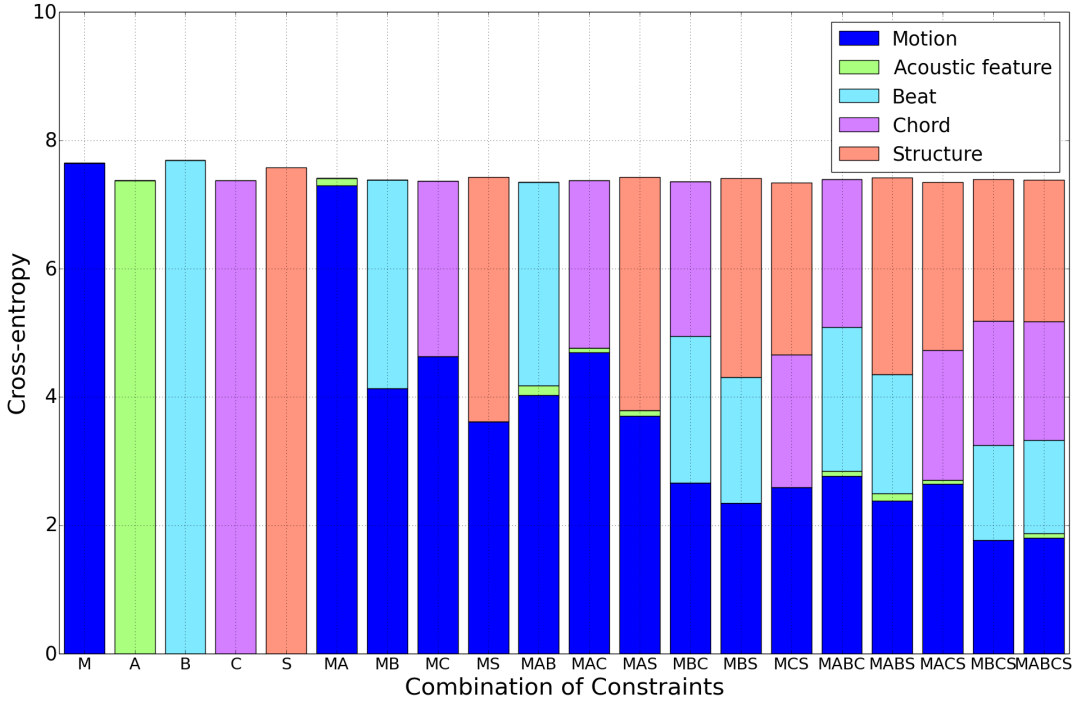


Figure 3. Cross-entropies of motion dataset calculated with probabilistic models with different combinations of music constraints. Height of each bar represents value of cross-entropy, and height of each stacked colored bar indicates ratio of contribution in predicting motion from the music content. Each M, A, B, C and S in the horizontal axis represents motion fragment, acoustic feature, beat, chord and structure, respectively. The cross-entropy decreased by combining several musical constraints ($H(\mathbf{X}) = 7.643 > H(\mathbf{X}|A, B, C, S) = 7.383$), which means the predictability of motion fragments was increased. The contribution of structure label tends to be larger than other musical constraints.

shots of the generated choreographies are shown in Figure. 4.

Ten music excerpts were used in the experiment. The music excerpts were sampled from a song (RWC-MDB-P-2001 No. 07) in the RWC Music Database [11]. For every music excerpt, two different 20-second choreographies were shown to the evaluators. They were asked to choose the one which they felt was more natural. One choreography was generated with motion connectivity constraints, which we proposed in this paper, and the other one was generated without them. The order of showing the two different choreographies was randomized per piece of music.

Five evaluators participated in the experiment. The evaluators did not have any particular knowledge of rules that affect the quality of dancing motions. Therefore, we asked them to intuitively choose a more natural choreography from each pair.

The numbers of chosen choreographies which were generated with the proposed method are shown in Table 1. The evaluators found more than 8 choreographies with motion connectivity constraints to be more natural than the other. The mean accuracy for choosing the motion-connectivity constrained choreography was 0.94 ± 0.11 where 0.11 is the 95% confidence interval by the Student's t-test.

6. DISCUSSION

The objective evaluation in Section 5.1 indicates that combining various musical constraints are useful in automated choreography. The structure label is the most valuable information for predicting the dance motion. Features, such as beat labels and chord labels, also contribute in generating choreography. The subjective evaluation in Section 5.2 indicates that the motion connectivity constraints are effective in maintaining the naturalness of the choreography.

We confirmed that the probabilistic modeling is useful in combining several different constraints driven by different music content analysis modules. It can also generate choreographies by maximizing the probability of the concatenated motion fragments.

To improve the quality of the choreography, we are planning to consider various types of audio features such as spectral flux and chroma vectors. These features can be used to reflect detailed information of music content in the choreography. For instance, spectral flux holds information of acoustic events, such as note onsets, and can be used to make the choreography aligned to the melody notes. Chroma vectors might be useful especially when using delta-chromas to capture the chord changes and when reflecting those changes in the choreography.

Another promising direction for improving the quality is to increase the amount of dance motion data. As our approach is data driven, we expect more variety in generating

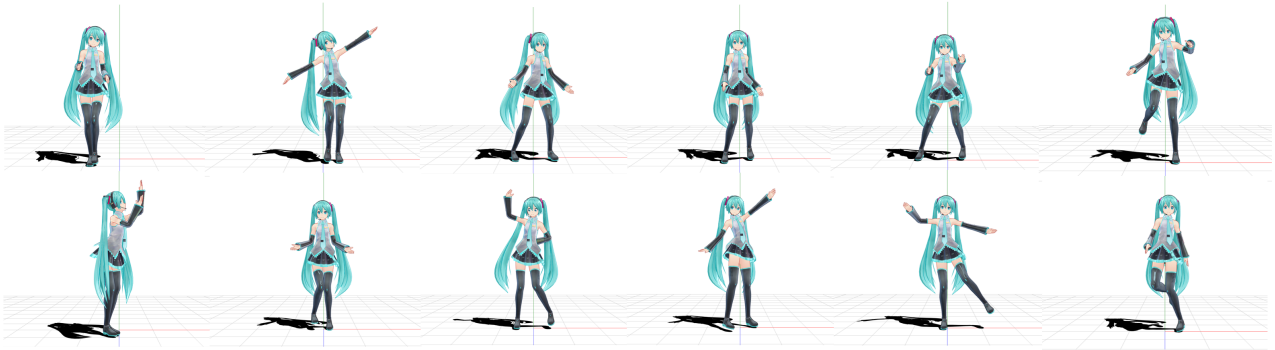


Figure 4. Example of generated choreography with the proposed method.

choreography leveraged by the various dance motions in a larger database. Furthermore, the subjective evaluation can provide more statistical evidence by using a larger dataset.

Finally, we plan to tune the parameters of the probabilistic models. For now, the variance of the Gaussian distribution used as a kernel function is fixed to 1.0, and this value can be tuned with the maximum likelihood framework by using the training data. The value of the variance corresponds to how the motion fragments are roughly categorized as motions with the same character. This may affect the quality of predicting the motion fragment from the music content and therefore needs to be investigated.

7. CONCLUSION

We investigated how to generate choreography automatically by leveraging music content. The proposed method used various music features, not only low-level features such as MFCCs but also features such as structure labels and chord labels to generate choreography. Furthermore, we proposed a set of motion connectivity constraints to ensure the naturalness of the dance motion. These two types of constraints, musical constraints and motion connectivity constraints, were taken into account in a novel probabilistic modeling framework that enabled generating natural music-content driven choreography. Our future work includes more improvements in automated choreography by leveraging more music content analysis techniques that have been considerably developed in the sound and music computing community. The generated sample of our automated choreographies can be found at https://staff.aist.go.jp/s.fukayama/SMC2015/.

Acknowledgement *Hatsune Miku* is copyrighted by Crypton Future Media Inc., and we used it in our research to render the computer graphics of the generated choreography under a PIAPRO license (www.piapro.net). The 3D model of *Hatsune Miku* used in our research was created by koron. This work was supported in part by OngaCREST, CREST, JST.

8. REFERENCES

[1] K. M. Chen, S. T. Shen, and S. D. Prior, "Using music and motion analysis to construct 3D animations and visualisations," *Digital Creativity*, vol. 19, no. 2, 2008.

[2] C. Panagiotakis, A. Holzapfel, D. Michel, and A. A. Argyros, "Beat synchronous dance animation based on visual analysis of human motion and audio analysis of music tempo," in *Proc. ISVC 2013*, 2013, pp. 118–127.

[3] T. Shiratori, A. Nakazawa, and K. Ikeuchi, "Synthesizing dance performance using musical and motion features," in *Proc. ICRA 2006*, 2006, pp. 3654–3659.

[4] J. W. Kim, H. Fouad, J. L. Sibert, and J. K. Hahn, "Perceptually motivated automatic dance motion generation for music," *Computer Animation and Virtual Worlds 2009*, vol. 20, pp. 375–384, 2009.

[5] G. Alankus, A. A. Bayazit, and O. B. Bayazit, *Computer Animation and Virtual Worlds*, no. 16, pp. 259–271, 2005.

[6] R. Fan, S. Xu, and W. Geng, "Example-based automatic music-driven conventional dance motion synthesis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 3, 2012.

[7] M. Lee, L. Lee, and J. Park, "Music similarity-based approach to generating dance motion sequence," *Multimedia Tools and Applications*, vol. 62, no. 3, pp. 895–912, 2013.

[8] F. Ofli, E. Erzin, Y. Yemez, and A. M. Tekalp, "Learn2dance: Learning statistical music-to-dance mappings for choreography synthesis," *IEEE Transactions on Multimedia*, vol. 14, no. 3, 2012.

[9] S. Oore and Y. Akiyama, "Learning to synthesize arm motion to music by example," in *Proc. WSCG 2006*, 2006, pp. 201–208.

[10] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web service for active music listening improved by user contributions," in *Proc. ISMIR 2011*, 2011, pp. 311–316.

[11] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in *Proc. ISMIR 2002*, 2002, pp. 287–288.

Movement Perception in Music Performance – A Mixed Methods Investigation

Jan C. Schacher, Hanna Järveläinen, Christian Strinning
Institute for Computer Music and Sound Technology ICST
Zurich University of the Arts, Zürich, Switzerland
{jan.schacher,hanna.jarvelainen,christian.strinning}@zhdk.ch

Patrick Neff
Department of Psychology
Zurich University
patrick.neff@uzh.ch

ABSTRACT

What are the effects of a musician's movement on the affective impact of experiencing a music performance? How can perceptual, sub-personal and cognitive aspects of music be investigated through experimental processes? This article describes the development of a mixed methods approach that tries to tackle such questions by blending quantitative and qualitative methods with observations and interpretations. Basing the core questions on terms and concepts obtained through a wide survey of literature on musical gesture and movement analysis, the iterative, cyclical advance and extension of a series of experiments is shown, and preliminary conclusions drawn from data and information collected in a pilot study. With the choice of particular canonical pieces from contemporary music, a multi-perspective field of questioning is opened up that provides ample materials and challenges for a process of converging, intertwining and cross-discipline methods development. The resulting interpretation points to significant affective impact of movement in music, yet these insights remain subjective and demand that further and deeper investigations are carried out.

1 Introduction

In this article we explore the potential that a mixed methods approach provides to investigating movement in music performance. The central question of this investigation is if and how those aspects of a musician's playing actions that carry affective potential can be observed, measured and classified. This research is done by blending empirical and systematic, quantitative with qualitative and interpretative analysis methods in a convergent concurrent design [1]. Within this triangulation, the disciplines of Music Analysis, Music Psychology – both in quantitative and qualitative modes – and Music Technology encircle the musician's practice.

The challenge of this methodology is to find ways to bridge between the different disciplinary methods, to make things commensurable and to achieve an equivalence of results necessary for interpretation. One of the aims of this article is to explore the intersections and particularities of the different perspectives, and to ascertain and get a clearer notion of the validity of such a multi-perspective approach.

Copyright: ©2015 Jan C. Schacher 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.

As the literature on 'Musical Gesture' [2] from the past decade and a half has shown, music perception and music making is highly multimodal and therefore needs to be investigated in a cross-disciplinary way. When considering music as a broad category that represents an inherently cultural phenomenon, then the number of involved domains becomes even larger. This fact makes delimiting the field of enquiry essential. The choices in the enquiry discussed here are made from a perspective that focuses mainly on the act of music performance on a fundamental perceptual level rather than through stylistic and cultural categories or through dimensions of signification in musicological terms. The choice is also informed by the necessity to have the topic be grounded in the our own practice, expertise and interests.

2 Background

Before detailing the investigation we are undertaking to elucidate aspects of movement perception in musical performance, it is important to situate our point of view and choice of methods. The standpoint we are taking is informed by complementary but also competing fields; complementary in the overlap of perspectives, competing with regard to validation of results between quantitative and qualitative approaches.

2.1 Mixed Methods Research

Mixed methods research began in the late 1950s in the social, behavioural and human sciences, when a third way was postulated to complement and unite the two dominant strands of quantitative and qualitative methodologies. The central idea is that through triangulation a higher validity of results can be achieved [3]. Through the four types of data-, investigator-, theory- and methodological triangulation, but importantly also through between-methods triangulation three effects arise: convergence, inconsistency, and contradiction [4]. All three effects can provide richer explanations of social phenomena, because creative ways of collecting data need to be developed, thicker and richer data collected, different theories synthesised or integrated, contradictions uncovered, and competing theories validated [5]. Even if a short definition of mixed methods research describes it as "the combination of methodologies in the study of the same phenomenon." [4, p.291], this does not explain how validity can be achieved, if this is indeed the goal. The purpose of mixing methods is understood as a critical step in designing a research project, which – apart from touching on a deeper level of knowledge generation – also has practical implications. Mixed

methods research can serve to: "validate and explicate findings from another approach and produce more comprehensive, internally consistent, and valid findings; provide more elaborated understanding and greater confidence in conclusions; handle threats to validity and gain a fuller and deeper understanding; and provide richer/more meaningful/more useful answers to research questions." [6, p.122] The gain of using this framework is as much a strategic one as it is pragmatic in guiding research activities. These inner and outer effects have to do with the context of the research, thus in social sciences the justification has a different import than in music research. A comprehensive definition that is discipline-agnostic and positions triangulated mixed methods in a relevant way is given by Johnson et al.: "Mixed methods research is the type of research in which a researcher or team of researchers combines elements of qualitative and quantitative research approaches (e.g., use of qualitative and quantitative viewpoints, data collection, analysis, inference techniques) for the broad purposes of breadth and depth of understanding and corroboration." [6]

2.2 Research on Movement in Music

Our investigation spans a range of disciplines that overlap in some way with the domain of music and movement research. The following overview of topics and sources may serve as a map to delineate the field of references we operate in: Musicology approaches on gesture and semiotics of music [7–10], linguistics related to music and gesture [11–13], music psychology dealing with emotion and affective mechanisms [14–16], cognitive sciences delineating embodiment and 'enaction' [17, 18], psychological philosophy of body perception [19, 20], gestalt psychology and ecological embedded-ness [21, 22], human-computer interaction (HCI) for music [23], and systematic as well technical research on machine learning and mapping [24, 25]. The categorisations proposed by Cadoz [26] and Wanderley [27] were milestones in the domain of electroacoustic music related to gesture analysis, in particular the concept of a 'gestural channel' of transmission as well as the categorisation of instrumental gestures into excitation, modification and selection gestures. Within systematic musicology several projects deal with music-technological methods for investigating musical gesture in a multimodal way. The investigations on sound-tracing [28] and gestural music mimicking provide an important reference point [29], and the topical publication on musical gesture by Godøy [30] is particularly relevant. Similarly important is the research by Camurri on expressive gesture including the development of the Eyesweb platform [31, 32]. The tools developed by Gillian in Belfast [33] and Princeton [34] provide a useful basis for the software developments in our own project.

The field of dance research needs to be taken into account as well, where the shared fundamentals of Laban movement analysis [35] connect to music performance, even if specific projects in dance deal with other issues, such as notation, transmission and documentation [36, 37].

3 Terminological Explorations

Based on these background topics, we set out to make a map related to the term 'musical gesture'. The main purpose of this analysis was to clarify the standpoint from which our own investigation is led. By taking definitions from the aforementioned literature [9, 26, 30] and adding specific references [38, 39], the purpose is to render visible the relationships of aspects of 'gesture' between composer, performer and listener, looking on the one hand at the presence of traces of embodied thinking in the score/text and on the other hand at the effect of the performer's movement on the audience.

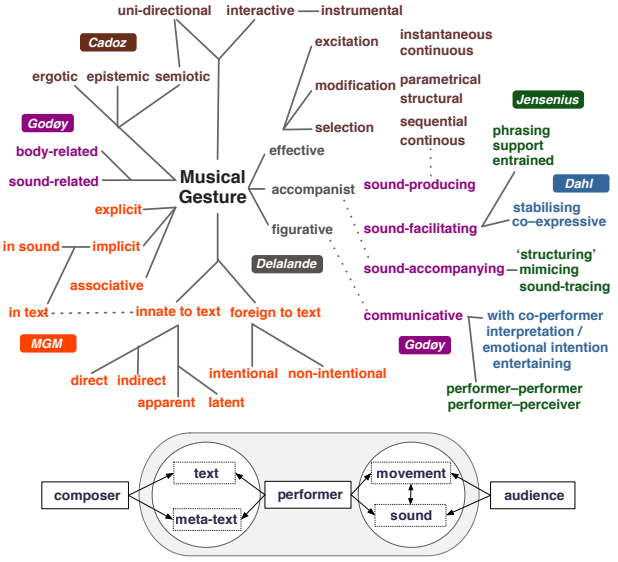


Figure 1. A map of terminologies on Musical Gesture and a relational schema of the music performance actors and mediating elements. Dashed lines show equivalence.

In the lower part of Figure 1 the relational schema of the actors in a music performance situation is sketched out. Not all musical styles manifests all the elements in the same way, but all elements are always present. Interesting for our analysis is on the one hand the tripartition between composer, performer and audience, and in a particular the equivalence of text (i.e., score), meta-text (i.e., common practices) with sound and movement as carriers of meaning. The schema should not be read as a directional flow going from left to right, but rather make visible the interdependence of all the elements.

The upper part of Figure 1 maps out aspects of movement and gesture, on the one hand as categorised by key authors in the field and on the other hand by putting them in relation with our particular standpoint (see left-lower section in orange marked MGM). Where in the left half of the tree the categories show fundamental relationships between the body (Godøy) [30], the instrument (Cadoz) [26], sound and text (MGM), the right part unfolds aspects of music performance relevant for musician and audience (Delalande [9], Godøy, Jensenius [38], Dahl [39]). What is interesting are the different weights generated by the score (text) and the music-playing act. In this regard the categories proposed by Cadoz of 'ergotic, epistemic and semiotic' gestures or the sound-facilitating aspects proposed by Jensenius apply only in a limited sense, in particular when considering that

a composer's own 'gesturality' is transported in the score – along or embedded within the instructions for the instrumentalist.

The map shows how key authors dealing with musical gesture take different perspectives, emphasising either the affordances (action spaces) or the perceptions of gestural actions. This reveals to us that the discourse on musical gesture has taken into account neither the role of the composer nor that of the text (score) in constituting or informing the bodily domain. An in-depth discussion of the terminological implications in relation to 'Pression' can be found in [40]. [1]

4 Designing the Study: What Music, Which Piece, Which Aspects?

The choice of musical style and specific piece for such an investigation determines the types of results or interpretations that can be gained. With a focus on the performance moment and with a standpoint that is oriented towards composer and performer, rather than the audience, our selection of a piece has to fulfil several criteria. Even though for the larger research project within which this work is done electro-acoustic and live-electronic music is central, looking at a more traditional piece has some advantages. The problem with perceiving movement in electronic music should be obvious, since some of the actions that produce sound are imperceptible in physical action since they are mediated through technical means. In a parallel task, live-electronic, interactive and technological music is indeed explored, but these will only be touched upon very briefly in the qualitative section.

For this study we chose the seminal piece for solo violoncello 'Pression' by Helmut Lachenmann from 1970 (in the 2010 version) performed for us repeatedly by cellist Ellen Fallowfield. This piece represents a important exemplar of what Lachenmann calls 'Musique Concrète Instrumentale'. The entire score carries action-notation mixed with standard notation; it describes the movements and extended playing-techniques on the instrument rather than the sounding result. The idiom of the piece is based on extended sounds of the instrument, which – together with the tightly choreographed movements – makes it useful for analysis both from a textual, music-analytical as well as point of view focusing on the performer's physicality [41]. As the title suggests, this piece explores the aspect of pressure, both of the bow and the hands on the instrument. This ties in well with the one playing aspect we are investigating in our mixed qualitative and quantitative method, namely that of *effort*.

When looking at the central question of this investigation about observing, measuring and classifying affective potential carrying aspects of music performance, the question is where to start. Since the focus lies on the movement and gestures of the performer rather than the sound and music, a look to a neighbouring field may provide the answer. In the movement analysis by Laban the term 'effort' provides the central pivot for describing corporeal performance: "words and ... music are both apt to overshadow the truth of this effort display as it becomes apparent through the performer's bodily actions. ... Every human movement is indissolubly linked with an effort, which is, indeed, its origin and inner aspect. Effort and its resulting action ... are always present in any bodily movement; otherwise they could not be perceived by others." [35, p.9/21] Without adopting all the subtleties of the subcategories in the Laban effort concept, i.e., the aspects of Weight, Time, Space and Flow, the fundamental idea that all *affect* in a perceiver is generated by the resonance with the *effort* by the performer, in our opinion holds true for musicians as well.

The approach to designing this study is exploratory and done in a convergent concurrent manner [1]. Although the hypothesis and question is clear, when we began this enquiry we didn't have a definitive plan and methods all lined up. By proceeding with iterative steps that implement one part of the method after another, and by evaluating the results at each step, we are capable of adjusting and refining not just methods but also the scope and the domain investigated. This is an ongoing process which is not finished, even if we have reached a point where we have preliminary results and reflections to draw first conclusions from.

With the choice of musical material made, the two complementary investigations are carried out. They are done in parallel, since the shifting focus in one method leads to the adaptation of elements in the other. The iteration of several prototype *task-and-survey* modules establishes the certainty that the method is sound and we can proceed. The following two sections describe respectively the qualitative and the quantitative activities of what we merely consider a pilot study, which has by no means uncovered all we need to know.

5 Quantitative Experiment

Continuous self-report methods are widely used in evaluation of emotional response to music [42]. A pilot experiment was conducted, in which subjects rated *perceived effort* in a video recording of a performance of 'Pression'. The performance was assessed separately, based on either the audio or the video. The goal was to find out how similar or different the ratings would be based on audio or video, and which modality dominates the perception and what the contributing factors are. In addition to the conceptual and terminological considerations laid out earlier, the choice of the measured attribute evolved through preliminary iterations. Attributes included musical intensity, perceived tension, musical tension, amount of emotion, aesthetic response, emotion expressed, arousal, and valence. In a preliminary iteration of the present experiment, subjects assessed intensity, but this proved to be problematic, since it was too easily associated with loudness. We decided that the chosen attribute needed to be perceivable both in the auditory and the visual modality. Moreover, the attribute had to be associated with the gestures and movements of the performer and be as little as possible associated with valence (pleasantness) as possible. Based on these insights, the choice of the attribute of *perceived ef-*

[1] See also `http://mgm.zhdk.ch/mindmap/mindmap.html` for an interactive version of the map.

fort seemed the best fit.

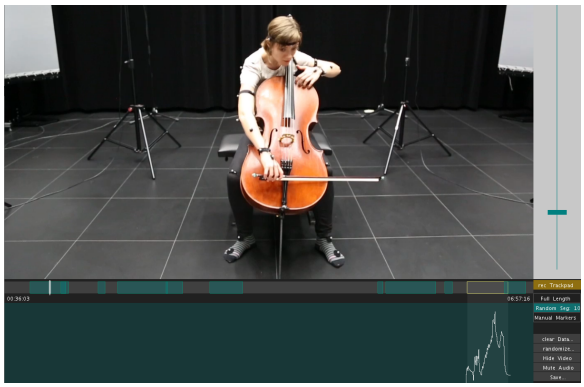


Figure 2. View of the survey software: segments on the timeline, virtual slider and time-series data for segment 10.

The subjective ratings about this were given by pressing on a force sensor while watching or listening to the recorded performance in a custom software (see Fig. 2 and refer to [43] for more details on the technical elements of this investigation). Subjects were instructed to press harder with increasing perceived effort. The pressing force was recorded at 10 ms intervals and mapped logarithmically to a numeric scale between 0 and 100.

Eleven segments of 4–40 seconds duration were selected from the complete performance, representing the various sound materials and playing techniques present in the piece. The presentation mode was treated as a within-subjects design; all subjects rated all 11 segments for both the video and audio conditions. The presentation order of the segments was randomised, and the order of the audio and video trials was balanced across subjects. Half of the subjects started with the audio and the other half with the video condition. N = 6 subjects took part in the pilot experiment, including the present authors. All except one are trained musicians.

The result of the measurement was a non-stationary, dependent time series for each subject. Median time series were computed across the subjects for each segment and presentation mode, as seen for selected segments in Fig. 3.

Results of the audio and video conditions were surprisingly closely related. The general profiles of the median time series for audio and video were similar in all segments but one. Furthermore, the audio and video medians were very close to each other in five segments. In the remaining segments the audio ratings were either higher than the video (segments 4, 5, and 6) or vice versa (segments 1 and 2).

The mean audio and video ratings across segments were positively correlated ($\rho = 0.81$). The range of the video ratings was narrower compared to the audio ratings (see Fig. 3). The results suggest that the perception through the two modalities is contradictory, when soft audio is combined with movement in the video, as in segments 1 and 2, or if loud and/or unpleasant audio is combined with calm bowing movements of the left hand, as is the case in segments 4 and 5 (see middle of Fig. 3). A future goal is to find out what the total percept would be in these mismatched situations.

A further observation is that in the audio ratings, at the end of the piece, effort is perceived only as long as there is sound. In the video condition, similar or even increased effort was perceived until the player put the bow down, released the attention or tension and thus finished her performance. In this performance, as is often the case, the physical release occurred several seconds after the sound had already died down.

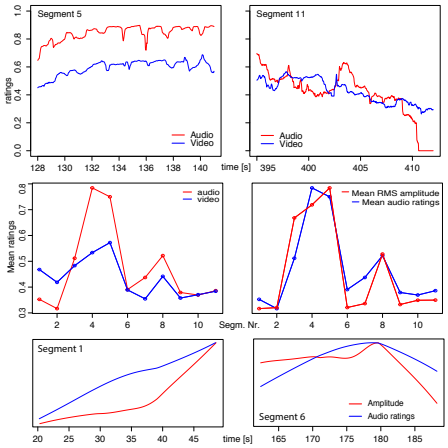


Figure 3. Top Row: Median time series for segments 5 and 11 in the effort measurement experiment. Middle Row: Mean audio and video ratings over all segments. Mean audio amplitude and mean audio ratings over all segments, linearly scaled to equal range. Bottom Row: Trends in audio amplitude and audio ratings

Audio RMS envelopes were computed for each segment using the MIR Toolbox for Matlab [44], and Quantity of Motion (QoM) was computed for the video segments. Significant positive correlations were found between the objective characteristics and ratings: $\rho = 0.93$ between mean audio amplitude and mean audio ratings and $\rho = 0.68$ between QoM and mean video ratings. Moreover, audio amplitude correlated highly with video ratings ($\rho = 0.87$) and QoM with audio ratings ($\rho = 0.84$), which is further evidence that cases of extreme mismatch between the modalities were rare.

Trends were extracted from both the audio ratings and audio amplitudes, as shown in Fig. 3, bottom row. In 7 of the 11 segments they are similar, indicating that increased loudness is followed by an increase in perceived effort in the performance.

This pilot study suggests a hypothesis that effort is perceived similarly based on the auditory and visual aspects of a music performance. Future plans include extension and revision of the experiment from a pilot to a larger scale study with 'fresh' test subjects. At present, the subjects who took part in the experiment are already familiar with the previous stages of development and therefore know the performance too well. The experiment will also include the combined audio-and-video condition in addition to audio or video conditions alone. Preliminary experiments were already made for this third condition, but since measurements were done using a slider instead of a touch sensor, the results cannot be compared at present. The analysis will be extended from the descriptive level to an inferential model, and more auditory features will be considered

in addition to amplitude envelope, such as spectrum, onset detection, tempo, attack time, brightness, roughness, and pitch. Motion capture data is now available as well from the same recording, including acceleration data of the different points; this will be taken into consideration instead of the simpler QoM measure. A goal of the extended study will be to explore through analysis of individual segments, what causes the observed differences between audio and video ratings.

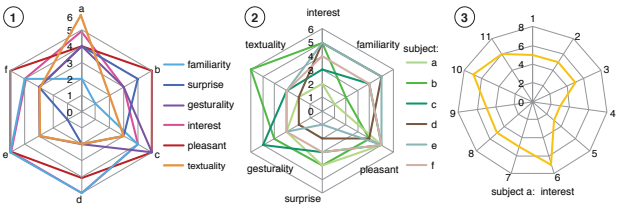
6 Qualitative Methods

After the measurement-based part of the study, let us now explore the complementary part that deals with subjective assessment of the *identical* musical performance materials. The question of how to retrieve individual qualitative data concerning perception and performance of music performance, possibly even abstract electroacoustic gestural music, poses a challenge since neither the form nor the language of the subject matter are directly accessible or established by convention. As laid out earlier, in order to reach a validated selection of terms in a similar iterative fashion as with the *perceived effort* in the quantitative track, we base the concepts and aspects for the subjective and qualitative enquiry on the literature accumulated. In addition to the more traditional approaches that make use of a general gesture terminology coming from linguistics [13], or describe the gestural morphology in the actual context [30], be it on a phenomenological/epistemic level [2, 45], or by focusing on functional aspects [38], we add terms from the relational schema of music performance actors (see Fig. 1) and the score–action dichotomy that is present in the piece by Lachenmann. In addition to the musical categories we introduce additional general impression and preference ratings with a compilation of items from [16]. The aim of this qualitative approach is to blend and apply the terminologies and concepts with a questionnaire related to the 'Pression' study.

6.1 Survey

As outlined above, several cycles of *task-and-survey* tandem modules were carried out. The terminologies derived from literature and our conceptual analysis were complemented with generic terms like 'gesturality' or 'expressivity' and compiled into a questionnaire and set up alongside the continuous self-report tasks of the quantitative track. Each subject, after completing the entire *perceived effort* task, filled out a questionnaire for each segment they had previously rated. Participants were asked to categorise the same 'Pression' segments according to the given terminologies via multiple choice and by adding comments about their choice for each segment. The terms are organised in the following categories:

– General: interest, familiarity, pleasantness, surprise, 'gesturality', 'textuality'.
– Phenomenological : 'ergotic', epistemic, semiotic [2].
– Movement type: trajectory-, pattern-, force-based.
– Functional: communicative, sound-producing, sound-facilitating, sound-accompanying [30].
– Musically supporting: melody, harmony/musical structure, timbre, sound level, rhythm, tempo.
– Morphological: impulsive, sustained, iterative [46, 47].



person	interest	familiarity	pleasant	surprise	gesturality	textuality	semantic	movement	functional	helping	morphological
a	3	4	1	1	2	4	2	2	2	3	2
b	3	5	1	2	5	1	3	1	2	4	2
c	2	5	3	1	4	6	1	2	3	2	3
d	5	4	4	2	1	1	1	2	2	3	2
e	1	5	3	1	1	5	1	2	2	3	1
f	2	5	2	2	1	4	1	1	2	2	3
avg	2.67	4.67	2.33	1.5	2.33	3.5	1.5	1.67	2.17	2.83	2.17
std dev	1.87	0.27	1.47	0.3	3.07	4.3	0.7	0.27	0.17	0.57	0.57
min	1	4	1	1	1	1	1	1	2	2	1
max	5	5	4	2	5	6	3	2	3	4	3
median	3	5	3	2	2	4	1	2	2	3	2
mode	3	5	1	1	1	4	1	2	2	3	2
agreement	T	T	F	F	F	T	T	T	T	T	T

Figure 4. Ratings of a segment's categories by all subjects (1), all subjects by category (2), or a single participant judging a single quality across all segments (3). The table shows all ratings from segment five, familiarity is the overall mode agreed on: it is the only section in the piece with a 'normal' bowing tone.

This first iterations, dealing with 'Récitation 1' by Georges Aperghis, as performed for us by the renowned singer Donatienne Michel-Dansac, served as a feasibility test for understanding and checking acceptance of the terminologies as well as for validating the actual questionnaire form.

The feedback and experience from the previous round was integrated into a digital form; the survey was condensed to a single- or forced-choice format for every categorical subsystem with an added description of the concepts as quoted from the sources. Taken together, these iterations demonstrated a certain *interrater-reliability* as most of the segments were categorised similarly by all participants. In some of the points of the questionnaire divergent individual interpretations and ratings remained, which need to be conserved for further analysis, unless they can be attributed to methodological issues. In order to rule out remaining confounding influences by the selection of categories as well as the focus set by the rater, in a third iteration of the questionnaire, we asked the participants to provide specific information about those elements within each segment that had led to the actual categorisation. At this stage, the analysis of the data is done on the visual interpretation of basic descriptive statistics and plots (see Fig. 4). This is due to the limited number of participants, and the fact that the same subjects have done several iterations and are already too attuned to the process. The result of this analysis is two-fold: on the one hand each segment obtains a rating for all given categories and the most commonly named mode wins, on the other hand the deviations from this norm prove to be what raises most questions, showing the inter-individual difference in appreciation and interpretation of the different playing techniques observed.

For the next iteration of the study we will synthesise the insights into an online questionnaire that integrates

closed/forced choice items as well as open formats and directly embeds audio-visual materials. The participant will be able to propose their own categories, and to comment on categorisations and the entire survey. The specific language of the category systems stemming both from literature and performance practice needs to be translated into more accessible, everyday language to be suitable for non-expert, non-musician participants.

6.2 Observation and Evaluation in the Field

In parallel to the tandem method described here, we are extending the mixed methods with further purely qualitative approaches. The main goal is to evaluate the given concepts and insights through observation of concrete artistic processes of creation, rehearsal and performance with audience. This extended method provides a blend of a Grounded Theory approach [48] with more general, ethnographical and social-science approaches of qualitative research [49]. By accompanying artistic work through musical rehearsals and performance in a first phase, and by collecting interview data from the performers and composers in a second phase, the overarching research-questions of the project are approached from an angle that is a step more removed from empirical data analysis. After the observation, the collection of materials is done with the artist in semi-structured interview with narrative aspects [49]. The notes, interviews and other traces will be textually analysed and condensed into reports as well as thematic layouts which are then discussed and re-synchronised with the other research tracks.

7 Discussion

The final step remaining in a mixed methods investigation is the blending of the results obtained in the two tracks in a triangulated interpretation that is appropriate for the research question. Although this might seem to be a final step in the process, in fact, the cross-contamination of the two perspectives already occurred throughout the iterative development cycles. A central anchor for the research was given by the fact that even though both methods collect data in their separate ways, the data-gathering occur back-to back, and both rely on subjective, i.e., personal opinions and derive the categories from a common model. This is particularly important because the objects of investigation are perceptual qualities, rather than physical or physiological invariants [50].

When comparing the test-segments for significant deviations from a consensual base-line given by the subjects, several aspects come to the attention. Since 'Pression' is a piece for violoncello, almost all of the body-parts are constrained in their placement and kept under tight control in relationship to the instrument and bow, and particularly by the unusual, extended playing techniques that constitute this piece. The only exception is the head, which has relative freedom of movement, except where damping the strings with the chin is demanded by the composer. In segment 6 (legno saltando, bow below the bridge, top of p.3 in 2010 score) the divergence between the effort ratings in audio and video (see bottom right of Fig. 3) and the agreement in four out of five categorisations (phenomenological, functional, musically supporting, morphological), as well as a comment saying that this is "a very gestural segment", indicate a positive affective impact, which is stronger than in other segments. Even when looking at the point-light display of motion-capture data of the same performance the salient feature, apart from the bouncing bow, is the way he player emphasises the light and springy movements of the bow and the bouncing sounds with similar movements of the head. In contrast, when observing the data obtained from the fifth segment (the second half of *Largo Feroce, am Saitenhalter gepresst*, bottom of p.2 in 2010 score), it is already visible from the data that this section contains a high effort level (perceived more in the auditive than the visual domain, see the top left of Fig. 3) and subjectively generates low levels of pleasantness, surprise, and interest, and a remote text/score relationships. The data from both the qualitative and the quantitative tracks confirm a uniform opinion by all the test-subjects. When listening to this forceful scratching section, which is the hallmark of 'Pression', the negative affective impact of this section becomes evident.

After attempting an interpretation of the preliminary data and information gathered in this process, a higher level analysis of the research process is needed. The study described here is not the only part of the investigation: the developments of the method and the embedding of the different layers into a larger fabric form integral part of the process. By looking beyond the iterative cycle of *task-and-survey* modules, it is evident that they need to be framed by the terminological definitions and classifications, and observations and interviews about artistic processes. We believe that it is within this wider context that the interpretations based on our mixed perspectives will ultimately bring their best results. However, much needs still to be done. The quantitative track will proceed through a new full cycle of a *task-and-survey* experiment with fresh subjects, basing its analysis on more reference-data. The qualitative strand needs to solidify the observational, ethnographical process and bring together the insights from both the structured, systematic studies and the grounded theory approach that accompanies artistic creation and development processes. Finally, the synthesising, interpretation part of the method needs to find more ways of validation with richer data-sets and complementary analysis methods both in the qualitative, subjective and quantitative, empirical domains.

8 Conclusion

In this article we describe the methods development process of an investigation into cross-modal perception of key musical performance aspects. Starting from a perspective that is based on an embodied, ecological perspective of music perception, a blend of methods is sketched out and iteratively tested that mixes qualitative and quantitative methods. The selection of musical material that serves as testing ground is crucial. The choice of European contemporary music is motivated by the need to work with music that is less bound to traditional harmony and melody and

symbolic music analysis, and more conducive to perception where the musician's performance action and the resulting phenomenal sound-world come to the foreground. The choice of aspects to investigate is equally important, on the one hand the *perceived effort* in continuous self-report, and on the other hand the subjective opinions about *six dominant categories* stemming from literature on musical gesture.

The quantitative experiment suggests that effort is indeed a meaningful attribute to measure the perception of music performance. At this point the study is still inconclusive and serves mainly to form hypothesis and collect first experiences in a non-tonal context. A larger study will further test the hypothesis that effort is perceived in both auditory and visual modalities and how they relate. Exploring and validating mixed methods research for music perception investigations is the central goal of this article. The interpretation that is carried out by blending the results from the two domains clearly shows significant effects of movement on affective impact of music performance, and demonstrates some of the ways this occurs. Thus far the project raises questions that can only be approached using a mixed methodology: Can effort be a structure-defining attribute that could serve both as an analytic and compositional device, i.e., effort-based music analysis and effort-based composition? How is the notion of effort already part of an artistic process, be it in composition, performance and music listening?

The hybrid approach presented evidently generates results that remain subjective and tied to the specifically selected study-object. At the same time it shows a path forward that is truly multi-perspective and has the potential to unlock those elements of perception with affective power that are situated in the highly multi-modal and complex enfolded 'thing' we call Music.

Acknowledgments

This investigation is carried out in the research project 'Motion Gesture Music' at the Institute for Computer Music and Sound Technology of the Zurich University of the Arts and is funded by the Swiss National Science Foundation Grant No. 100016_149345.

9 References

[1] G. Guest, "Describing mixed methods research an alternative to typologies," *Journal of Mixed Methods Research*, vol. 7, no. 2, pp. 141–151, 2012.

[2] C. Cadoz, M. M. Wanderley *et al.*, "Gesture-music," in *Trends in Gestural Control of Music*. Paris: Ircam, Centre Pompidou, 2000, pp. 71–94.

[3] E. Webb and K. E. Weick, "Unobtrusive measures in organizational theory: A reminder," *Administrative Science Quarterly*, pp. 650–659, 1979.

[4] N. K. Denzin, *The research act: A theoretical introduction to research methods*. New York: McGraw-Hill, 1978.

[5] T. D. Jick, "Mixing qualitative and quantitative methods: Triangulation in action," *Administrative science quarterly*, pp. 602–611, 1979.

[6] R. B. Johnson, A. J. Onwuegbuzie, and L. A. Turner, "Toward a definition of mixed methods research," *Journal of mixed methods research*, vol. 1, no. 2, pp. 112–133, 2007.

[7] R. Hatten, *Interpreting Musical Gestures, Topics and Tropes: Mozart, Beethoven, Schubert*. Indiana University Press, Bloomington, IN, 2004.

[8] A. Gritten and E. King, *Music and Gesture*. Ashgate Publishing, 2006.

[9] F. Delalande, M. Formosa, M. Frémiot, P. Gobin, P. Malbosc, J. Mandelbrojt, and E. Pedler, *Les Unités Sémiotiques Temporelles - Éléments nouveaux d'analyse musicale*. Marseille: Édition MIM, 1996.

[10] M. Leman, *Embodied Music Cognition and Mediation Technology*. Mit Press, 2007.

[11] D. Lidov, *Is language a music?: Writings on musical form and signification*. Indiana University Press, 2005.

[12] D. McNeill, *Hand and Mind*. Chicago University Press, 1992.

[13] A. Kendon, *Gesture: Visible action as utterance*. Cambridge University Press, 2004.

[14] P. N. Juslin and J. Sloboda, *Handbook of Music and Emotion: Theory, Research, Applications*. Oxford University Press, 2010.

[15] E. F. Clarke, *Ways of Listening – An ecological Approach to Perception of Musical Meaning*. Oxford University Press, 2005.

[16] B. W. Vines, C. L. Krumhansl, M. M. Wanderley, I. M. Dalca, and D. J. Levitin, "Music to my eyes: Cross-modal interactions in the perception of emotions in musical performance," *Cognition*, no. 118, pp. 157–170, 2011.

[17] F. J. Varela, E. T. Thompson, and E. Rosch, *The Embodied Mind: Cognitive Science and Human Experience*. Cambridge Mass: MIT Press, 1991.

[18] R. I. Godøy, "Gestural-Sonorous Objects: Embodied Extensions of Schaeffer's Conceptual Apparatus," *Organised Sound*, vol. 11, no. 2, pp. 149–157, 2006.

[19] S. Gallagher, *How the Body Shapes the Mind*. Oxford: Clarendon Press, 2005.

[20] D. Legrand, "Pre-Reflective Self-Consciousness: On Being Bodily in the World," *Janus Head*, vol. 9, no. 2, pp. 493–519, 2007.

[21] D. Katz, *Gestalt Psychology, Its Nature and Significance*. New York: The Ronald Press Co., 1950.

[22] J. J. Gibson, *The Ecological Approach to Visual Perception*. Lawrence Erlbaum, 1986.

[23] P. Dourish, "Embodied Interaction: Exploring the Foundations of a New Approach to HCI," *Unpublished paper, on-line: http://www.ics.uci.edu/˜jpd/ publications/misc/embodied.pdf*, 1999.

[24] D. Arfib, J. M. Couturier, L. Kessous, and V. Verfaille, "Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces," *Organised Sound*, vol. 7, pp. 127–144, 7 2002.

[25] B. Caramiaux and A. Tanaka, "Machine learning of musical gestures," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2013)*, Seoul, South Korea, 2013.

[26] C. Cadoz, M. M. Wanderley *et al.*, "Gesture-Music," *Trends in Gestural Control of Music*, vol. 12, pp. 71–94, 2000.

[27] M. M. Wanderley and M. Battier, *Trends in Gestural Control of Music*. Paris, IRCAM, Centre Pompidou, 2000.

[28] R. I. Godøy, E. Haga, and A. R. Jensenius, "Exploring music-related gestures by sound-tracing: A preliminary study," in *Proc. of the 2nd Intl. Symposium on Gesture Interfaces for Multimedia Systems.*, 2006.

[29] E. Haga, "Correspondences between Music and Body Movement," Ph.D. dissertation, University of Oslo, 2008.

[30] R. I. Godøy and M. Leman, *Musical Gestures: Sound Movement and Meaning*. New York: Routledge, 2010.

[31] A. Camurri and P. Ferrentino, "Interactive environments for music and multimedia," *Multimedia Systems*, vol. 7, no. 1, pp. 32–47, 1999.

[32] A. Camurri, S. Hashimoto, M. Ricchetti, A. Ricci, K. Suzuki, R. Trocca, and G. Volpe, "Eyesweb: Toward gesture and affect recognition in interactive dance and music systems," *Computer Music Journal*, vol. 24, no. 1, pp. 57–69, Spring 2000.

[33] N. Gillian, R. B. Knapp, and S. O'Modhrain, "The SARC EyesWeb Catalog: A Pattern Recognition Toolbox For Musician Computer Interaction," in *Proc. of the Conference on New Interfaces for Musical Expression (NIME09)*, Pittsburgh, USA, 2009.

[34] N. Gillian and R. Fiebrink, "A Hands-On Workshop on Gesture Recognition and Machine Learning for Real Time Musical Interaction," in *Proc. of the Conference on New Interfaces for Musical Expression*, Ann Arbour, Michigan, 2012.

[35] R. Laban, *The Mastery of Movement*, revised 4. ed. Alton, Hamphire, UK: Dance Books Ltd., 1950 (1980/2011).

[36] B. Bermúdez-Pascual, "(Capturing) intention: The life of an interdisciplinary research project," *International Journal of Performance Arts & Digital Media*, vol. 9, no. 1, pp. 61–81, 2013.

[37] C. Fernandez, "The TKB Project: Creative Technologies for Performance Composition, Analysis and Documentation," in *ECLAP 2013, LNCS 7990*, P. Nesi and R. Santucci, Eds. Springer Verlag, 2013.

[38] A. Jensenius, M. Wanderley, R. 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.

[39] S. Dahl and A. Friberg, "Visual perception of expressiveness in musicians' body movements," *Music Perception*, no. 24, 2007.

[40] C. Strinning and G. Toro-Pérez, "Vor der Erstarrung - Bewegungsstrukturen in Helmut Lachenmanns Pression," *Dissonanz = Dissonance*, forthcoming 2015.

[41] T. Orning, "Pression Revised, Anatomy of Sound, Notated Energy, and Performance Practice," in *Sound & Score, Essays on Sound, Score and Notation*, P. de Assis, W. Brooks, and K. Coessens, Eds. Leuven University Press, 2013, pp. 94–109.

[42] E. Schubert, "Continuous self-report methods," in *Handbook of Music and Emotion: Theory, Research, Applications*, P. N. Juslin and J. Sloboda, Eds. Oxford University Press, 2010.

[43] J. C. Schacher, "Music Means Movement - Musings on Methods of Movement Analysis in Music," in *Proceedings of the 2nd International Workshop on Movement and Computing (MOCO'15)*, Vancouver, Canada, August 14–15 2015.

[44] O. Lartillot and P. Toiviainen, "A Matlab Toolbox for Musical Feature Extraction from Audio," in *International Conference on Digital Audio Effects, DAFx*, 2007, pp. 237–244.

[45] I. Choi, "Cognitive Engineering of Gestural Primitives for Multi-modal Interaction in a Virtual Environment," in *International Conference on Systems, Man, and Cybernetics*, vol. 2. IEEE, 1998, pp. 1101–1106.

[46] P. Schaeffer, *Traité des Objets Musicaux*. Paris: Editions du Seuil, 1966.

[47] D. Smalley, "Spectromorphology: explaining soundshapes," *Organised sound*, vol. 2, no. 2, pp. 107–126, 1997.

[48] B. G. Glaser and A. L. Strauss, *The discovery of grounded theory: Strategies for qualitative research*. Aldine de Gruyter, 1967.

[49] U. Flick, *An introduction to qualitative research*. Sage, 2009.

[50] M. B. Küssner and B. Caramiaux, "Motor invariants in gestural responses to music," in *Proceedings of the International Conference on the Multimodal Experience of Music*, Sheffield, 23-25 March 2015.

Psychoacoustic impact assessment of smoothed AM/FM resonance signals

Antonio José Homsi Goulart
Computer Science Department
University of São Paulo
São Paulo - Brazil
antonio.goulart@usp.br

Joseph Timoney, Victor Lazzarini
Computer Science Deparment
Maynooth University
Maynooth, Co. Kildare - Ireland
joseph.timoney@nuim.ie
victor.lazzarini@nuim.ie

Marcelo Queiroz
Computer Science Department
University of São Paulo
São Paulo - Brazil
mqz@ime.usp.br

ABSTRACT

In this work we decompose analog musical resonant waveforms into their instantaneous frequency and amplitude envelope, and then smooth these estimations before resynthesis. Signals with different amounts of resonance were analysed, and different types and lengths were tested for the smoothers. Experiments were carried out with amplitude smoothing only, frequency smoothing only, and simultaneous smoothing of amplitude and frequency signals. The psychoacoustic impacts were evaluated from the point of view of dynamic brightness, tristimulus and spectrum irregularity. We draw conclusions relating the parameters explored and the results, which match with the sounds produced with the technique.

1. INTRODUCTION

Resonance is the tendency of a system to vibrate sympathetically at a particular frequency in response to energy induced at that frequency [1]. Resonances play an important role in computer music and a number of techniques have been proposed for their synthesis, including FOF [2], VOSIM [3], ModFM [4] and Phase Distortion [5]. Frequency modulation [6] methods are interesting because they can offer a flexible yet computationally inexpensive solution. However, care has to be taken in selecting appropriate modulation functions if the result is not to sound lifeless in comparison to their analog synthesizer counterparts. Instruments such as MiniMoog, Korg MS-20, TB-303 [7] present unique, readily identifiable resonant sounds. Transporting the compelling nature of the analog sound to the digital domain is an interesting problem, particularly if we want to preserve the feel of the sound, but not simply mimic it.

This work is a first investigation on the potential of using an AM/FM signal decomposition followed by smoothing and resynthesis for the modeling and synthesis of resonance signals. A link can be made with the modulation synthesis by decomposing a discretized analog signal into an AM/FM representation using an analytic signal approach [8]. This representation captures everything in two descriptors, the changing envelope (AM) over time and the frequency excursion around the fundamental (FM). These quantities display an 'average' behaviour for the collection of components in the signal at a particular time instant [9].

Albeit using different tools than ours, a similar work regarding assessment of analysis followed by modification of the parameters and resynthesis was performed in [10]. Acoustic musical instruments recordings were decomposed with Fourier analysis and different modifications were tested within an additive synthesis context for the comparison of the original and resynthesized sounds. Another work [11] explored AM/FM decomposition of musical instruments using energy separation, in order to analyse vibrato/tremolo and determine synthesis parameters for an excitation/filter model. In [12] the reverberation on voice recordings was analysed in terms of its impact on an AM/FM decomposition.

The extent of the modulations' variations is interesting regarding its relationship to the perceived sound, and it is a subject that has not received much attention in the literature. However, if we have a good perceptual intuition about these signals we should be able to design digital modulation-driven resonance generators that sound more exciting. A straightforward approach to achieve this is to form the decomposition of suitably chosen analog generated resonance signals and then manipulate their AM and FM quantities and observe the outcome in a series of controlled experiments.

Also, we highlight a different kind of approach for the processing of musically-interesting sounds, which is not as widely used in the computer music literature as the Fourier analysis and additive synthesis approach [13]. Analysis / resynthesis with AM/FM decomposition can also be seen as a different paradigm where we consider the sound signal in terms of a single harmonic oscillator model, with varying instantaneous amplitude and frequency. Whereas in the Fourier paradigm the signal is viewed as the resultant of superimposed oscillators, in the AM/FM model we might think of a mass-spring system with varying mass (e.g. where the body suspended from a spring is a liquid container with controllable inlet and outlet). We are then able to take advantage of the compactness of dealing with only a couple of signals to apply our desired modifications/adaptations to the sounds.

Copyright: ©2015 Antonio José Homsi Goulart 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.

We start the paper by talking about the signals we used and their decomposition based on amplitude envelope and instantaneous frequency estimation, followed by a discussion regarding the process of smoothing and resynthesis. This is complemented by an explanation of the psychoacoustic metrics and the results we obtained, and our conclusions. All code was written in Octave and is available [1] alongside a comprehensive set of sound examples.

2. ANALOG SYNTHESIZER RESONANCES

Analog synthesizers are usually based on the subtractive synthesis model [7], where a raw rich excitation signal is the source for a modifier, typically a set of filters that will impose their characteristics and tailor the sound.

The TB-303 is an electronic bass synthesizer that was introduced in the 80s, being heavily explored by dance music producers since then. We chose the '303' based on our interest of working with waveforms containing strong resonances widely accepted as being musically relevant. The '303' filters are characterised by a sharp cutoff of 24 dB/octave [14] being able to produce very apparent and clean resonances. The '303' also features an "Env mod" knob, that adjusts the filter's envelope signal influence on the filter's cutoff frequency. By turning this control clockwise, the envelope will sweep the cutoff frequency over a greater range. When turned counter-clockwise, the filter's envelope will have very little affect on the filter's cutoff frequency [15].

Another control in the '303' is a "Decay" knob, which controls the filter's envelope decay time. Longer envelope decay times will allow high frequencies to pass through the filter for a longer amount of time. Turning this control counterclockwise will shorten the amount of time high frequencies can pass through the filter [15]. A "Cutoff" parameter sets the cutoff frequency of the low-pass filter [16] and the "Resonance" sets the Q factor, accentuating frequencies close to the cutoff [15].

We recorded lots of samples using the sawtooth waveform, exploring variations within the mentioned controls. Figure 1 illustrates as an example some periods of a very resonant waveform we used in the experiment. Notice the characteristic appearance of resonant waveforms, with a series of rapid large oscillations at the resonant frequency imposed on a periodic variation at the note pitch frequency.

Four waveforms were selected for the experiments, based on complementary settings of "Resonance", "Env mod" and "Decay". The "Cutoff" was always left fully open to avoid losing the contribution of the higher partials. The settings are summarized and labeled in Table 1.

Resonance	Env Mod	Decay	Label
60%	max	min	A
60%	max	60%	B
max	75%	25%	C
max	75%	75%	D

Table 1. Knob settings for the experiments

[1] www.ime.usp.br/~ag/dl/smc15files.zip

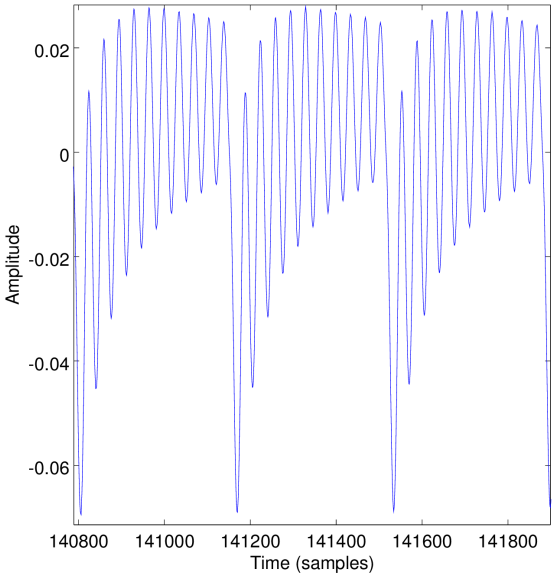


Figure 1. Some periods of a waveform generated with high resonance and decay time settings for the '303'

Waveforms A and B present the same values for a mild Resonance and a deep Env Mod, but the Decay value is minimal for A and medium for B. Waveform A is characterized by a fairly smooth sound, because although the values for Env Mod and Resonance are not small, the minimal Decay imposes a quick drop of the note, preventing the development of the modulation and resonance. The generous value for the Decay in Waveform B establishes a tailored resonance throughout the sound. Waveforms C and D are based on the maximum value for resonance and a strong Env Mod, differing by the small and large values for the Decay. The resonance generated in these cases is more aggressive, ringing for all the sound duration in Waveform C and predominately as a sweep in D.

3. AM/FM ANALYSIS

The AM-FM decomposition is a powerful method for the analysis of non-stationary signals [9]. Consider a signal

$$x(t) = a\cos(\omega t + \phi), \tag{1}$$

where a is the amplitude, ω is the frequency and ϕ an initial phase. The argument $(\omega t + \phi)$ is the instantaneous phase, and its derivative ω is the instantaneous frequency (IF).

We could also modulate both the amplitude and the frequency of the signal in (1) to give [8]

$$x(t) = a(t)\cos(\theta(t)), \tag{2}$$

where the phase derivative $\dot{\theta}(t) = f(t)$ is the IF of a signal, and $a(t)$ its instantaneous amplitude (IA). The IF can be described as the frequency of the sinusoid that locally fits the signal at instant t [9].

The AM/FM signal analysis is intended to decompose a signal into functions for the AM (related to the IA signal)

and the FM (related to the IF signal). A number of techniques exists for that [17] [8]. In this work we assume a mono component signal and apply the analytic signal based approach based on the Hilbert Transform (HT) decomposition. The HT of a signal $x(t)$ is given by [9]

$$\hat{x}(t) = x(t) * \frac{1}{\pi t}. \quad (3)$$

This creates a 90° phase shifted version of the original, from which we build the analytic signal related to $x(t)$ as

$$z(t) = x(t) + j\hat{x}(t) = |z(t)|e^{j\theta(t)}. \quad (4)$$

For a signal of the form of (2), $|z(t)|$ and $\dot{\theta}(t)$ can be used as estimates for the AM and FM. Once we have these estimates for $a(t)$ and $f(t)$, we use them to resynthesize the original signal with the expression

$$y(t) = a(t)\cos\left(\int_{-\infty}^{t} f(\tau)d\tau\right) \quad (5)$$

4. SMOOTHING AND RESYNTHESIS

In this work we want to investigate the extent of the impact caused by modifications of the estimated signals for AM and FM, so before the resynthesis we apply smoothing on the AM and FM signals. The expression for the output signal is given by

$$y(t) = (a * w)(t)\cos\left(\int_{-\infty}^{t} (f * w)(\tau)d\tau\right), \quad (6)$$

being $w(t)$ the window and '$*$' the convolution operator. We also experimented the smoothing of only one signal at a time, either the AM, replacing $(f * w)(\tau)$ by $f(\tau)$ in (6) or the FM, replacing $(a * w)(t)$ by $a(t)$.

We tested two types of windows for the smoothing, namely the rectangular (Boxcar) and the Hanning windows. For each window, two lengths were experimented, 20 and 100 samples. We will refer to these configurations as B20, B100, H20 and H100, respectively. The types and lengths of the windows were chosen based on experiments comparing the sidelobes behaviour [18] and influence on sound for different lengths of widely used windows.

5. PSYCHOACOUSTIC EVALUATION

Besides the important subjective assessment of the results, some psychoacoustic metrics [19] were chosen to quantify objectively the results of smoothing the AM and FM signals. The dynamic brightness, tristimulus and spectrum irregularity of the sounds were observed. Now we will introduce the tool used to derive the harmonic estimation, and after that discuss the metrics.

5.1 Complex Signal Phase Evolution (CSPE)

For the derivation of the psychoacoustic metrics of a sound we need to know the contribution of its frequency components. The CSPE is a tool to decompose a signal into its sinusoidal components, working around the limitations of the Discrete Fourier Transform (or the Fast Fourier Transform). If sr and N respectively are the sample rate of the signal and N the size of the window for the analysis, the Fourier method presents a limited frequency resolution. Partials located exactly at multiples of $\frac{sr}{N}$ are clearly identified, but frequencies located far from these multiples are distorted in the analysis. Another problem related to the DFT/FFT is the time/frequency accuracy tradeoff, or uncertainty principle, where good resolution for one information comes with the detriment of the other (resulting in bias either in frequency or time).

In order to enhance the results of the FFT, the CSPE algorithm analyse the phase evolution of the components between N points frames of the signal and its time-delayed version [20]. An FFT analysis is performed on the signal, and another FFT is performed at a one-sample delayed version. The delayed version spectrum is multiplied with a complex-conjugate version of the signal, resulting in a frequency dependent function [21], from which we derive the spectral envelope as the set of values $a_k, k = 1..M$, where M is the highest relevant partial and a_k is the weight of partial k. More details about the CSPE and its mathematical development can be found in [20] and [21].

5.2 Some psychoacoustic metrics

Deriving metrics from audio excerpts, and correlating them to parameters that lead to these sounds, can enlighten comprehension, perception, and composition, as timbre attributes emerge [19]. Thus, here psychoacoustics evaluations help us quantify objectively the results of the modifications we introduced. Next we discuss the metrics used in this work.

5.2.1 Brightness

The brightness, or spectral center of gravity of a spectrum, is related to the spectral centroid, and "may be thought of as the harmonic number at which the area under the spectral envelope described by a_k is balanced" [22]. In such a way the brightness can be intuitively related to the most prominent portion of the spectrum of a sound. Specially when dealing with resonances, which impose a high selectivity on the spectrum, the brightness is an indication of the spectral localization of the resonance.

There are different definitions for the brightness of a sound. In this work we calculate it with the expression [22]

$$Br = \sum_{k=1}^{N} ka_k / \sum_{k=1}^{N} a_k \quad (7)$$

As noted by Beauchamp [22], brightness appears as a common feature in works that investigate timbral modification. It was shown [23] that brightness is usually the metric that presents larger variations within psychoacoustic experiments, suggesting that it is a strong measure to characterise sounds.

5.2.2 Tristimulus

According to Jensen [19], the tristimulus concept was introduced in [24] as a timbre equivalent to the color attributes in vision. It was used to analyse the transient be-

haviour of musical sounds. The tristimulus (T_{r1}, T_{r2} and T_{r3}) metrics are defined as [19]

$$T_{r1} = a_1 / \sum_{k=1}^{N} a_k \tag{8}$$

$$T_{r2} = (a_2 + a_3 + a_4) / \sum_{k=1}^{N} a_k \tag{9}$$

$$T_{r3} = \sum_{k=5}^{N} a_k / \sum_{k=1}^{N} a_k \tag{10}$$

Notice that $T_{r1}+T_{r2}+T_{r3} = 1$, so we can instantly see a lot of information about a specific instrument by plotting a T_{r3} x T_{r2} graph, like the example (from [19]) presented in Figure 2, which shows the case for some known acoustic instruments. If the point falls close to origin the instrument should present a strong fundamental, because T_{r2} and T_{r3} would be small, so T_{r1} would be close to 1. An analogous reasoning holds for the other corners, so close to the right corner of the triangle is the case of strong higher frequency partials, while the last corner would be one related to strong mid-range partials.

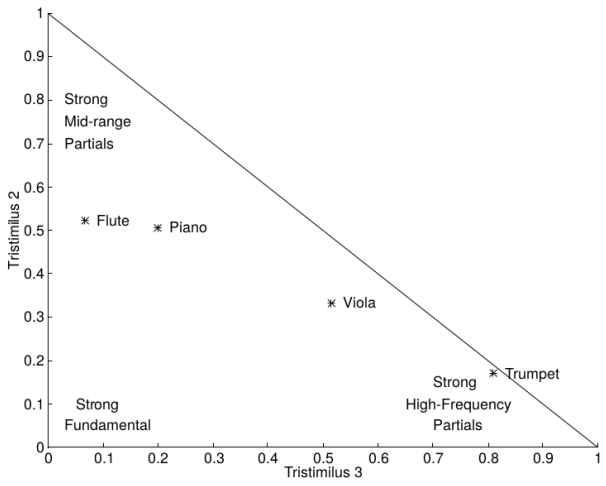


Figure 2. Visualising the tristimulus triangle. Source: [19]

A relation between the tristimulus metrics is also similar to another interesting psychoacoustic metric, the warmth, defined [25] as the energy contained in the partials up to 3.5 times the fundamental frequency of a sound, over the energy of the partials from 3.5 times the fundamental frequency to the uppermost harmonic. As highlighted in [25], it is likely that a processing which increases the warmth, or the tristimulus 2, will decrease the brightness, as the spectral centroid will become lower.

5.2.3 Irregularity

The irregularity of a spectrum is a measure of its "jaggedness", as termed by McAdams *et al.* in [10], where they conclude that this metric is one of the most perceptually important parameter regarding timbre discrimination.

There are several formulas for the calculation of spectrum irregularity. We use the one given by [19]

$$Ir = \sum_{k=1}^{N} (a_k - a_{k+1})^2 / \sum_{k=1}^{N} a_k^2, \quad a_{N+1} = 0 \tag{11}$$

Since the expression for irregularity involves neighbouring partials, it reflects ripples in the spectrum, or the more extreme case of missing partials. An example of a sound with high spectral irregularity would be the square wave, or also the clarinet, which have components at odd harmonics only. These are known as having a hollow timbre, in contrast with the buzz timbre of an impulse train, which has zero irregularity.

5.3 Results

In this section we present and discuss some of the graphs we obtained in the study. All the graphs considering all the cases are available for download.

5.3.1 Brightness

Figures 3 and 4 show that the unprocessed waveforms start with a high brightness value and soon stabilise. Notice that the brightness values for A are smaller than those for C, and that was expected, due to the more modest settings used on the '303' for its generation. A's milder brightness can also be checked by listening to the samples.

When we consider the smoothed reconstructions for Waveform A (Figure 3) we see that the B100 is brighter than H100 during the early portion of the sound, but then they meet at a value smaller than the brightness for the pure waveform. B20 and H20 showed similar values, always close to the original. With Waveform C (Figure 4), the brightness values for all the 4 smoothers and the unprocessed case are practically the same.

Figure 5 shows the brightness values when only the AM is smoothed and Figure 6 when only the FM is smoothed. Notice that in this case the FM-only smoothing does not significantly impact the results, but the AM-only smoothing with the higher order windows augments the brightness. Similar behavior was observed for the other 3 waveforms.

Considering the processing of the highly resonant Waveforms (C and D), the brightness was not affected. Considering the milder resonance of A and B, the higher order smoothers imposed a smoothed brightness, but the small order smoothers augmented it.

It is interesting to notice that when we apply AM-only or FM-only smoothing in Waveform A with B100 and H100, the result is an increased brightness. However, when we apply both simultaneously the brightness is reduced, so a sort of cancellation happens, probably due to the creation of harmonics on the lower register.

5.3.2 Tristimulus

The tristimulus plots for the highly resonant C and D show a smaller variation comparing to the Waveforms A and B. Figure 8 show that tristimulus for C remains confined in the area where T_{r3} is high. Figure 7 show for A a larger excursion for T_{r2} and T_{r3} (and consequently, for T_{r1}).

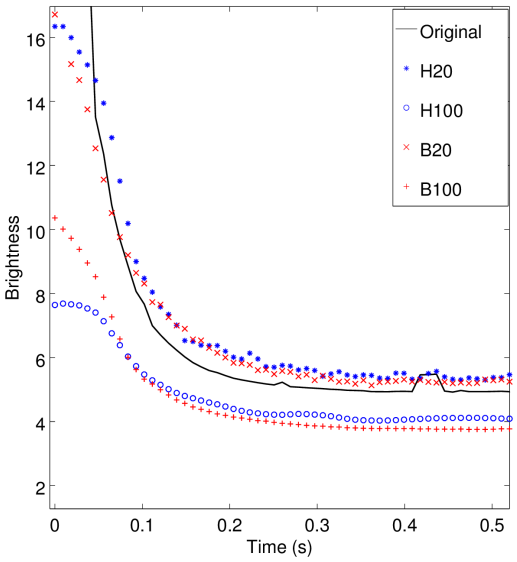


Figure 3. Comparison of brightness values over time for waveform A AM/FM smoothing using all the configurations considered. H20 smoother plotted with blue '∗', H100 with blue circles, B20 with red 'x', and B100 with red '+'. This legend code holds for all plots in this section.

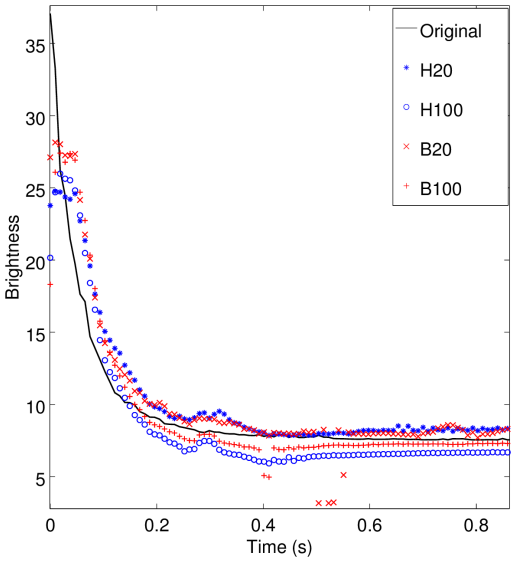


Figure 4. Comparison of brightness values over time for waveform C AM/FM smoothing using all the configurations considered

Figure 9 shows close values for tristimulus when considering only AM smoothing and the original case, but the FM-only smoothing (Figure 10) produces a larger variation for tristimulus. We can also check that the higher order smoothers were more effective for this variation than the low order ones.

Like what happened with brightness, we see that the tech-

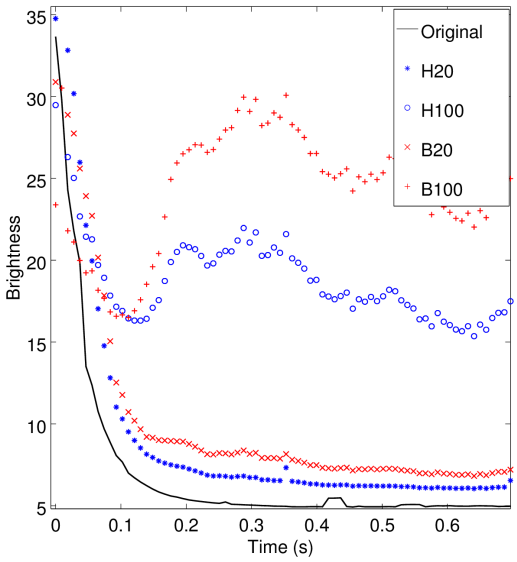


Figure 5. Comparison of brightness values over time for AM-only smoothing, using waveform A

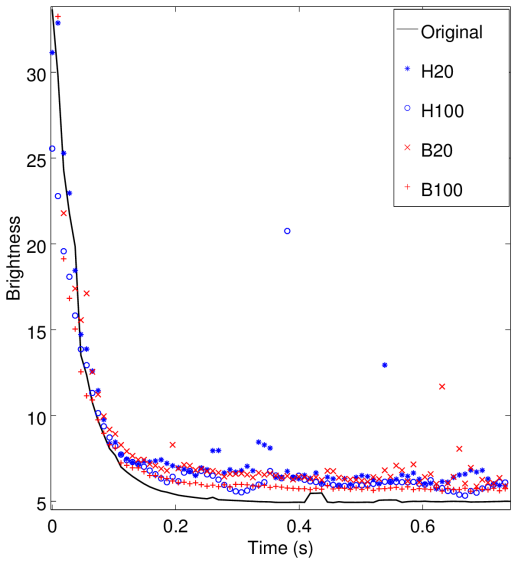


Figure 6. Comparison of brightness values over time for FM-only smoothing, using waveform A. The spikes observed are artefacts from the harmonics finding process

nique does not significantly affect the tristimulus for the waveforms with high resonance, but for mild resonance sounds the smoothers impose a variation proportional to their length. In contrast with the brightness values, here it is the FM smoothing that is the main source of variation in the processing.

5.3.3 Irregularity

The plots for the irregularity of Waveforms A (Figure 11) and B show a constant value throughout the sound. That is

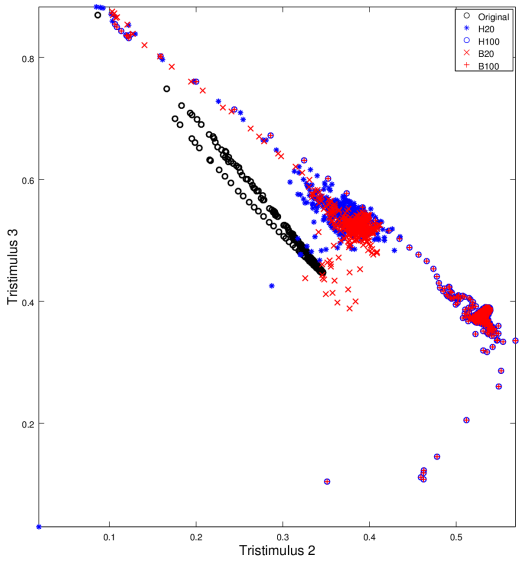


Figure 7. Tristimulus triangle for waveform A AM/FM smoothing with all the configurations considered

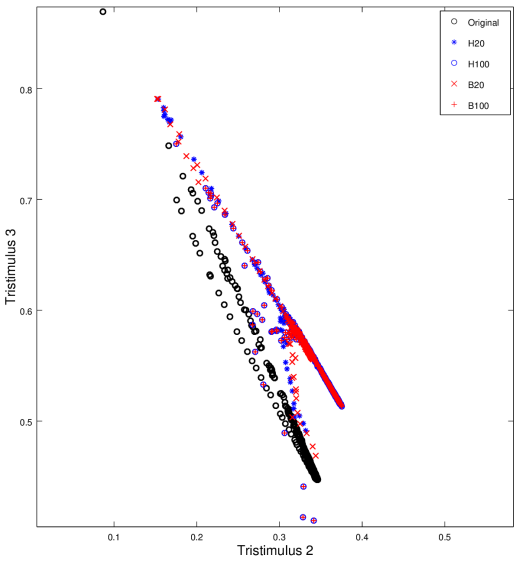


Figure 9. Tristimulus triangle for AM-only smoothing, using waveform A

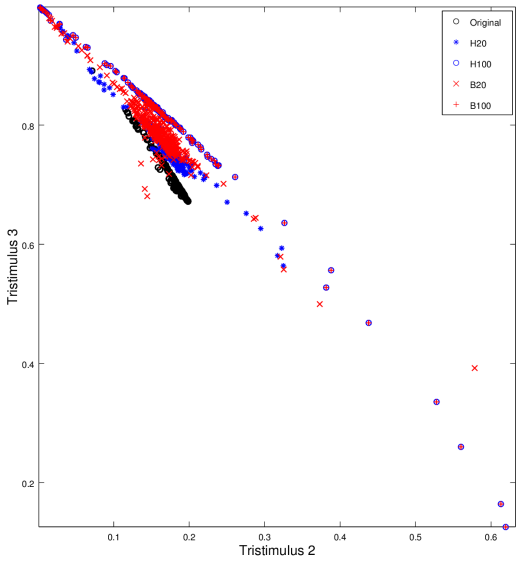


Figure 8. Tristimulus triangle for waveform C AM/FM smoothing with all the configurations considered

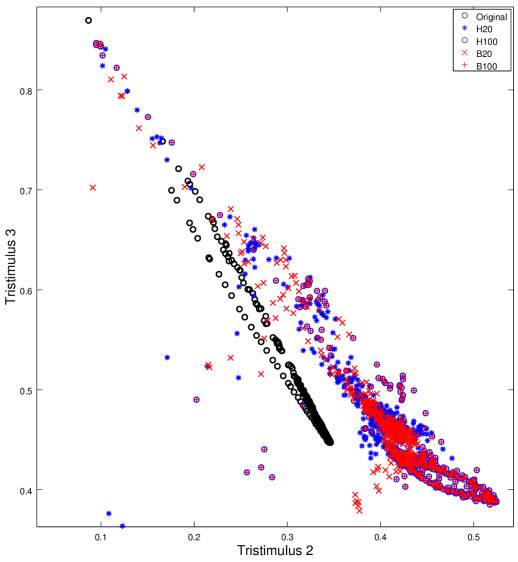


Figure 10. Tristimulus triangle for FM-only smoothing, using waveform A

not the case for C (Figure 12) and D, which show a periodically varying irregularity, with a large excursion.

Figure 11 shows that the H100 smoother matches the unprocessed case when processing A, while the small order smoothers decrease the irregularity and the B100 doubles it. The plots considering Waveforms C and D show similar values for all the smoothing configurations, with non-fluctuating values smaller than the originals.

Figures 13 and 14 show the cases for the AM-only and FM-only smoothing for the Waveform C. Notice that for the AM-only smoothing case the values are all similar, and smaller than the original, while for the FM-only smoothing only the higher order smoothers decreased the irregularity.

6. CONCLUSIONS

As a general trend, from the brightness and tristimulus points of view, it is not that effective to smooth signals with strong resonance, although it certainly changes the sound. Also, the tristimulus is more affected by smoothing of the instantaneous frequency component, while the irregularity is more affected by the AM smoothing.

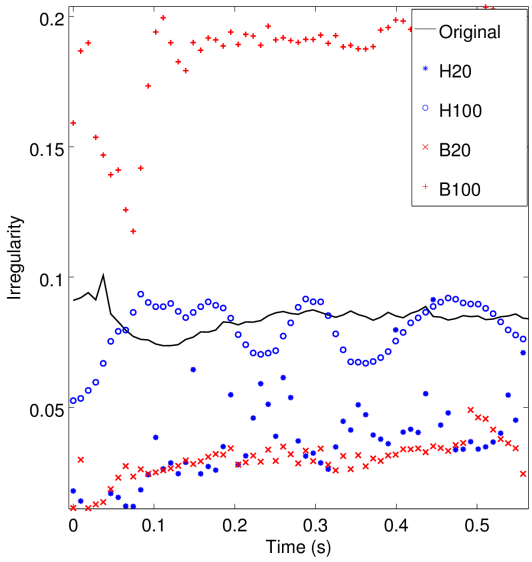


Figure 11. Comparison of irregularity value over time for waveform A AM/FM smoothing using all the configurations considered

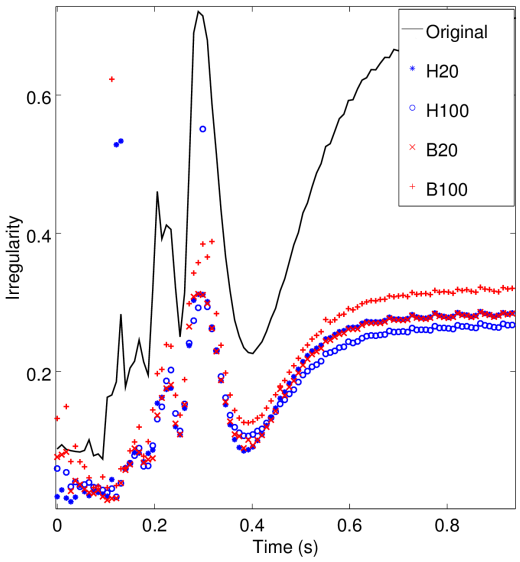


Figure 13. Comparison of irregularity value over time for AM-only smoothing, using waveform C. The spikes observed are artefacts from the harmonics finding process

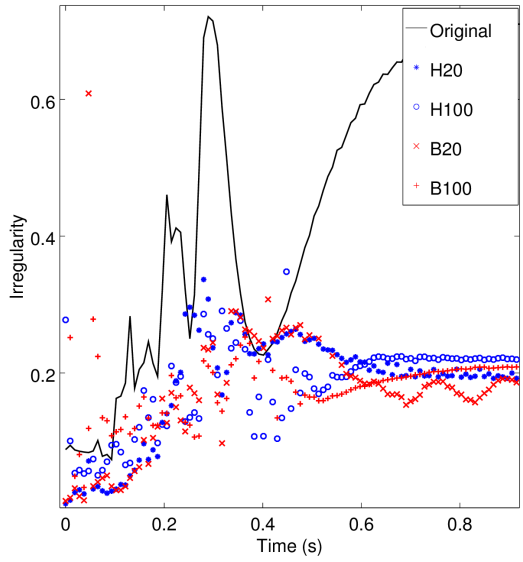


Figure 12. Comparison of irregularity value over time for waveform C AM/FM smoothing using all the configurations considered

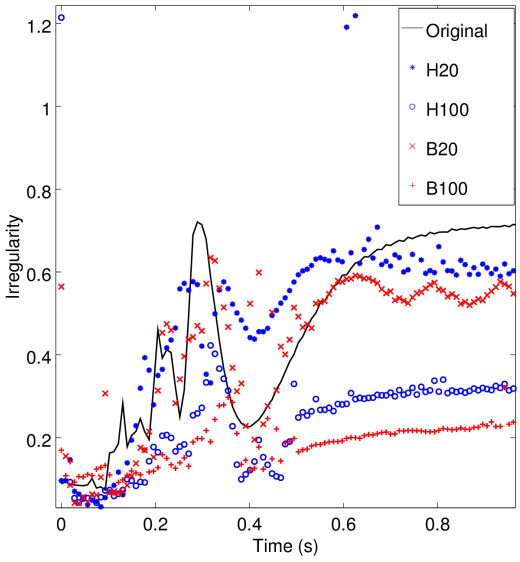


Figure 14. Comparison of irregularity value over time for FM-only smoothing, using waveform C. The spikes observed are artefacts from the harmonics finding process

According to the psychoacoustic metrics obtained in the study, it seems that the more irregular the input signal spectrum is, the more similar the perceptual outcome of smoothing will be in comparison to the original, regardless of the window used. Values will be typically lower, indicating a smearing or flattening of the spectrum, typical of modulations with high depth or index. There could be a relationship between this flattening and the spectrum whitening described in [26], but this needs further investigation.

The processing of modest and mild resonances, however, presents variations compared to the original case, and the new sounds obtained are musically interesting, indicating that AM/FM based techniques can be useful for the introduction of liveliness into flat resonance sounds. According to what was expected, the longer smoothers led to sounds with less artefacts in comparison to the sounds produced with the short smoothers. Also, the Hanning smoothers produced less artefacts than the Boxcar's.

Currently we are investigating possibilities to generalize the framework as a suite of AM/FM audio effects. It seems that all smoothers are an interesting possibility for the processing of sounds with a modest to medium resonance, as the overall original brightness shape is preserved, so the effect keeps a lot of the sound's original feel.

Acknowledgments

The research leading to this paper was performed during Antonio's internship period at Maynooth University, and was partially supported by CAPES (proc num 8868-14-0).

7. REFERENCES

[1] G. Loy, *Musimathics - Volume 1*. Cambridge, MA, USA: MIT Press, 2006.

[2] X. Rodet, "Time-domain formant-wave function synthesis," *Computer Music Journal*, vol. 8, no. 3, 1984.

[3] W. Kaegi and S. Tempelaars, "VOSIM - A new sound synthesis system," *Journal of the Audio Engineering Society*, vol. 26, no. 6, pp. 418–425, 1978.

[4] V. Lazzarini and J. Timoney, "Theory and practice of modified frequency modulation synthesis," *Journal of the Audio Engineering Society*, vol. 58, no. 6, 2010.

[5] M. Ishibashi, "Electronic musical instrument (patent US 4658691)," Patent 4 658 691, April, 1987.

[6] J. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *Journal of the Audio Engineering Society*, vol. 21, no. 7, pp. 526–534, 1973.

[7] M. Russ, *Sound synthesis and sampling*, 3rd ed. Burlington, MA, USA: Taylor & Francis, 2008.

[8] B. Picinbono, "On instantaneous amplitude and phase of signals," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, 1997.

[9] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal-part 1: Fundamentals," *Proceedings of the IEEE*, vol. 80, no. 4, Apr 1992.

[10] S. McAdams, J. W. Beauchamp, and S. Meneguzzi, "Discrimination of musical instrument sounds resynthesized with simplified spectrotemporal parameters," *J. Acoust. Soc. Am.*, vol. 105, no. 2, pp. 882–897, 1999.

[11] R. Sussman and M. Kahrs, "Analysis and resynthesis of musical instrument sounds using energy separation," in *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-96)*, vol. 2, May 1996, pp. 997–1000 vol. 2.

[12] I. Arroabarren, X. Rodet, and A. Carlosena, "On the measurement of the instantaneous frequency and amplitude of partials in vocal vibrato," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 1413–1421, 2006.

[13] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell System Technical Journal*, vol. 45, no. 9, 1966.

[14] T. Stinchcombe, "Diode ladder filters," http://www.timstinchcombe.co.uk/index.php?pge=diode, online; accessed 27 April 2015.

[15] J. Flickinger, "Future retro - your guide for the revolution," http://www.future-retro.com/pdf/revolutionmanual.pdf, online; accessed 19 June 2015.

[16] R. Corporation, "TB-303 owner's manual."

[17] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 4, pp. 744–754, Aug 1986.

[18] A. H. Nutall, "Some windows with very good sidelobe behaviour," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 1, pp. 84–91, Feb 1981.

[19] K. Jensen, "Timbre models of musical sound: From the model of one sound to the model of one instrument," Ph.D. dissertation, 1999.

[20] R. A. Garcia and K. M. Short, "Signal analysis using the complex spectral phase evolution method," in *Audio Engineering Society Convention 120*, May 2006.

[21] J. Wang, J. Timoney, and M. Hodgkinson, "Using apodization to improve the performance of the complex spectral phase estimation (CSPE) algorithm," in *Proceedings of the China-Ireland International Conference on Information and Communications Technologies*, 2009.

[22] J. Beauchamp, "Synthesis by spectral amplitude and 'brightness' matching of analyzed musical instrument tones," *J. Audio Eng. Soc*, vol. 30, no. 6, 1982.

[23] G. von Bismarck, "Sharpness as an attribute of the timbre of steady sounds," *Acta Acustica united with Acustica*, vol. 30, no. 3, pp. 159–172, 1974.

[24] H. F. Pollard and E. V. Jansson, "A tristimulus method for the specification of musical timbre," *Acta Acustica united with Acustica*, vol. 51, no. 3, pp. 162–171, 1982.

[25] T. Brookes and D. Williams, "Perceptually-motivated audio morphing: Warmth," in *Audio Engineering Society Convention 128*, May 2010.

[26] J. J. Wells, "Methods for separation of amplitude and frequency modulation in Fourier transformed signals," in *Proceedings of the International Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, 2010.

MULTICHANNEL COMPOSITION USING STATE-SPACE MODELS AND SONIFICATION

Rosalia Soria-Luz
The University of Manchester
rosalia.sorialuz@postgrad.manchester.ac.uk

ABSTRACT

This paper explores the use state-space models and real time sonification as a tool for electroacoustic composition. State-space models provide mathematical representations of physical systems, making possible to virtually capture a real life system behaviour in a matrix-vector equation.

The paper presents an inverted pendulum, a mass-spring-damper system, and a harmonic oscillator, implemented in Supercollider, and different real time multichannel sonification approaches, as well as ways of using them in electroacoustic composition.

1. INTRODUCTION

Sonification has become a very important tool to convey information and represent/analyse data using sound. In recent years a musical interest has developed as a result of the aesthetic awareness on the different ways of representing data in meaningful ways [1] [2] [3].

Mathematical models play a very important role in the sound synthesis field. For instance physical modelling, modal synthesis and and digital waveguides aim to model the acoustic behaviour of sound producing objects [4] [5].

This paper shows the use of mathematical models of dynamic systems for the creation of synthetic sound, however this models don't represent the acoustic nature of such systems but rather motion behaviours to be sonified using abstract sound synthesis techniques.

This paper explores the uses of real time sonifications of state-space models as a source for electroacoustic composition. A motivation to use state-space models is finding ways to produce synthetic sound which evolves in an organic way.

Sonifiying state-space models can be described as a parametric model based interactive sonification [6]. It is based on the implementation of physical systems, as mathematical models in their state-space form, and the possibility of exciting and sonifying them in real time. The advantage of the state-space form is that a vector containing the states of system can be available in real time, making simultaneously available several variables from the same system. The aim in these sonifications is not necessary accurately representing the system's behaviour, but rather looking for diverse sound timbres, or behaviours when simultaneously mapping all system variables into sound generation, or sound transformation parameters. A multichannel approach is used to add an spatial element in the composition process.

Copyright: ©2015 Rosalia Soria-Luz 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.

2. STATE-SPACE REPRESENTATIONS

State-space mathematical models are a useful tool to represent dynamical physical system's behaviour in a compact way. Provided the system is linear and time invariant, it is possible to formulate a matrix equation that describes the systems behaviour . The matrix representation is called the state-space representation of a continuos system dynamics and can be written as follows:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1}$$

Where $x(t) \in R^n$ is the n-dimensional state vector, $u(t) \in R^m$ is the m-dimensional input vector, and $y(t) \in R^p$ is the p-dimensional output vector. A and B are constant matrixes, containing information about the characteristics defining the system. C determines which states are the model outputs, and D is a feedback matrix connecting directly the output to the input [7].

The state-space models used for this paper consist of one input and n outputs, depending of the nature of the system. The implemented models are sampled digital representations of the continuos systems.

The sampled version of (1) with sampling period T_s can be represented as follows:

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{2}$$

Where $\Phi = e^{aT_s}$ and $\Gamma = \int_0^{T_s} e^{As} ds$ are obtained by considering a zero order sample and hold circuit, and the simplified notation k refers to a general time $t_k \in \{t_0, t_1 ... t_N\}$ [8].

3. SYSTEM DESCRIPTION AND IMPLEMENTATION

The three specific systems are described briefly to understand the physical nature of their behaviour. Although the descriptions show the continuos systems, the implemented ones are digital representations in the state-space form (2).

Matrixes A and B were calculated using MatLab. The specific variable values will be shown for each system [1]. The corresponding matrixes Φ and Γ for a specific sampling period were calculated using MatLab.

3.1 Mass-Spring-Damper

The mass-spring-damper system is depicted in Fig. 1. The input to the system is the force $f(t)$ applied to the mass. The system outputs $y(t)$ are the mass position (displacement) and velocity [9].

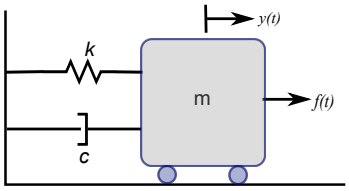


Figure 1. Implemented Mass Spring Damper system illustration.

When a force is applied to the mass, it will start a unidimensional movement in the direction of the applied force. This movement has an oscillatory elastic nature due to the action of the spring (k) , and will be dampen due to the action of the damper (c). The movement will eventually stop if no further force is applied.

The implemented sampled version of this system with $m = 1kg$, $k = 1N/m$, $c = 0.2Ns/m$, sampling period $T_s = 0.1$ secs, and in the form (2) has the following system matrixes:

$$\Phi = \begin{bmatrix} 0.99503 & 0.09884 \\ -0.09884 & 0.97526 \end{bmatrix}, \Gamma = \begin{bmatrix} 0.00496 \\ 0.09884 \end{bmatrix}, \quad C = [1, 1], \quad D = 0 \tag{3}$$

3.2 Inverted Pendulum

The inverted pendulum system is depicted in Fig 2. The system consists of an inverted pendulum mounted on a mobile cart. If the system is not controlled, the pendulum will fall over when an input force is applied to the cart. For this reason a controlled system was implemented [10]. The aim of the controller is that when the cart is displaced to a desired position, the pendulum is able to come back to equilibrium vertical position. The input of the system is a number representing a desired position for the cart (meters). The outputs are the cart's displacement (meters), cart's velocity, the pendulum angle (radians) and pendulum angular velocity.

The inverted pendulum with $M = 0.5, m = 0.2, I = 0.006, l = 0.3$ and a sampling period $T_s = 0.02$ secs is

[1] For more details on the calculation of matrixes A, B, C, D consult the specific references for each system.

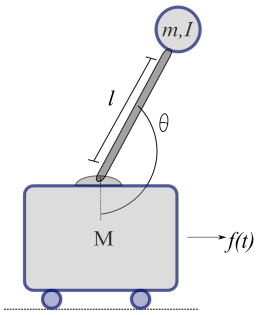


Figure 2. Implemented Inverted Pendulum physical representation.

represented by the following matrixes:

$$\Phi = \begin{bmatrix} 1.0056 & 0.0130 & -0.0085 & -0.0017 \\ 1.1263 & 1.6069 & -1.7003 & -0.3420 \\ 0.0141 & 0.0076 & 0.9800 & 0.0057 \\ 2.8168 & 1.5178 & -4.0073 & 0.1460 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} -0.0056 \\ -1.1263 \\ -0.0141 \\ -2.8168 \end{bmatrix}, C = [1, 1, 1, 1], \quad D = 0 \tag{4}$$

3.3 Harmonic Oscillator

A harmonic oscillator system representation is depicted in Figure 3. It consists of a mass attached to a spring and connected to a rigid wall in a non friction environment. If no force is applied, the mass remains in its equilibrium state. If a force is applied to the mass, an elastic force due to the spring will try to restore the equilibrium state, producing a periodic movement. The system input is therefore a force applied to the mass, and the outputs are the mass unidimensional position, and velocity [11].

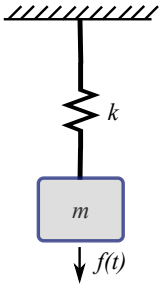


Figure 3. Harmonic Oscillator physical representation.

The digital implemented system with $\sqrt{k/m} = 133$, [2] and $Ts = \pi/32$, is represented by the following matrixes:

$$\Phi = \begin{bmatrix} 0.8819 & 0.4714 \\ -0.4714 & 0.8819 \end{bmatrix}, \Gamma = \begin{bmatrix} 0.1181 \\ 0.4714 \end{bmatrix}, \quad C = [1, 1], \quad D = 0 \tag{5}$$

[2] This ratio guarantees a solution for the motion equations [12].

3.4 SuperCollider Implementation

The implementation in Supercollider was achieved by creating a class to handle linear algebra matrix operations. The class arguments are the matrixes Φ, Γ, C, D; and the numeric input u . The class outputs are systems state vector x. State variables update in real time according to the sampling time parameter Ts, which can also be modified in real time.

The implemented systems work according to the following criteria: (1) All systems require only one numeric input; (2) the systems state-space is the set of variables in the vector $x(k)$ describing the systems behaviour (i.e. position and velocity in the mass-spring damper system); (3) the output vector depends on the matrix C, the elements in this vector represent physical variables of the system; (4) the systems current output depends on the current input value, a previous input value, the current state and a previous state; (5) the system's output vector updates according to a sample period time parameter; (6) the numeric input u, and sample period T_s can be changed in real time: the output state vector updates accordingly to this values.

The implementations were tested using SuperCollider version 3.6.5, on a 2.3 GHz Intel Core I7 processor, OS X version 10.8.5. The average time to compute a state vector for the inverted pendulum is 99.681998108281e-06 secs, and for the mass-spring-damper system and harmonic oscillator 71.91300028353e-06 secs.

4. SONIFICATION AND STATE-SPACE MODELS

4.1 Sound Synthesis

A first sonification approach is generating sound synthesis. This can done by simultaneously mapping all outputs from a chosen system into different parameters of the same synthesiser. Even though AM and FM techniques are used to create this sonifications, sound becomes part of the model dynamics and evolves in an organic way as the modulating signals come from the same system.

As an example, a combination of AM/FM synthesiser was used to sonify the mass-spring-damper system. As the model behaviour is available in real time, it is possible, to "virtually play it" by applying numerical inputs in real time.

Having in mind that the output variables represent position and velocity, it is possible to design a synthesiser to sonify them. Fig. 4 shows the stereo sonification schematic diagram. The position is mapped into the $f1$ parameter, it is scaled and used to modulate in frequency a sine wave. The velocity is also mapped into a $f2$ parameter, after scaling, it modulates in frequency a saw tooth wave. The scaling values for both variables are shown in Table 1.

Additional scaling factors are added in the syntheziser, this is represented by the different coefficients in the schematic diagram. This gives diversity to the sound although both channels are related as both are sonifications of the same model. Once the synthesiser is set-up, the numerical input can be changed in real time and the sound will evolve in time according to the system's nature. What is expected is a sinusoidal behaviour exponentially decaying. This means that if the input doesn't change, the output states will reach a stationary state with almost no oscillations. A recording of this sonification process can be found at [3] .

Variable	original range (abs)	scaled range
position	0-16000	30-16000
velocity	0-6000	200-6200

Table 1. Scaling values for the the mass-spring-damper system variables.

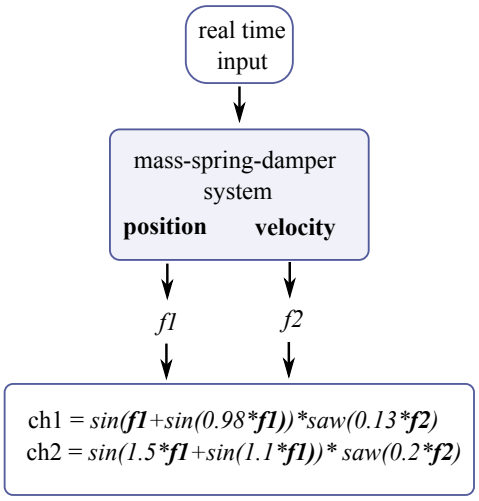


Figure 4. Mass-Spring-Damper system stereo sonification example.

Using the same synthesiser and same scaling values it is also possible to sonify the harmonic oscillator. This system is also oscillatory, but as it doesn't have any damping element, it will continue oscillating without decaying with an amplitude proportional to the input force applied. Even though the same synthesiser is used, the sound has a different behaviour, making it possible to have variations of the same timbre. This fact can be heard at [4] .

4.2 Multichannel approach

It is also possible to expand the stereo sonification idea to a multichannel approach. This can be achieved by creating multichannel synthesisers all driven simultaneously by the same model. As an example, a 4-channel sonification for the inverted pendulum is depicted in Fig. 5. The 4-ch synthesiser is designed to have three control variables. Pendulum position, cart velocity and pendulum angle are scaled and mapped to simultaneously control these variables. Table 2 shows the scaled values for the inverted pendulum.

The cart position is mapped so it modulates in frequency a sine wave, the velocity and angle modulate a low frequency triangle wave. Extra scaling factors are added per channel in the sinthesizer, as shown in Fig. 5. The extra scaling factors have the purpose of creating sound families by placing related sounds in different channels (speakers), to involve the spatial factor.

[3] https://dl.dropboxusercontent.com/u/88409515/sonif1.aif
[4] https://dl.dropboxusercontent.com/u/88409515/sonif2.aif

Variable	original range (abs)	scaled range
position	0-16000	80-6480
velocity	0-5700	8-1148
angle	0-13000	2.5-132.5

Table 2. Scaling values for the inverted pendulum.

The expected behaviour is an oscillating movement that will settle when it reaches the desired position according to the real time input. Velocity will settle at 0 when the final position is reached, and the same will happen for the angle, as the pendulum has to end up in a vertical position. This means that after a change in the input, sound will vary according to changes in position, velocity and angle; if the input doesn't change, the sound will reach a stationary state. A recording of this sonification example can be found at [5].

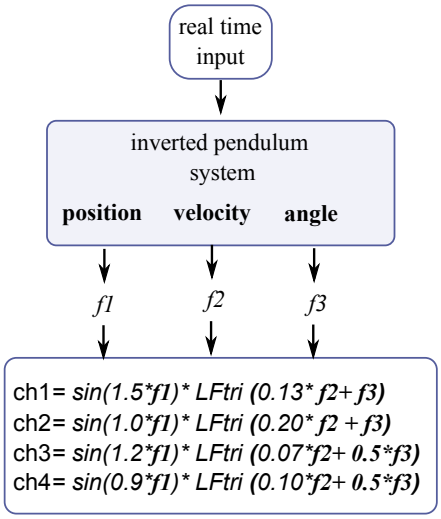


Figure 5. Inverted Pendulum system. 4-ch sonification example.

4.3 Sound transformation

Sonification can also be achieved by transforming sound. As an example, a VOSIM oscillatoris used to sonify the spring-mass-damper system [6]. The position is scaled and mapped into the pulse trigger frequency whereas the velocity is mapped into the VOSIM frequency. Velocity is also used to modify a time delay parameter that affects the VOSIM oscillator. Fig. 6 shows and schematic diagram for this example. A fragment of this sonification can be found at [7].

5. COMPOSING WITH STATE-SPACE MODELS

5.1 Sonifications and Sound Synthesis

As the models behaviour is available in real time, a combination synthesiser-sonification can be seen as an instrument

[5] https://dl.dropboxusercontent.com/u/88409515/sonif3.aif

[6] VOice SIMulation, "... is a signal consisting of sequences of sin^2 pulses each of the same duration and decreasing amplitude" [13].

[7] https://dl.dropboxusercontent.com/u/88409515/sonif4.aif

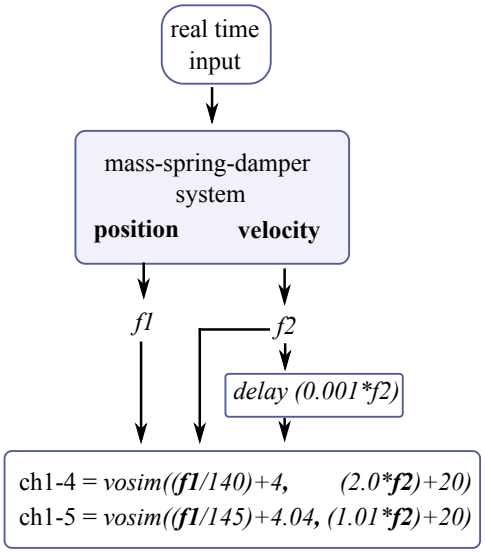


Figure 6. Mass-spring-damper system. Sound synthesis and sound transformation 8-ch sonification example.

that can be played in real time. Depending on the mapping and synthesiser complexity, it is possible to generate different gestures, textures [14] and timbres.

This implies that the composition process includes a sound synthesis design step which involves choosing a model to sonify; awareness of the model's behaviour to achieve meaningful mappings of the different parameters (knowing numeric range of the outputs for different inputs and consider this when scaling variables); and experimenting with pairs model - synthesiser to explore behaviour and timbre possibilities for different inputs.

So far only the input-output aspect of the models has been discussed. However, there is a third factor included in the implementation and that is the sample period T_s. This parameter controls how often the output states update according to the input. Even though this parameter could remain fixed, it is possible to change it in real time to virtually "time stretch" or expand the systems behaviour, or even more, to freeze the system time evolution, making it possible to time stretch, expand, of freeze timbres.

Fig. 7 shows an interface created in Supercollider to interact with sonification synthesisers and model parameters in real time. The state-space input window allows the user to interact with the models by changing the input parameters according to Table 3.

Symbol	Parameter
t1	Sampling period value in secs (0.01-1)
uc	Model input value 0 -10
uc1	Model input value 0 -1000
uc2	Model input value 0 -15000

Table 3. Interface parameter to interact in real time with state-space models.

The total input to the model is $U = uc + uc1 + uc2$; this combination is useful to have different input resolutions.

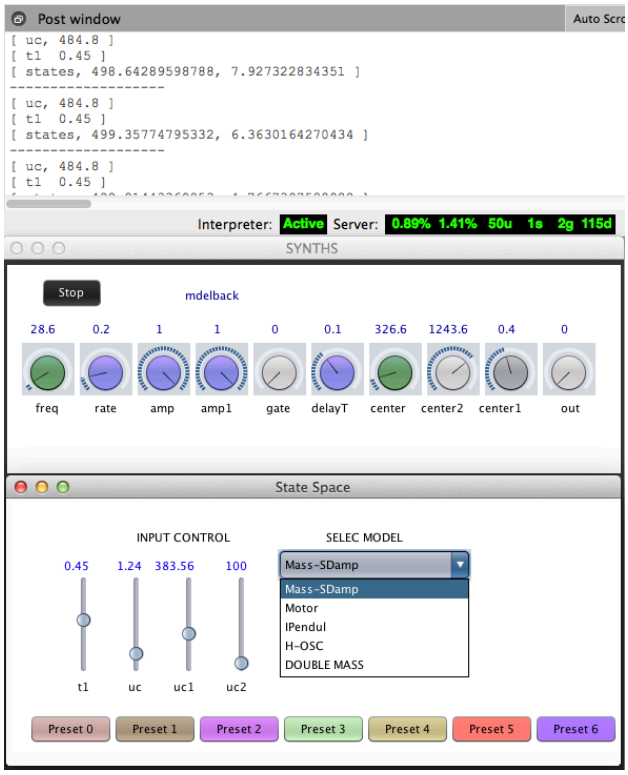


Figure 7. Supercollider interface for real time interaction with state-space models and sonification sinthesizers.

The buttons labeled as ”preset 0” to ”preset 6” trigger the models real time evolution, for different sonification presets [8]. If a button is active, the chosen model updates in real time; if not active, the model remains in the latest updated state. The interface also allows switching between the models in the menu. This is be useful when sonifying different models using the same synthesiser.

If the preset buttons are inactive, the selected model is frozen, meaning the current state is frozen, therefore timbre is frozen. In this condition it is also possible to manually change the synthesiser parameters. For example in Fig. 7, ”freq”,”rate”, etc. That means manually manipulating the timbre once the model has brought the synthesiser to a particular timbre.

5.2 Sonic Outcomes

Even though stereo sonifications can create interesting timbres, the multichannel sonification approach allows experimenting with timbre and space in a wider sense.

One possibility is the creation of sound families by placing numerically related sounds, a different one per speaker, but all controlled by the same model. An example of this is the sonification depicted in Fig. 5. In this case timbre also includes the spatial factor.

A second option is generating multichannel gesture and texture. This can be controlled by manipulating the sampling period parameter T_s in the model input: if short, the models arrive faster to a stationary state, creating gesture; if T_s is larger, sound evolves slowly creating smooth transitions and texture.

In addition, it is possible to map state-space variables into panning parameters to create gestural spatial effects.

5.3 Composition Possibilities

Once a satisfactory sonification is achieved, real time interactions can be recorded and used in fixed media or mixed media compositions. Some possibilites are including directly recorded sonification excerpts, they may be as long as desired; splitting multichannel sonification recordings; spatially rearranging them or using only selected audio channels; and applying further sound transformations to any of the above.

5.4 Example: Time Paradox

5.4.1 Sonifications and Structure

Time Paradox (2015) is an 8-channel fixed media piece composed using sonifications of the mass-spring damper system and singing bowls stereo recordings. The sonifications where designed to fit in stems of 4 or 8 channels. This offers the possibilites of locating sound families in different spatial locations when using 4-channel sonifications, and to create more atmospheric textures of related sounds when using 8-channel approaches.

The sonifications are based on a 4-channel pulse wave synthesiser (details not shown in this paper), 8-channel FM granular synthesiser, the synthesiser depicted in Fig. 6, and the 8-channel VOSIM synthesiser depicted in Fig. 8.

For the VOSIM sonification the system variables were pre-scaled with the values shown in Table 1. The mass position controls the trigger frequency for each pulse, the velocity controls the wave frequency and also controls the number of squared sine-waves to use in each pulse [9].

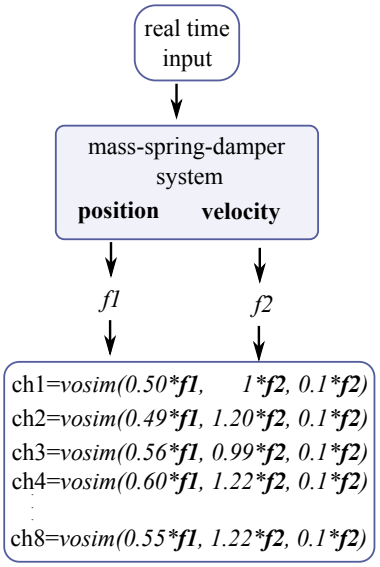


Figure 8. Mass-spring-damper VOSIM sonification for Time Paradox.

[8] It is possible to add as many presets as desired.

[9] http://doc.sccode.org/Classes/VOSIM.html

The GrainFM synthesiser consists on granular synthesis with frequency modulated sine tones [10]. The system states control the synthesiser parameters as follows: position mapped into grain trigger rate and carrier frequency; velocity mapped into modulating frequency.

The synthetic sound materials were created using the interface shown in Fig. 7.

Regarding sound materials, the piece consists of two sections. Fig. 9 shows the sound material structure per section. Section 1 includes direct recordings of sonification fragments, such as FM, Pulse, VOSIM + delay and fragments of singing bowls. Section 2 consists on split versions of the FM sinthesizer, this means using different pairs of selected channels and relocating them spatially; unmodified recording of 8ch VOSIM fragments, and stereo and 4-channel expansion of singing bowls.

However, it was necessary to create transitions between sections and musical ideas. A shaped 8ch noise synthesiser was used as transition material and it is not a sonification of the system. A stereo version of Time Paradox can be found at [11].

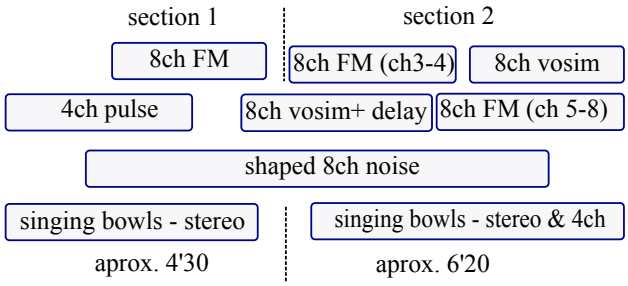


Figure 9. Time Paradox sound material structure.

6. CONCLUSION

The paper illustrates the use of state-space models in composition. Sonified state-space models offer new possibilities for creating multichannel sound materials for electroacoustic composition. As the physical systems behaviour is available in real time, a sonified state-space model can be seen as virtual instrument that can be played and recorded in real time.

They also present the possibility of creating evolving timbres as sound "obeys" the systems behaviour. This is particularly interesting when mapping simultaneously all variables of a system into sound parameters . The spatial element is also important, as multichannel sonifications can be designed fitting stems of mono, stereo or any number up to 8 channels.

One of the main advantages of using sonified models, is that sound evolves in a more organic way, as the models obey laws of physics, preventing them for abrupt changes. However, it is also possible to slow down, speed up or freeze the time evolution, or to "virtually" apply input forces that would not be possible in real life, providing different possibilities in timbre and sound evolution, such as gesture, texture, or smooth variations.

Regarding the composition process, untransformed recordings can be included as part of fixed media or mixed media pieces, but also further sound transformations can be applied as well as multichannel split or spatial rearrangements.

As a state-space framework has been developed in Supercollider, additional physical systems can be easily implemented, offering new sonification-composition possibilities for electroacoustic composition.

[10] http://doc.sccode.org/Classes/GrainFM.html

[11] https://soundcloud.com/rosalia-soria/time-paradox-stereo

Acknowledgments

The author thanks the University of Manchester PDS awards organization for funding this research.

7. REFERENCES

[1] M. SHASS. Gamma Sonification. [online] Available: http://shass.mit.edu/news/news-2014-gamma-sonification-mit-students-make-music-particle-energy.

[2] *Sonificatoin in Music*, International Conference on Auditory Display. IEM, 2009.

[3] J. G. N. Thomas Hermann, Andy Hunt, *The Sonification Handbook*, J. G. N. Thomas Hermann, Andy Hunt, Ed. Comeniushof, Gubener Str. 47, 10243 Berlin, Germany: Logos Verlag Berlin GmbH, 2011.

[4] S. Bilbao, A. Torin, P. Graham, J. Perry, and G. Delap, "Modular Physical Modeling Synthesis Environments on GPU," *Proceedings of the International Computer Music Conference. Athens, Greece, ICMC 2014 - International Computer Music Conference, Greece, 14-20 September.*, 2014.

[5] S. Bilbao, *Numerical Sound Synthesis*. John Wiley and Sons Inc, 2009, ch. 1. [Online]. Available: www.dawsonera.com

[6] J. G. N. Thomas Hermann, Andy Hunt, *The Sonification Handbook*. Comeniushof, Gubener Str. 47, 10243 Berlin, Germany: Logos Verlag Berlin GmbH, 2011, ch. 9, p. 198.

[7] R. L. Williams and D. A. Lawrence, *Linear State Space Control Systems*. Hoboken NJ: John Wiley and Sons, February 2007, ch. 1, pp. 5–6.

[8] K. J. Aström and B. Wittenmark, *Computer Controlled Systems, Theory and Design*. Prentice Hall Information and System Sciences Series, 1997, ch. 2, pp. 34–37.

[9] W. L. Brogan, *Modern Control Theory*. Prentice Hall International Inc., 1991, ch. 3, pp. 74–79.

[10] B. Messner and D. Tilbury. Inverted Pendulum: State-Space Methods for Controller Design. [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=ControlStateSpace

[11] K. J. Aström and B. Wittenmark, *Computer Controlled Systems, Theory and Design*. Prentice Hall Information and System Sciences Series, 1997, ch. 2, p. 37.

[12] S. C. Bloch, *Introduction to Classical and Quantum Harmonic Oscillators*. John Wiley and Sons, 1997, ch. 1.

[13] T. S. Kaegi Werner, "VOSIM-A New Sound Synthesis System," *Journal of the Audio Engineering Society*, vol. 26, pp. 418–426, 1978.

[14] D. Smalley, "Spectromorphology: Explaining sound-shapes," *Organised Sound*, vol. 2, no. 2, p. 126, 1997.

Sensors2OSC

Antonio Deusany de Carvalho Junior
Universidade de So Paulo
dj@ime.usp.br

Thomas Mayer
Residuum
thomas@residuum.org

ABSTRACT

In this paper we present an application that can send all events from any sensor available on an Android device using OSC and through Unicast or Multicast network communication. Sensors2OSC permits the user to activate and deactivate any sensor at runtime has forward compatibility with any new sensor that may become available without the need to upgrade the application for that. The sensors rate can be adjusted from the slowest to the fastest, and the user can configure any IP and port to set receivers for OSC messages. The application is described in detail with some discussion about Android device limitations and the advantages of this application in contrast with so many others that are available on the market.

1. INTRODUCTION

In the context of applications, we had already transformed mobile devices into many instruments, synthesizers, and controllers. Some applications present nice user interfaces with lots of knobs, sliders, and buttons, and are able to send MIDI or OSC to other devices with a new value set on the interface, acting as a general controller. These applications can suit users' needs in most cases, but the sensors available are almost the same all the time, and the users cannot decide if they want to get the values. This condition limits the usability and makes all devices acting as if they were the same.

The devices are upgrading in so many ways and music software is not able to follow their velocity. It is based on the famous race of hardware and software development, and as soon as new hardware is on the market, lot of applications are upgraded to support this new hardware in order to keep old users that are expected to buy this new hardware. Musicians and performers that are waiting for the new technologies are included in this group of users that will buy high tech devices as soon as they become available in order to experience its advantages and integration with old applications or devices, like digital audio workstations (DAW) and synthesizers.

A solution for this problem is to create flexible applications based on backward compatibility and forward compatibility concepts. The first concept is probably the most aimed and used on mobile application development due the support libraries available. However, the latter is not so simple to provide because developers will probably not know the next evolutions of hardware and cannot always test new hardware beforehand.

On the other hand, the Android API follow some development aspects that can permit some inference about the next updates on the codes. The SensorManager lets you access all sensors in the same way, and the data dispatched by each sensor are always represented in an array of floats. Furthermore, the sensors have an ID that does not change between devices or system versions. Sensors2OSC is an application that take advantage of these conditions and provide not only backward compatibility but also forward compatibility for the users of mobile devices. Additionally, Sensors2OSC uses OSC [1] to name each sensor with a human readable prefix and can be received by many languages and programs.

In the next sections we are going to present some related works that have similar functionalities. A precise description of Sensors2OSC is presented on Section 3, where one can find how to use and configure the application for any performance. Some case study are discussed in the end in order to invite readers to try this application even just to experiment.

2. RELATED WORKS

We have lots of applications using OSC on Android devices. Most of them can be used to create nice interfaces and send updated values from the widgets and controls using any OSC patterns. The lack of support for many sensors available on Android devices is a special limitation on all of them, and it has been requested by many users at online forums. We are going to present the most used and discuss the support of sensors available on all of them.

One of the most famous OSC application for Android devices is the TouchOSC [1]. This application provides a control surface for interaction based on many controls. The user can use some default interfaces that are included in the app, and we have the TouchOSC Editor for many operating systems that can be used to create any layout and customize the TouchOSC application depending on your use. Regarding the interaction using sensors, this application only supports accelerometer sensor and the values are sent continuously if enabled at the settings, so the user cannot control any option of this sensor from the main screen.

At the time of this paper, the highest rated application on both iOS app store and Google Play is Control [2]. Al-

[1] TouchOSC: http://hexler.net/software/touchosc

Copyright: ©2015 Antonio Deusany de Carvalho Junior 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.

though similar to TouchOSC, Control uses web technologies and the user can create the interface using HTML, CSS, and Javascript. The interface design and configuration is described using JSON format, and it is possible to get the interface from the web or through OSC messages. The main limitation of this application is that it only supports accelerometer, gyroscope, and magnetometer (compass) sensors and cannot be easily upgraded due to its dependency of PhoneGap [2] libraries support for new sensors.

urMus [3] is an environment for mobile application development that also uses script language to create applications. This application has an event-handling for mobile sensor events, with lots of features that permits easy musical application development. It is possible to load scripts and also share code between users over the network in some applications developed with urMus [4]. This application supports the same sensors as Control, as well as GPS. The users can create any interface in Lua [3] language and send values through OSC. One can use any pattern of OSC prefix to send messages using urMus, the application can only understand its own defined prefix pattern.

MobMuPlat [4] is another example of mobile application that can be used to send the sensors values using OSC. Android devices can receive events accelerometer, gyroscope, orientation, compass, GPS, and joystick values. However, most of the options regarding the configuration of sensors are provided only for iOS devices.

Excluding TouchOSC, these applications are opensource and free. They represent the variety of available options for OSC users, but they are all limited to certain number of sensors. In the next section we will present a new solution for those who want to take advantage of all sensors available, independent of the device, including forward compatibility with every new sensor.

3. SENSORS2OSC

All mobile devices are coming with many new sensors, and the quantity and quality of the sensors are constantly increasing. On the other hand, it is not easy to use these sensors for musical interaction due to lack of good applications supporting all new sensors. We decided to create Sensors2OSC to solve these problems and permit the mobile device to be used in its full potential.

As the name may explain, the aim of this app is to get all sensors values and send through OSC. In this way, a performer can use any sensor from your mobile device to control applications on the same network. One can control many applications at the same time due to the support of OSC by many programming languages and programs, what implies in the requirement of adding only a receiver to a specific host and port in order to receive all messages sent by the mobile device.

Sensors2OSC has been created using Android Studio and gradle, the new pattern of mobile application development proposed by Google. The code is open source and has been tested in many versions of Android API, with help of many users. We support from Android API 8 (Froyo) to the newest ones, and we tried to design the app for forward compatibility regarding the sensors and the interface constraints. More details about the application are presented below.

[2] PhoneGap: `http://phonegap.com/`
[3] Lua: `http://www.lua.org/`
[4] MobMuPlat: `http://www.mobmuplat.com/`

3.1 Available sensors

Instead of limiting the application to only few sensors, we decided to load all available sensors at startup. This approach permits the use of all sensors available, and if a sensor will become available to the API in the future the user will have it enabled in the application. Some sensors have different number of values for each new event, and in this case we will have different OSC messages for each value.

Table 1 presents the OSC prefix for each sensor that can be available at Android devices at the time of this paper. The ID is the same ID used at Android API in order to identify the sensor. The prefix presented here are associated with the dimensions of the sensors. If a sensor has only one value, then the value will be sent as:

`<osc_prefix> "f" <value>`

If a sensor has multiple values, then the value will be sent as:

`<osc_prefix>/<coordinate> "f" <value>`

The coordinates are used by sensors with more than one dimension and their names differ depending on the number of dimensions as well. Sensors with 3 dimensions will have the coordinates `X`, `Y`, and `Z`; the ones with 4 dimensions are dispatched with `X`, `Y`, `Z`, and `cos`; and the sensors with 6 dimensions will have `X`, `Y`, `Z`, `dX`, `dY`, and `dZ`. These coordinate systems are modelled after the systems of currently available sensors.

The application is being updated for every new sensor added on Android API. We have also applied forward compatibility concept during the development. In this case, if you use this application on a device with a new sensor that is not already supported by our mapping, the ID of the sensor will be used as a prefix we are considering all new sensors with 6 dimensions. Another important point is that all sensor values are represented as float number from Android API, and that is the format used for all OSC messages sent through this application.

3.2 Main screen and controls

The main screen of the app has a main switch to start and stop sending values from enabled sensors using the configurations defined on the settings. All sensors available are presented on this screen and we have a switch for each coordinate of the sensor.

Once sending data for a coordinate is turned on, the app starts to send the OSC message defined for the specific sensor and coordinate, if the main switch is turned on. The user can turn off any coordinate at anytime. This functionality may be useful in many cases when you want to set a value and don't mind for next changes. As soon as you turn the switch on, the new events are going to be sent. It is important to notice that you won't receive the last state

ID	OSC prefix	Dimensions
1	accelerometer	3
2	magneticfield	3
3	orientation	3
4	gyroscope	3
5	light	1
6	pressure	1
7	temperature	1
8	proximity	1
9	gravity	3
10	linearacceleration	3
11	rotationvector	4
12	relativehumidity	1
13	ambienttemperature	1
14	magneticfielduncalibrated	6
15	gamerotationvector	3
16	gyroscopeuncalibrated	6
17	significantmotion	1
18	stepdetector	1
19	stepcounter	1
20	georotationvector	4
21	heartrate	1
22	tiltdetector	1
23	wakegesture	1
24	glancegesture	1
25	pickupgesture	1

Table 1. Mapping from Android sensors to OSC messages

if you turn on a sensor that had stopped to trigger events in the past, like the *tilt detector* that only trigger once and is disabled afterwards.

Figure 1 presents the main screen with some sensors. This screen is scrollable and you can see the osc_prefix to the right side of the sensor name. In case a sensor has more than one dimension, you will see the name of the coordinates that can be attached to osc_prefix. The switch in the top is the main switch and it has a fixed position, so the performer can attempt to stop or start the data transmission at any time, assuming full control at run time.



Figure 1. Sensors2OSC main screen.

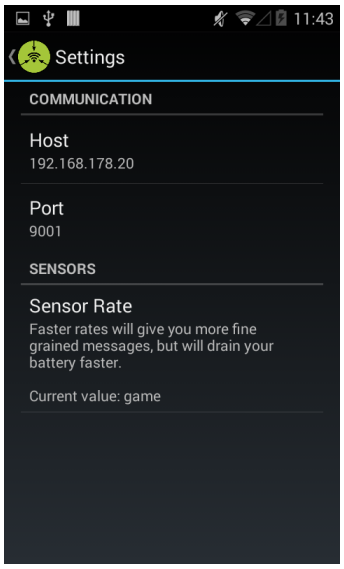


Figure 2. Sensors2OSC settings screen.

3.3 Network communication

The user can set the network details clicking at the settings button. It is possible to use Unicast or Multicast communication at this application. The mobile device with this application and the other device that will receive the values need to share the same network in case the user wants to send the OSC through Multicast. The range of Multicast address that can be used on IPv4 networks goes from 224.0.0.0 to 239.255.255.255 [5]. The Multicast has not been tested on IPv6 networks [6] with this application.

The Unicast communication requires a reachable IP address. It means that both devices need to be on the same network if the receiver is under NAT, or the receiver will need a reserved IP address on the Internet. Sensors2OSC supports both IPv4 and IPv6 address, and it is possible to use any hostname, like *localhost* or *example.com*.

The port number can be defined by the user at any time. However, it is recommended to use ports that are not reserved by any service. Some port number are assigned and some applications may have problems if the same port used for other purposes. The best port numbers to be used are the dynamic ports from 49152 to 65535 [7].

All of these settings described here are presented at Figure 2. It is also possible to define the sensor rate at this screen.

[5] Multicast addresses for IPv4 defined at IANA: http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml

[6] Multicast addresses for IPv6 defined at IANA: http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml

[7] Port number ranges defined by IANA: http://tools.ietf.org/html/rfc6335

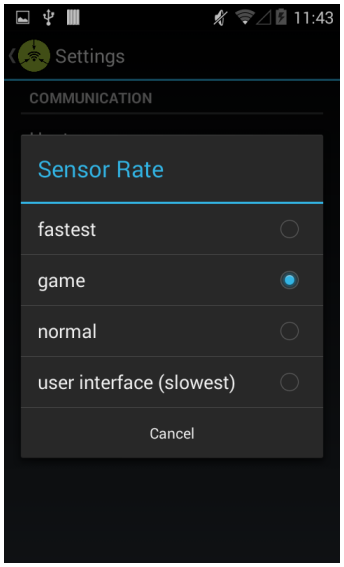


Figure 3. Sensors2OSC sensor rates screen.

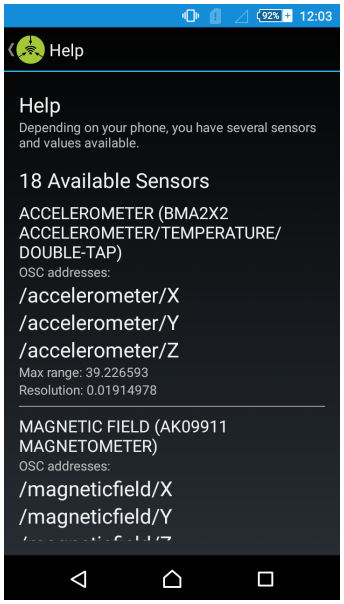


Figure 4. Sensors2OSC help screen at Sony Z3 Compact device.

3.4 Sensor rate

The sensor rate defines the number of events that is expected by the user from each sensor. This rate is not fixed because it depends on the sensor changes and the system interruptions.

Some sensors trigger new events very fast, e.g. accelerometer, gyroscope, orientation, and magnetometer. Even when the device is disposed on a table, values for these sensors change due to the noise from the environment and sensor characteristics. In this case, the sensor rate can help to adjust the frequency of each new event that is going to be sent.

Other sensors have different operation mode. The proximity sensor will only change its value when an object is close to the device screen, and is expected to change again when there is nothing near the sensor range. Some new sensors are named motion sensors and they will have a one-shot event. In this case, the sensor will send a unique value before being disabled after that, and the sensor rate may not affect the events. In this group we have the sensors defined for gestures like the wake, glance, and pick up gesture, and also the significant motion and tilt detector.

The available rates are presented on Figure 3. According to Android API specification, the fastest rate will try to send a new event as soon as possible. The other rates are not well specified but follow the scale presented on the Figure 3.

Depending on device and requirements, the user may change the rate. All events will be packed and sent through the network without buffering, and CPU usage, and in turn power consumption may increase at this point. If latency is crucial, higher sensor rates will detect and send value changes faster. Balance between number of active sensors and the sensor rate need some attention, but the user is allowed to do whatever is wanted.

3.5 Help menu

The help menu is a recommended first option in this application because of its information. At this menu the user can find the number, the name, and the model of available sensors, and also the OSC addresses used for each sensor value, the sensor maximum positive value, and the resolution.

An example of a possible visualization of this screen is presented at Figure 4. The device used in this case is an updated version of Sony Z3 Compact with Android API 21 (Lollipop). This device has 18 available sensors from 25 supported by actual Android API. The range and resolution of the sensors will depend on the model of the sensor and may differ between devices.

4. CASE STUDY

The users can receive OSC messages in many applications, and some mobile devices are becoming cheap. In this way, it is possible to control one single application with two different mobile devices at the same time using Multicast.

Imagine a Theremin controlled by two mobile devices. One device can send the values from accelerometer to control the amplitude and the other device sends values from the orientation sensor to control the frequency. Both devices are configured to use the same Multicast address and the same port number.

In this case, the user will hold each device in one hand and control the sound moving the device around the three dimensional axes. Another idea is to do the same thing with two participants, so each participant can hold a device and control the amplitude or the frequency. The users can also change the controls deactivating one sensor and activating the other one. In case both users send values from the same sensor, the result cannot be predictable due to the indetermination of packet ordering on the network

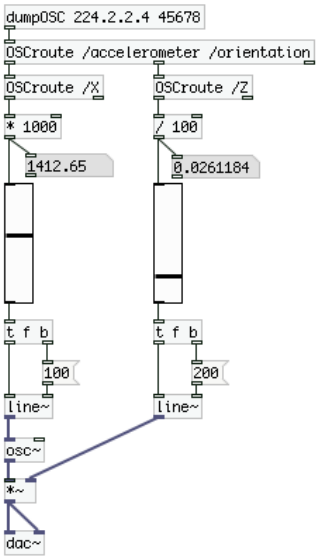


Figure 5. Case study of a Theremin controlled by two sensors and two devices.

while using Multicast and UDP packets.

A patch created in Pure Data to simulate this case study is presented on Figure 5. This patch receives OSC messages sent to Multicast address 224.2.2.4 and port 45678. The frequency of the oscillator is controlled by the X coordinate of the accelerometer sensor, while the amplitude is controlled by the Z coordinate of the orientation sensor.

An important information about the sensors are their ranges and resolutions. The user needs to verify these informations on Help menu before using the sensors at some application to adjust the values to its necessities. In this case study we had to adjust the values from both sensors to avoid bad values on the sound engine. The adjustment is optional and may vary also between the coordinates the sensors. For example, the orientation sensor have a range from 0 to 359 on X coordinate (the azimuth), -180 to 180 on Y coordinate (the pitch), and from -90 to 90 on Z coordinate (the roll). More information about the sensors can be found at Android developer web site [8].

5. CONCLUSIONS

In this paper we presented Sensors2OSC, an application that provide interaction using all sensors available on Android devices and OSC. This application is a sister of Sensors2PD [5], another application created by authors that sends sensors values to receivers on Pure Data patches that are loaded on mobile devices. The advances that we have with Sensors2OSC are the OSC format, Unicast and Multicast communication, sensor rates adjustment on the go, and a better nomenclature for prefix related to the sensors. Some of these options were suggested during the presentation of Sensors2PD in the last SMC conference.

All of these applications are member of Sensors2 [9], an attempt to create flexible applications that can send all events dispatched from sensors available on Android devices to applications using OSC, Pure Data, and other applications and languages like Csound, Processing, SuperCollider, and ChucK in a near future. At this point, we already evaluated the communication with Pure Data patches, SuperCollider, and Processing, however, we are planning performances using Csound, ChucK, and other systems.

A distant but important related work uses Csound as main *patching* language and accepts sensor values events at Android devices [6]. The users can load Csound orchestras and scores on mobile devices, uses interface options for controlling the sound, and uses accelerometer events for interaction. This application seems to be a proof of concept and is also an inspiration to the development of Sensors2CS using Csound as sound synthesizer in Android devices and receiving the values from all sensors.

Sensors2OSC is distributed with open source code that can be accessed on the repository [10]. The application has internationalization to English, German, and Portuguese. The authors are planning to publish the applications also on F-Droid [11], that is an installable catalog of Free and Open Source Software, and also on Google Play [12]. These resources may help users to learn how to use the sensors in their applications and can help non-technical users to install and use the applications without problems.

6. REFERENCES

[1] M. Wright, "Open sound control: An enabling technology for musical networking," *Org. Sound*, vol. 10, no. 3, pp. 193–200, Dec. 2005. [Online]. Available: http://dx.doi.org/10.1017/S1355771805000932

[2] C. Roberts, *Control: Software for end-user interface programming and interactive performance*, 2011.

[3] J. W. Kim and G. Essl, "Concepts and practical considerations of platform-independent design of mobile music environments," in *Proceedings of the International Computer Music Conference*, 2011, pp. 726–729.

[4] S. W. Lee and G. Essl, "Communication, control, and state sharing in networked collaborative live coding," in *Proceedings of 14th International Conference on New Interfaces for Musical Expression (NIME)*, Goldsmiths, University of London, London, UK, June 2014.

[5] A. D. de Carvalho Junior, "Sensors2PD: Mobile sensors and WiFi information as input for Pure Data," in *Joint Conference: 40th International Computer Music Conference and 11th Sound and Music Computing Conference*, 2014.

[6] V. Lazzarini, S. Yi, J. Timoney, D. Keller, and M. Pimenta, "The mobile csound platform," in *Proceedings of the International Computer Music Conference*, 2012.

[8] Android developer: `http://developer.android.com/`

[9] Sensors2: `http://sensors2.org/`

[10] Online repository of Sensors2OSC: `https://github.com/SensorApps/Sensors2OSC`

[11] F-Droid: `https://f-droid.org/`

[12] Google Play: `https://play.google.com/`

Cooperative musical creation using Kinect, WiiMote, Epoc and microphones: a case study with *MinDSounDS*

Tiago Fernandes Tavares, Gabriel Rimoldi, Vânia Eger Pontes, Jônatas Manzolli
Interdisciplinary Nucleus of Sound Communication
University of Campinas - Brazil
tiago@nics.unicamp.br

ABSTRACT

We describe the composition and performance process of the multimodal piece *MinDSounDS*, highlighting the design decisions regarding the application of diverse sensors, namely the Kinect (motion sensor), real-time audio analysis with Music Information Retrieval (MIR) techniques, WiiMote (accelerometer) and Epoc (Brain-Computer Interface, BCI). These decisions were taken as part of an collaborative creative process, in which the technical restrictions imposed by each sensor were combined with the artistic intentions of the group members. Our mapping schema takes in account the technical limitations of the sensors and, at the same time, respects the performers' previous repertoire. A deep analysis of the composition process, particularly due to the collaborative aspect, highlights advantages and issues, which can be used as guidelines for future work in a similar condition.

1. INTRODUCTION

MinDSounDS is an multimodal piece for computer, movement, WiiMote, flute, Brain-Computer Interface (BCI) and images, world premiered at the Generative Arts 2014 conference (December 2014, Rome). It was composed to be controlled live by a group of performers by means of a network of consumer sensors. The work is based on previous piece, namely Re(PER)Curso [1], and illustrates how the aesthetic experience can be related to an organization that emerges from the interaction between the performers and a virtual environment.

MinDSounDS narrates the story of a virtual avatar – a humanoid projection on screen – that learns the movements of a human dancer and builds its own movements. This process is mediated by human performers, which interact among themselves and with the virtual environment. As the avatar builds its own movements, it also interacts with humans, thus actively joining the performance group.

We defined that the piece would be composed by the whole group, without a prior agreement on its content or its language. Each of the involved musicians, which are the authors of this paper, had their own set of skills and their own artistic intentions towards what *MinDSounDS* should become. Communication in these conditions has proven essential, and, at the same time, not trivial, as it is easy to find misunderstandings of several natures.

We conducted a collaborative composition process for related to each one of these instruments, which gave rise to specific problems and advantages related to group work. Prior work by Cornacchio [2] has discussed issues related to group musical composition in music classrooms, and we have noticed some similarities to our process. However, our process was not bounded to a clear goal or musical language, which gave rise to specific difficulties and discussions.

Through this process, we developed the piece as an expression of the group's multidisciplinarity, which reflected in the sensor network multimodality. Because of the group's cooperation, we were able to build interesting mappings between the sensors inputs and their sonic and visual representations. The use of different sensors was a natural result of the process, as each of them had an important artistic contribution to the piece.

The group's composition proposal allowed the development of an iteractive method for composing mappings between gestures and media, which was especially important in the case of the Kinect. Prior art mainly focused on mappings defined by the composer and delivered as instructions for the performer [3, 4] or in processes in which the composer and the performer are the same person [5–7]. In *MinDSounDS*, the composition process considered a dance movement repertoire as part of the performance, thus composing a virtual environment that enhanced the movement possibilities of the performer.

The result of our process also presents sensible differences from prior art. We do not design a virtual environment that emulates real interactions [6, 8], and, at the same time, we do not design an arbitrary virtual instrument [3, 4] or interactive control of sound effects [5, 7]. Instead, we use motion data to augment the expressive possibilities of the dancer, respecting their original repertoire and progresively exploring new expressive aspects.

Our approach towards the Epoc was also significantly different from related work using BCI. We have found that previous work has largely focused on the sonification of brain waves [9–11], which means that sound is generated using voltages measured in the scalp as raw material. In this approach, the musical intentions of the user are disregarded during the composition process, even if they can be

Copyright: ©2015 Tiago Fernandes Tavares, Gabriel Rimoldi, Vânia Eger Pontes, Jônatas Manzolli 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.

indirectly controlled by training.

In other approaches, BCI was used to trigger events, attempting to mimic actions that could be performed using the body [12, 13]. However, state-of-the-art BCI systems yield several false negatives and false positives in intentional triggers. Therefore, previous work has used post-filtering techniques like offline usage [14], beat synchronization [12] and low-pass filtering [13] to overcome these difficulties.

We overcame this problem by incorporating the BCI concept into the piece construction. The BCI device was responsible to mediate a high-level process whose fine details were controlled by the dance performer and a timer. Therefore, we incorporated the BCI in a context in which false positives and false negatives would not cause drastic consequences to the performance.

The remainder of this paper is organized as follows. Section 2 describes the implementation of the sensor network. Section 3 discusses the advantages and drawbacks found in the composition and performance processes. Last, Section 4 brings conclusive remarks.

2. SENSORS

MinDSounDS relies on the interaction between performers and a virtual environment by means of sensors. This interaction took place by means of mapping between sensor data and sound and visual representations. The process of building these mappings was an important part of the construction of the virtual environment.

An important aspect of *MinDSounDS* is that it aims at creating specific causalities between inputs and outputs, that is, avoids generative processes that are not controlled – or, at least, controllable – by the performers. This comes from the group's perception that the audience should be able to understand the relation between the perforers' movements and the audio and video responses. Thus, our composition process greatly acounted for consistency between actions and their mappings.

During the composition process, we used different kinds of sensors to provide musicians with diverse expressive possibilites. As depicted in Figure 1, each sensor data is used in a different context, and interferes with other sensors, jointly controlling synthesis processess. Below, we present a throrough explanation of the interaction related to each sensor.

We used a motion sensor to capture dance movements of a performer. Different movements should lead to different sonic responses, but it is initially unclear how to make these responses meaningful to the piece's intention and the performer's repertoire. In Section 2.1, we describe how the process of building this mapping was conducted to mediate between these aspects.

A game controller with an accelerometer emulate virtual bells. Due to the computational nature of these bells, we found issues concerning the sensor's sensitivity. Also, as we describe in Section 2.2, the accelerometer was added with dynamic filtering capabilities, increasing the expressive potential of the controller.

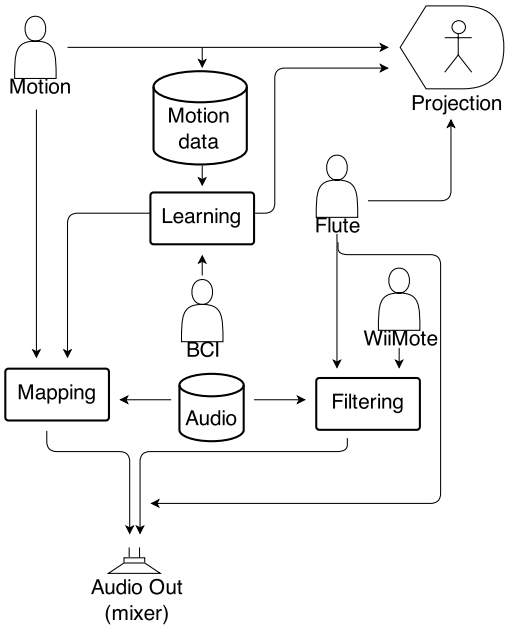


Figure 1. MindSounds interaction diagram, depicting how each sensor data interacts with the others.

We also explored the Brain-Computer Interface (BCI), which translates voltages between key points in the user's scalp to triggers thay may be used as game controllers. The BCI has been increasingly used in musical contexts for different purposes. We have developed a particular musical language, suitable for both the purposes of the piece and the characteristics of the BCI, which is described in Section 2.3.

Last, Music Information Retrieval (MIR) techniques allowed using an acoustic flute as a controller. The information derived from these techniques was bounded to the control of characteristic of a video projection, thus the composition process raised new possibilities, as well as specific restrictions. We describe this process, and its results, in Section 2.4.

2.1 Kinect

The Kinect is a motion capture sensor developed for gaming purposes. Using specialized software, it is possible to obtain a tri-dimensional position (using $p = (x, y, z)$ triples) for each of the body's limbs (elbows, knees, hands, etc.) at a frame rate of 30 Hz. The positions of limbs were interpreted as related to the performer's torso, namely the kinesphere.

The kinesphere was used due to the performer's dance repertoire, which comprises mostly arm and leg positionings as a form of expression. The kinesphere allowed a more precise acquisition of these movements, while at the same time disregarding jumps and dislocations through the stage. From a purely technical view, this also added the advantage of reducing the time required to calibrate the sensor to different venues.

There is no theoretically best mapping between limb movements and controls, as this depends on the perform's movement repertoire, sound designer's technique repertoire and

the piece's intention. Since the piece's creative intention was unclear at early stages of the composition process, the mapping's construction comprised several iteractions between the performer and the sound designer, assisted by the remaining of the group. In this process, mapping proposals were presented and discussed, leading to a final decision.

The mappings we have found more interesting for the piece are shown in Table 1, but it is very likely that they will be re-built in other future work. This will happen not because they are not good in any sense, but because they are the result of a composition process, which will, inevitably, happen again. However, we have developed useful strategies for finding this mapping, which may be employed again in the future.

Movement	Control
Hands around kinesphere	Spatialization (panning) Sample selection Video control
Distance between hands	Pitch Sample selection
Feet velocity	Sound intensity
Relative feet position	Granulation control

Table 1. Mapping of gestures to controls using the kinect

We have found that it may be useful not to map all movements to audiovisual representations. This gives the performer a greater freedom to develop a more natural dance sequence, including movements whose contribution to the piece is solely visual. This means that, while the motion sensor enables live control of computer-based sound and video, it may also constrain dance movements, potentially harming the performance.

The same hold for another decision, regarding the nature of the movements that will be mapped. Nowadays, there exists technology that allows mapping specific dance moves (for example, a spin) to an event trigger. We did not want to use this because we wanted to allow an exploration and improvisation process to be part of the dance performance.

Therefore, we opted to use more general movement parameters as controls. An example that worked was the panning control, done by the position of the hands around the kinesphere. This mapping allows a great variation on the movement, for example, regarding the performer's elbows and shoulders, while resulting in the same controls.

We have also noted that discrete controls that trigger to specific movements should be used carefully. Triggers are efficient for some purposes, like selecting sound samples, but they may restrict the performer's movements in order to avoid false positives or false negatives. Thus, their extensive use may inhibit the performer's fluency.

Continuous mappings, on the other hand, are unable to trigger discrete events. In our composition process, they were easier to incorporate into the dance performance, because they were felt more as movement suggestions than as coreographed steps. Thus, we were careful to maintain balance between discrete and continuous movement mappings.

By using movement velocity as a sound intensity control, we were able to map a perceived visual effort to a perceived auditory effort. This helped on our goal of allowing the audience to understand the mapping process, as it emulates the behavior of acoustic instruments. In these instruments, a stronger effort usually reflects on a stronger sound, allowind the control of event dynamics, which are important for expressive performances.

It is also important to note that hese mappings were not all used at the same time, but scattered on particular movements of the piece. Each of them induced a different exploration of the sonic space by the performer, leading to the use of a different repertoire of gestures, sounds and visuals, as highlighted in Figure 2. Thus, although we aimed at not creating an invasive and restrictive virtual environment, the interaction possibilities inevitably favoured particular movements over others.

The process of finding mappings, gestures and sounds that would fit the purposes of the piece demanded a great amount of interaction between all members of the group, especially the sound designer and the dancer. During this process, one of the greatest problems we faced was due to the abscence of a language that could consistently and efficiently convey sonification ideas, which lead to many misunderstandings. Another problem is that the implementation of a new mapping proposal was very time-consuming, as it demanded understanding the movement and translating it into code.

We used a similar interactive approach to develop mappings and sonifications for the WiiMote. The nature of the controller lead to the development of different algorithms. The process regarding the WiiMote is described in the next section.

2.2 WiiMote

The WiiMote is a handheld console that contains nine buttons and a three-axis accelerometer, which were mapped according to Table 2. Using third-party software, it is possible to acquire the accelerometer data at 100 Hz, as well as triggers related to pressing the buttons. In comparison to the Kinect, it has a faster response, but also yields significantly more noise.

Input	Control
Button A	Enable percussion
Slap gesture	Use percussion
Directional buttons	Record data for adaptive filter
Accelerometer	Control filter interpolation
Button B	Use filter

Table 2. Mapping of inputs to controls using the Wiimote. They are further explained along this section.

The device was used to control a virtual percussion device. This functionality could be enabled or disabled through one of the buttons, and, if enabled, triggered by using the device as a drumstick in the air, in a *slap* gesture.

Detecting a slap gesture was done detecting acceleration values above a pre-defined threshold in any axis. The pitch



Figure 2. Examples of the interaction in two different movements of the piece. Different movements were used to control different visual representations.

and roll parameters during the *slap* gesture controlled filters that would modify the percusive sounds. Thus, different angles of attack resulted in sounds with diverse spectral content.

The controller was also linked to an interpolated filter derived from ambient sound. This application was based on recording sound samples captured from microphone and interpolating them, using the result as the impulse response of a FIR filter. In our piece, the we acquired sound samples from the acoustic flute, and applied the resulting filter to pre-recorded vocal samples that controlled the soundscape.

To control this functionality, four buttons were used to trigger recording in four different audio buffers. The resulting impulse response would correspond to their weighted sum, in which the weights were controlled by the pitch and roll of the WiiMote. While a fifth button was pressed, the system would apply the filter to the audio output.

Hence, a variable, interpolated filter was developed. Its control using the accelerometer quickly became intuitive, with the advantage of preserving the presencial action of the performance because of the live movements of the performer. The hardware has show to be reliable and fast for low-level audio control, which was not the case for all sensors, as will be seen.

2.3 Epoc

The Epoc is a consumer device that provides a Brain Computer Interface (BCI). It consists of several electrodes and an accelerometer, which provide readings of the Electroencephalogram (EEG). Its software suite works under the assumption that similar thoughts correlate to similar EEG signals, hence allowing memorizing mental states and ultimately providing the ability to use thoughts to control software.

We have found two main problems with the use of the device, which are shared by many BCI approaches. The first problem is the instability of training – the system has to be calibrated each time it is used, and the user must keep a clear mind during the use. The second is the high amount of false positives and false negatives.

These limitations were avoided by using the device in a context that allows for errors without drastic consequences to the performance. This means that we developed a musical paradigm in which these false negatives and false positives would be part of the musical discourse, instead of undesireable artifacts. For this reason, we used the BCI device for the control of high-level parameters.

In the musical context, the BCI was used in a piece movement in which the avatar is learning the movements of the dancer. These movements are recorded directly from the dancer, in previous movements. The learning process is represented by the application of a recombination algorithm.

The recombination algorithm takes as input the recordings of the dancer's limb position. Then, it applies a random time-shift in each stream, thus creating combinations of limb positions that are impossible to be performed by a human being, but are rendered on the screen creating a perceptually weird form. Through the learning process, the amount of time-shift allowed in each stream is reduced, which makes the rendered form gradually assume humanoid appearance, leading to the perception that the avatar is slowly immitating the performer's movements.

In this context, the BCI device was used to trigger a next step in the learning process, corresponding to a new maximum value for the random time-shift. The next maximum time-shift is defined as the previous value minus a fraction of the elapsed time since the start of the previous step. The beggining of each step is also marked by the sound of a bell.

As a result, it was possible to estimate the duration of the movement and its possible outcomes, which was useful for planning the interaction with the other musicians. The detail-level of the piece, that is, the time when each learn-

ing step would be triggered, could be actively controlled by the musician. This way, we were able to overcome the limitations of the device while still using it in a meaningful way.

2.4 Computational Ear

Using MIR techniques, we were able to use an acoustic flute as a musical controller. Audio was acquired from the instrument using a microphone, and processed yielding spectral and temporal features of sound. Later, these paramenters were used to modify the visual part of the piece.

We chose to use two audio features to control continuous values in video processing. The Chroma feature determines a range of hue while the Loudness determines the luminosity of rendered textures on video. This allowed mapping note classes to projection colors, which was done arbitrarily.

However, the decision of using audio for this purpose implied in other artistic decisions. The chosen features (Loudness and Chroma) only make sense in the context of sound with defined pitch. This means that, while this controller was used, the performer should explore sonorities in which pitch remain as a main parameter.

The technical issues presented in this section had a deep impact on the final format of the composition. They were an important part of the composition process through which we obtained a sensor network aimed at building the concept of Presence in the context of the piece. Further discussion regarding this process will be conducted in the next section.

3. DISCUSSION

The process of composing *MinDSounDS* was a cooperative process that integrated both the artistic and the technological points of view. As a result, we developed significant advances impacting both the final outcome – the piece itself – and the conduction of its composition process. Hence, we believe that *MinDSounDS* can be part of a base repertoire in future work.

In the cooperation process, we found problems that may arise in diverse environments. Since there was no prior guideline to follow, the group struggled to make *MinDSounDS* an artistic expression that comprised the desires of all musicians. This is partially inevitable, and an important part of the cooperation process, but we were able to detect some guidelines that may be useful in the future.

Group time management, in our process, was poor, which meant that in several ocasions there were scheduled activities that did not require the presence of some group members. This lead to a waste of time and contributed to the loss of focus. Although we were aware of this, it was not an easily avoidable situation because the objectives of each activity were not clear during the process.

Another issue related to time regards the fact that the musician that would perform with the sensor device was not the composer of the corresponding interaction. This implied in an interactive composition process in which there are two oposing points of view, one related to building a lean, usable system and the other related to constructing an artistically meaningful interaction. We adopted the solution of composing partial mappings, as discussed before, but this process had its own difficulties.

The interaction between the composer and the performer consisted of taking proposals by both musicians and trying to explore its possibilities (for the performer) or trying to implement it (for the composer). As the perform explores possibilities, new proposals arise, and the same holds for the implementation of the proposals by the composer. The first issue regarding this process involved finding proposals that could integrate the musical background of each musician, as well as the piece's proposal.

Another problem was related to the long time required for implementing proposals by each composer. This inevitably generated long periods of idle time, which had a negative impact on activity sessions and, ultimately, in the interaction process. Therefore, we detected a clear demand for a framework allowing these interactions to be built faster, so that the exploration and composition process may follow the musicians' pace.

We also faced problems regarding the construction of the piece's artistic proposal. Since we did not have a clear idea of what we were trying to implement, or even the musical language that we would follow, the final result emerged from our interactions. Following this proposal is advantageous in the sense that it allows experimenting a broader range of techniques, but also prevents a deeper individual experimentation on particular issues.

The indefinition of the expected result of a process is a known and well-studied issue both in music – for example, in improvised performance – and computer science – as there are specific software engineering technicques that deal with it. The case composing *MinDSounDS* is different from an improvised performance because the group was also responsible for building the musical instruments, and, moreover, each instrument had a deep impact on the others. Also, it was not the same as a software engineering case because the problem was not supplying functionalities for a client's demand, but building the demand from an initial, abstract idea.

Thus, it became clear that we lacked an effective process for communication of repertoire, expectations and analysis of the results. This points to a direction for future work, which is studying issues related to composing music in groups without a prior style agreement. In this sense, it is important to preserve artistic freedom and the feeling of participation, while introducing guidelines for cooperation.

Nevertheless, the piece was succesfully composed and presented, and is now a unit of structural cohesion. This property emerged from the composition process, generating a unique piece in which all parts involved presented important contributions. Also, this process was an important step towards understanding musical cooperation, and its analysis will have great impact on future work.

The mappings and algorithms we employed in the piece were also the result of this cooperation process. This process was different from two very frequent ones: the solo

musician that is both the composer and the performer, and the cascade workflow in which the perform executes instructions from the composer. Thus, composing lead to a greater understanding of each musician's role in the piece, and, from this point of view, this process was more important than the final result.

4. CONCLUSION

We described the process of composing the multimodal piece *MinDSounDS*, highlighting the technological and artistic issues that arised. We showed how each sensor was applied on the control of specific parts of the piece. Moreover, we discussed how the process of finding these mappings was relevant to the piece.

The piece was composed in a cooperative process, without the pre-definition of a final objective or an explicit artistic language. This gave rise to a series of problems, which were handled by the group and had a deep impact on the composition process. Finally, we finished and presented the piece, and also learned on aspects that could be improved in future work.

We take special care on presenting how each sensor cooperates to the piece. We discuss the algorithms and technological limitations of each sensor. As a result, the use of each sensor becomes differentiated, improving its contribution to the final artistic result.

Addressing technical and artistic limitations, especially the cooperation issues during composing and rehearsing, present a clear direction for future work. This direction should point at developing protocols that allow a creative interaction between composers and performers while providing and effective use of the team's time. These aspects are often conflicting, but this is a problem that must be studied in order to make cooperative composition a more efficient process.

Acknowledgments

The authors thank the Brazilian agencies FAPESP and CNPq for funding this research.

5. REFERENCES

[1] A. Mura, J. Manzolli, P. F. M. J. Verschure, B. Rezazadeh, S. L. Groux, S. Wierenga, A. Duff, Z. Mathews, and U. Bernardet, "re(per)curso: An interactive mixed reality chronicle," in *SIGGRAPH*, Los Angeles, 2008.

[2] R. A. Cornacchio, "Effect of cooperative learning on music composition, interactions, and acceptance in elementary school music classrooms," Ph.D. dissertation, Graduate School of the University of Oregon, 2008.

[3] M.-J. Yoo, J.-W. Beak, and I.-K. Lee, "Creating musical expression using kinect," in *Proceedings of NIME*, 2011.

[4] A. R. Jensenius, "Kinectofon: Performing with shapes in planes," in *Proceedings of NIME*, 2013.

[5] G. Odowichuk, S. Trail, P. Driessen, W. Nie, and W. Page, "Sensor fusion: Towards a fully expressive 3d music control interface," in *Communications, Computers and Signal Processing (PacRim), 2011 IEEE Pacific Rim Conference on*, Aug 2011, pp. 836–841.

[6] S. Sentürk, S. W. Lee, A. Sastry, A. Daruwalla, and G. Weinberg, "Crossole: A gestural interface for composition, improvisation and performance using kinect," in *Proceedings of NIME*, 2012.

[7] S. Trail, M. Dean, T. F. Tavares, G. Odowichuk, P. Driessen, A. W. Schloss, and G. Tzanetakis, "Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect," in *Proceedings of NIME*, Ann Arbour, Michigan, U.S.A., May 2012.

[8] M.-H. Hsu, W. Kumara, T. Shih, and Z. Cheng, "Spider king: Virtual musical instruments based on microsoft kinect," in *Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), 2013 International Joint Conference on*, Nov 2013, pp. 707–713.

[9] E. R. Miranda and B. Boskamp, "Steering generative rules with the eeg: An approach to brain-computer music interfacing," in *Sound and Music Computing*, 2005.

[10] ——, "Toward direct brain-computer musical interfaces," in *New Interfaces for Musical Expression*, 2005.

[11] S. Mealla, A. Väljamäe, M. Bosi, and S. Jordà, "Listening to your brain: Implicit interaction in collaborative music performances," in *New Interfaces for Musical Expression*, 2011.

[12] S. L. Groux, J. Manzolli, and P. F. Verschure, "Disembodied and collaborative musical interaction in the multimodal brain orchestra," in *New Interfaces for Musical Expression*, 2010.

[13] T. Mullen, R. Warp, and A. Jansch, "Minding the (transatlantic) gap: An internet-enabled acoustic brain-computer music interface," in *New Interfaces for Musical Expression*, 2011.

[14] B. Hamadicharef, M. Xu, and S. Aditya, "Brain-computer interface (bci) based musical composition," in *Cyberworlds (CW), 2010 International Conference on*, Oct 2010, pp. 282–286.

THE "HARMONIC WALK" AND ENACTIVE KNOWLEDGE: AN ASSESSMENT REPORT

Marcella Mandanici, Antonio Rodà, Sergio Canazza, Federico Altieri
Dept. of Information Engineering, University of Padova
{mandanici, roda, canazza, altieri}@dei.unipd.it

ABSTRACT

The *Harmonic Walk* is an interactive, physical environment based on user's motion detection and devoted to the study and practice of tonal harmony. When entering the rectangular floor surface within the application's camera view, a user can actually walk inside the musical structure, causing a sound feedback depending on the occupied zone. We arranged a two masks projection set up to allow users to experience melodic segmentation and tonality harmonic space, and we planned two phase assessment sessions, submitting a 22 high school student group to various test conditions. Our findings demonstrate the high learning effectiveness of the *Harmonic Walk* application. Its ability to transfer abstract concepts in an enactive way, produces important improvement rates both for subjects who received explicit information and for subjects who didn't.

1. INTRODUCTION

A tonal composition is perceived by listeners as a sequence of discrete pitch-events [1], matched with an underlying harmonic background. As soon as the listeners recognize that pitches belong to the same chordal entity, they group them into subsequent musical units, which subdivide the melody after metrical and harmonic rules. This process is called "melodic segmentation" and represents the basis of tonal music knowledge. Research in the field of music psychology has offered wide evidence that unlearned adults and children from the very early age of 4-5 years [2] have an implicit knowledge of the elementary harmonic organization.

Thus, employing a geometric interpretation of the spatial qualities of melodic segmentation and harmonic chord space, we use enactive [1] and spatial knowledge to reach the heart of the complex domain of tonal music composition and to manipulate its contents in a creative way. A preliminary study for the *Harmonic Walk*'s environment and a thorough description of the system architecture and theoretical background can be found in a prior publication [3], while previous assessment sessions showed the application's success rate in simple orientation tasks or in more difficult cognitive assignements such as melody harmonization [4] and melodic segmentation [5].



Figure 1. The *Harmonic Walk* while being tested by a high school student at the Catholic Institute "Barbarigo", Padova.

1.1 Enactive learning in the *Harmonic Walk*'s environment

The *Harmonic Walk*'s design is grounded on the spatial characteristics of two fundamental features of the tonal harmony language: The melodic segmentation and the tonality chord space. This musical features are interpreted geometrically [6] and transformed into masks employed to partition the floor surface, which is the actual interface between the user and the application's contents. The melodic segmentation is interpreted as a sequence of square blocks put along a straight line (see Fig. 2), while the tonality chord space (see Fig. 3), formed by three primary and three parallel roots, is laid on a six partitioned circular ring [5].

[1] Enactive knowledge is deeply linked to the experience of doing something. It provides implicit information about a specific task, allowing the subject to perform also very complex actions without having an explicit knowledge about them.

Copyright: ©2015 Marcella Mandanici, Antonio Rodà, Sergio Canazza, Federico Altieri. 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.

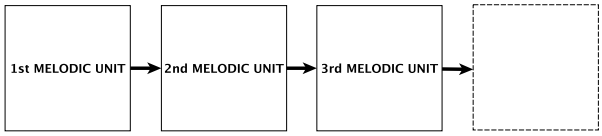


Figure 2. The geometrical representation of a tonal melody units sequence.

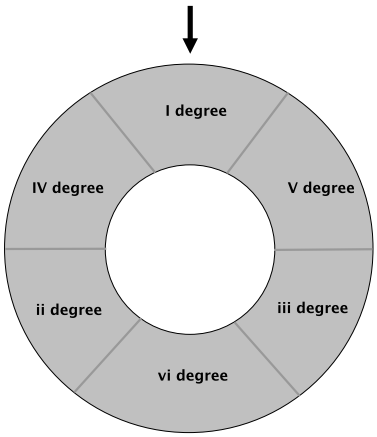


Figure 3. The geometrical representation of a tonality six roots harmonic space.

When following the first spatial schema (the straight line), locomotion is guided from block to block following the path displacement. The cognitive map of the path is easily understood, thanks to the fact that the sound file stops playing as soon as the melodic unit ends. The concatenative nature of the path invites the users to follow the melodic line and to move a step forward in time with the arrival of the next unit. But, in the second spatial schema the user is presented to the circular ring area, where only the starting point is marked and where various route possibilities are allowed. The first thing a user can do in this situation is to explore the sound of the six regions, containing the six roots of the tonality harmonic space. As soon as s/he discovers the chord of the first sounding area, s/he registers the first landmark, beginning so to feed the circular mask's cognitive map. Users can ignore every explicit notion about harmony, while they enactively learn how to move on the mapped interface. The sensorimotor information about the chords displacement coupled with the feeling of their harmonic functions guides them towards the accomplishment of the melody harmonization task.

The aim of this paper is twofold. Firstly, we want to test if the application is really efficient to drive the users to harmonize a tonal melody with and without explicit explanation from the teacher; secondly, we point to the cognitive aspect of the experience in the *Harmonic Walk* environment, trying to discover what users actually learn after one or more trials and training sessions. Assuming for true that users are endowed with some degree of implicit knowledge about tonal melody and harmony, we wonder: Is it possible to make this unconscious knowledge to emerge and to become a real ability? How important is the role of explicit, previously delivered information? Is enactive experience stronger than explicit information? And if so, in what domain? These are crucial points to foster the full body interaction style in learning environments, to make it really useful in educational curricula and to try to integrate it in the actual teaching practice.

1.2 Related Work

The *Harmony Space* project of the Music Computing Lab (University of Stanford), was elaborated in 1993, not only on a desktop interface [7] but also in a physical environment, employing a floor projection and a camera tracking system [8]. More recent systems like *Isochords* [9] or *Mapping Tonal Harmony* [10], are very complex environments which improve musical structure consciousness at a very high degree of knowledge. Other more intuitive approaches are offered by the *PaperTonnetz* [11] and the *Harmony Navigator* [12], where the chord selection, supported by a corpus based statistical model, is operated by hand gestures in the 3D space around the user.

The authors of [13] present some results from assessment sessions in the *SMALLab* environment,[2] a semi-virtual learning environment where users move freely producing visual and audio output. Their results suggest that receiving regular instruction before the exposure to the application's environment, significantly improves the learning of content. Moreover, they tried to test if an embodied experience in a reality based physical environment could lead to greater learning gains if compared to the same learning session in a desktop environment. Their findings confirmed a significant, but equal improvement for both groups, without the expected enhancement in favor of the embodied experience group. An interesting survey about augmented reality learning applications is presented in [14], where researchers try to measure the learning gain achieved by the use of augmented reality applications in education. They reported an effect size to student performance expressed by a Cohen's d value of 0.56.[3] Anyway, this result is subject to a wide variety due to the many important differences in the ways of use of augmented reality as well as in experimental design.

This article is organized as follows: In Section 2 we describe the system architecture, with the *Zone Tracker* application and the *Max/MSP* modules for sound production. Section 3 presents the test's aim and organization, while quantitative results, data gathering and meta-analysis are discussed and compared between the two assessment test sessions in Section 4. Conclusion and further work follow in Section 5.

[2] *SMALLab* (Situated, Multimedia Arts Learning Laboratory `http://smallablearning.com/`) has been developed by and a transdisciplinary group at the Arizona State University School of Arts, Media and Engineering in 2010.

[3] Cohen's d is defined as the difference between the control and treatment means divided by the pooled standard deviation (root mean square of the two standard deviations). After Cohen's (`http://en.wikipedia.org/wiki/Effect_size#Cohen.27s_d`) effect size table (`http://www.uccs.edu/lbecker/effect-size.html`), we define an effect size small if $0 < d \leq 0.2$, medium if $0.2 < d \leq 0.5$, and large if $d > 0.5$.

2. SYSTEM ARCHITECTURE

The *Harmonic Walk* architecture combines two software modules: The *Zone Tracker* application with video analysis algorithms and masks for surface division, and the *Max/MSP* patch containing audio files for sound output.

A camera mounted on the ceiling follows the user while moving on the underlying surface within the camera range. These data arrive to the *Zone Tracker* application, which subtracts the background obtaining a well shaped image blob. Comparing the blob's position with a previously stored mask, the system identifies the zone occupied by the user and sends this information through the *OSC* protocol [15] to the *Max/MSP* patch.

For our tests we use two different masks, corresponding to the straight path of the musical unit sequence and to the circle of the six chords of the harmonic space. The first mask subdivides the tracked zone in 5x5 squares, each one with a side of about 60 cm, roughly corresponding to the distance of a human step; the second is the circular ring mask depicted in Fig. 3.

For each tonal composition the *Max/MSP* patch stores :

1. the music audio file, segmented according to the harmony changes;
2. 6 audio files reproducing the chords of the song's tonal space, played with the same rhythm and timbre of the original composition.

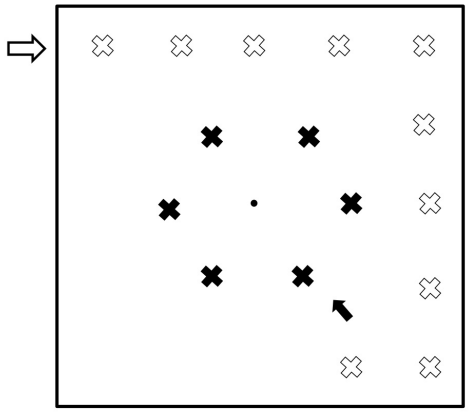


Figure 4. Visual tags of the straight and circular path of the *Harmonic Walk*.

The *Harmonic Walk* user interface appears to the user as depicted in Fig. 4. Along its borders there is the straight line of the musical unit sequence, while the circular ring is put at the center of the rectangular area. [4]

The user's paths are identified respectively through white and black crosses, with an arrow marking the beginning of each path. [5]

[4] We represent the masks as square and circular ring also if, due to camera pixel shape, the actual masks are respectively distorted in a rectangle and in an oval ring.

[5] The tags both in the straight line and in the circular ring are not descriptive, i.e. they are not marked with the syllables of change or with the chord names, but they have the function to indicate to the user the right position to be occupied on the application's surface.

3. THE *HARMONIC WALK*'S ASSESSMENT

The *Harmonic Walk* assessment tests were organized at the Catholic Institute "Barbarigo" in Padova, the seat of a music high school, as well as of various other kinds of schools. The assessment test was organized in two subsequent stages (December 2014 and January 2015), with the aim of collecting experimental data about the impact of the application's utilization with respect to four different subject categories selected among high school musicians and non-musicians students (see Table 1). In the first stage the subjects experienced the application's tasks without any information about the music contents; in the second stage one half of the group could get some information in a 1 hour demonstrative lesson, while the other half not. We considered the first test's results as control and the second test's results as experimental data. The results comparison is presented and discussed with respect to the success in the main application's task (the melody harmonization) and to the level of knowledge acquired in three selected aspects of the application's musical content.

3.1 Subjects

A total number of 22 high school students between 16 and 20 years old, took part in the first and second test. In the first test the students were equally subdivided into two different groups: one musically trained and the other not. The first group was taken from classes belonging to the music high school, with specific instrumental and music theory programs. The second was chosen from classes belonging to various kinds of high schools with no music programs in their curricula. In the second test we randomly divided each group into two subgroups of 5 and 6 subjects. Then, we selected one subgroup of musicians and one subgroup of non musicians to assist to a 1 hour demonstration lesson and we called them respectively "instructed Musicians" and "uninstructed Not Musicians". At the end, we obtained the following four groups of subjects and denominations:

instructed Non Musicians (iNM)	5 subjects
instructed Musicians (iM)	6 subjects
uninstructed Non Musicians (uNM)	6 subjects
uninstructed Musicians (uM)	5 subjects

Table 1. Subject distribution among the 4 groups.

3.2 Materials

For our two tests we choose a very popular, old style song, written in 1966 and interpreted by the Italian singer Adriano Celentano. The song (*Il ragazzo della via Gluck*) is very clear and easy in the melodic structure. The harmonic rhythm is not too fast and allows easy body movements in the physical space. The first musical phrase of the song is

composed of 11 segments and includes 10 chord changes and a total number of 3 different employed chords, two primary and one parallel (I, V and VI degree), all belonging to the same key.

3.3 Scenario

The *Harmonic Walk*'s test, both the first and second, is subdivided into three phases. In the first phase the user is presented to the straight path along the borders of the mapped area (see Fig. 2). Every step through the mapped regions produces the sound of the portion of the audio file corresponding to one melodic unit. If the user fails to step to the next zone in time, the audio file ends and the performance is interrupted. This feedback shows to the user the right harmonic rhythm, which represents a fundamental knowledge for melody harmonization, and makes her/him aware of the exact place where harmonic changes occur. In the second phase of the experience the user enters the circular ring containing the six roots of the tonality harmonic space, with the aim to explore it and to practice the right positions for a precise performance. When ready, s/he begins the third final phase: the harmonization task. Here the user can try to find the chords useful for the melody accompaniment of the composition, remembering the points of the song where the harmonic changes occurred and searching for the right chord to match. To achieve such a successful result implies a certain number of abilities like remembering the melody of the tonal composition, locating the chords in the circular mask and being able to reach the right place at the exact time requested by the melody.

3.4 Method

The high school students were always tested individually, in private sessions where only the test conductor and the music teacher were present.

3.4.1 First test (common to all the 22 subjects)

1. no previous information about the employed song is provided to the subjects
2. each student is presented with some written instructions about the tasks s/he had to accomplish, while a short demonstration about the environment and its interaction modalities is provided by the test conductor
3. the student undergoes the three phases of the test, lasting respectively a maximum of 5, 3 and again 5 minutes
4. after the time is expired or the last task is accomplished, the student fulfills a questionnaire for both quantitative and qualitative feedback about the test.

3.4.2 Second test (different for instructed and uninstructed subjects)

1. the 11 chosen subjects take part to a 1 hour demonstrative lesson where the test conductor and the teacher explain the aim of the test and show how the applications works. The conductor shows the meaning of the three phases of the test and how the required tasks could be accomplished (only for instructed subjects)
2. each student undergoes the three phases of the test, lasting respectively a maximum of 5, 3 and 5 minutes
3. after the time is expired or the last task is accomplished, the student fulfills a new questionnaire identical to that of the first test.

Instructed subjects follow the second test schedule from the beginning, while uninstructed subjects skip point n. 1.

4. RESULTS AND DISCUSSION

4.1 Quantitative Assessment

Also if the questionnaire fulfilled by the test subjects is very rich in both quantitative and qualitative information about the experience with the *Harmonic Walk* application, for the analysis of control and experimental data we concentrate only on quantitative assessment results. In particular, beyond the evaluation of the success with respect to the main application's task, the melody harmonization, we measure the knowledge acquired in the following three fields of the application's musical content:

1. the detection of the syllables of harmonic changes. The identification of the right syllables indicates that the subjects felt the right points where the harmonic changes occurred and that they are able to remember them. We postulate that this knowledge originates directly from the enactive experience acquired in the first phase of the test.
2. the number of harmonic changes involved in the harmonization task. To provide the correct answer, the subjects need to detect the harmonic changes and to count exactly how many they are. Actually, the number of the harmonic changes is the same of the syllables of change; but realizing this implies further memorization, musical reasoning and awareness from the subject.
3. the total number of chords (tonal functions) employed in the harmonization task. This is the most difficult answer, because it requires that subjects identify the harmonic functions and recognize them when they return in the song's first phrase. This ability is closely related to the enactive experience of the second and third test phases, where the subjects had to find the right chordal route in the circular ring of the song's tonality harmonic space.

4.2 Data Gathering

For the harmonization task data, we rely on the test conductor's reports, where a full success (FS) is recorded when the subject can produce in the assigned time a clearly

defined version of the melody harmonization that s/he considers valid. When the subject succeeds in harmonizing only the first part of the melody, where only tonic and dominant chords are employed, a partial success is reported (PS). If the subject, also after many trials, cannot provide an ultimate harmonized version, a failure is reported (NS). For the detection of the syllables of harmonic change, we provided the subjects with a grid where the syllables of the first song's phrase were reported. We asked them to cross those syllables corresponding to the harmonic changes, remembering the exact point where they stepped in through the melodic unit's blocks during the test's first phase. Only the right checked syllables are computed in the means showed in Table 3, while for the other two musical elements knowledge, we source our data directly from explicit subject's answers.

4.3 Meta-Analysis Methods

We consider the 1st test results as the control and the 2nd test results as the treatment. [6] After collecting the two test's records about the full success (FS), partial success (PS) and unsuccessful (NS) subjects in the harmonization task, we obtain a 2x3 contingency table (see Table 2) for each of the 4 subject categories. For each table the Fisher's exact probability [7] (P-value) is calculated, in order to express the degree of statistically significant association between the control and treatment results, assuming a P-value < 0.05 as the significance threshold.

Effect size [8] and Cohen's *d* are calculated on the basis of mean and standard deviation for the syllables of harmonic changes, number of harmonic changes and of employed chords in the harmonization task (see respectively Tables 3, 4 and 5).

4.4 Data Meta-Analysis

In this Section we analyze our data and organize them in 4 Subsections and Tables, depending on the musical content. For each assessed ability, we provide data interpretation based on the appropriate meta-analysis methods.

4.4.1 Harmonization task

The harmonization task contingency tables (Table 2), show statistically significant improvement results only in the category of instructed subjects (Musicians and Non-Musicians), with a very good result for the category of instructed Musicians (iM's P-value = 0.007), while uninstructed Musicians and Non Musicians are well beyond the significance threshold (respectively, P-value = 1 and 0.437).

Harmonization task

	1st TEST			2nd TEST			P-value
	FS	PS	NS	FS	PS	NS	
iNM	0	0	5	4	0	1	0.047
iM	1	2	3	6	0	0	0.007
uNM	1	0	5	0	1	5	*1.000*
uM	0	2	3	2	2	1	*0.437*

Table 2. Contingency tables of 1st and 2nd test results for the harmonization task for each of the 4 subject groups. FS is the number of fully successful subjects, PS is the number of partially successful and NS is the number of non successful subjects.

4.4.2 Syllables of change

Table 3 shows more uniform results among the various subject categories for the detection of the syllables of change, as all Cohen's *d*s are comprised in the same range of values, with a medium improvement for all the 4 subject categories.

Syllables of change

	1st TEST		2nd TEST		Effect size	Cohen's *d*
	Mean	SD	Mean	SD		
iNM	4.25	2.872	5.2	3.114	0.156	0.317
iM	5	2	6	3.464	0.174	0.353
uNM	3	2.944	3.75	1.258	0.163	0.331
uM	6	3.082	7	1.825	0.193	0.394

Table 3. Table of mean, standard deviation, effect size and Cohen's *d* of the results of the 1st and 2nd test for the number of correct syllables of change detected by the 4 subject categories.

4.4.3 Number of harmonic changes

The results for the detection of the right number of harmonic changes show a large improved level for the category of instructed subjects, while for uninstructed subjects we have a small improvement (*d* = 0.176), if not a negative effect in the category of uninstructed Non Musicians.

No. of Harmonic Changes

	1st TEST		2nd TEST		Effect size	Cohen's *d*
	Mean	SD	Mean	SD		
iNM	5.5	2.121	9.2	1.095	0.738	2.192
iM	7	2.549	10	0.408	0.634	1.643
uNM	4.3	1.528	4	0.816	*-0.121*	*-0.244*
uM	8.25	2.872	8.8	3.347	0.087	0.176

Table 4. Table of mean, standard deviation, effect size and Cohen's *d* of the results of the 1st and 2nd test for the number of correct harmonic changes detected by the 4 subject categories.

[6] Usually, in learning environment analysis, the control is given by the results obtained during a traditional lesson, without the help of any technology, while the treatment is given by the results obtained during sessions where the application is used to convey the same contents. In the case of the *Harmonic Walk* this was an impractical assessment condition, at least for the Non Musicians category of subjects. Indeed the melody harmonization is a rather difficult task, which requires a great amount of practice and musical knowledge, well beyond the level also of a music high school student.

[7] The Fisher's test is used in the analysis of contingency tables when sample sizes are small. Its results are always exact also if the frequency of values is less than 5 (the frequency validity limit for Chi-test).

[8] Effect size measures the magnitude of a treatment effect. By convention, it is a positive value if it is in the direction of improvement, otherwise it is negative.

4.4.4 The number of employed chords (tonal functions).

The results for the number of employed chords show a good improvement in the categories of Non Musicians (both instructed and uninstructed), while showing a negative value for instructed Musicians and a very small effect size for uninstructed Musicians (d = 0.054).

No. of Employed Chords (tonal functions)

	1 st TEST		2 nd TEST		Effect	Cohen's
	Mean	SD	Mean	SD	size	*d*
iNM	3.67	1.527	3	0	0.296	0.620
iM	3.4	1.673	3.75	0.957	*-0.127*	*-0.258*
uNM	5.4	3.714	3	0	0.415	0.913
uM	3.3	1.204	3.25	0.5	0.027	0.054

Table 5. Table of mean, standard deviation, effect size and Cohen's d of the results of the 1st and 2nd test for the number of employed chords (tonal functions) detected by the 4 subject categories.

4.5 Discussion

4.5.1 Test data evaluation

We submitted our subjects to various assessment conditions to discover what is the weight of the simple test repetition (for uninstructed Musicians and Non Musicians) and of the test repetition after a 1 hour lesson, where the test's musical content was explained and showed by the test conductor (instructed Musicians and Not Musicians). Our results show that, for the harmonization task, the lesson was very important, as instructed subjects (Musicians and Not Musicians) could achieve a very good result in the second test's session, while uninstructed subjects performed rather poorly. Anyway, the lesson didn't provide any technical detail about the harmonization task, but, rather, showed to the subjects how it could be practically done in the application's environment. Thus, the necessary information was transmitted to the subjects only by the observation of the test conductor's movements and interaction style and not by theoretical explanations, thereby proving the high power of knowledge communication of the *Harmonic Walk* learning environment. But the results of the harmonization task do not always coincide with the findings in the three musical content knowledge acquisition tests. For instance, in the detection of the syllables of change we observe a general uniform improvement among the 4 subject categories, ranging from a minimum of Cohen's d value of 0.317 for instructed Non Musicians, to a maximum of 0.394 for uninstructed Musicians. In any case, it is a medium difference, but it indicates a clear improvement in the musical knowledge, independently from the success in the harmonization task. The explanation of such a result can simply be the memorization of the song's words, which improved with a second trial. In any case, the information was explicitly provided to the instructed group during the lesson, but this didn't seem to affect significantly the test's results. The number of harmonic changes is again much better detected by the instructed group, with great Cohen's d values. This information, explicitly provided during the lesson, was clearly much easier to remember than the list of the syllables of the harmonic changes and, consequently, most of the instructed subjects could produce a better answer in the second test. In the perception of the number of the tonal functions employed in the song's first phrase harmonization, we record a right answer in all the non musician subjects. This is probably due to suggestions shared among the group components, and, consequently, this part of the test cannot be considered valid.

4.5.2 General results evaluation

We have now some element to answer the questions we pose at the beginning of this paper.

1. *It is possible to make implicit knowledge about tonal melody and harmony to emerge and to become a real ability?*
   If we consider the results obtained in the assessment of the only two available fields of musical knowledge (detection of syllables of change and number of harmonic changes), we see a general middle level improvement for the syllables of change and a limited to the instructed category low improvement for the number of harmonic changes. So, inside the boundaries of this experiment, the answer is yes, it is possible to make implicit knowledge to emerge, especially when the application's practice is matched with supporting explanatory lessons.

2. *How important is the role of explicit, previously delivered information?*
   When the information is easy to remember and has been given explicitly, as in the case one the number of harmonic changes, it actually affects the results, as shown in Table 4. Such kind of answers are also easy to be suggested, as in the case of the detection of harmonic functions (see Subsubsection 4.4.4). As a matter of fact these two answers are meaningful as a product of a reasoning, but are not meaningful as a mere repetition. This seems to ask for a better control in the communication during the lesson and for better refinements in the assessment method.

3. *Is enactive experience stronger than explicit information?*
   The enactive experience has been crucial in the performances registered for the harmonization task and for the detection of the syllables of change, as it has been able to transmit only by imitation and in a very short time a lot of information to the test subjects. Thus the answer is yes: Enactive experience is stronger than explicit information, at least in the case of the musical task we choose for our test.

4. *And if so, in what domain?*
   The authors of [13] tried to answer this question referring to a learning experience in the field of physics. They found that embodied physical learning leads to the same grade of improvement as a mouse-operated learning experience. Also, if we didn't yet perform a similar test for the *Harmonic*

Walk environment, we can say that music education has a lot to do with embodied knowledge also without the use of technologies. In our design policy, we included a lot of abstract concepts, but these were always acted through acquired or implicit embodied knowledge. Thus, we were not surprised by our test's results, and we consider the difficulty of translating the same communicative power in other fields of knowledge different from music, a challenge that requires more control and refined research methods.

5. CONCLUSION AND FURTHER DEVELOPMENTS

We submitted a group of musicians and non musicians high school student to a two phase assessment test under different conditions, with the aim to measure the educational power of the *Harmonic Walk* application. We found that a lot of information could be conveyed by simple imitation, skipping other forms of explicit knowledge transmission, like traditional class sessions. Moreover, explicit information content could improve the student's performance at some extent, but we could not prove if this major results were due to mere information memorization/suggestion or to actual knowledge and content awareness. The big challenge of test method refinement has to be faced in next assessment sessions, to try to better control the test conditions. Moreover, we missed a post-test assessment session, where we could outline the concepts acquired by our subjects and cause further reasoning and awareness.

Acknowledgments

We want to thank Leonardo Amico for his support in programming the *Zone Tracker* application, Cesare Contarini, Dean of the Catholic Institute "Barbarigo" of Padova and, particularly, Giuseppe Viaro, Music Technology Teacher for their cooperation during the assessment tests.

6. REFERENCES

[1] R. Jackendoff, *Consciousness and the computational mind. Explorations in cognitive science, No. 3. Ca*. Cambridge, MA, US.: The MIT Press., 1987.

[2] K. A. Corrigall and L. J. Trainor, "Musical enculturation in preschool children: Acquisition of key and harmonic knowledge," *Music Perception*, vol. 28, no. 2, pp. 195–200, 2010.

[3] M. Mandanici, A. Rodà, and S. Canazza, "The harmonic walk: an interactive educational environment to discover musical chords," *Proceedings of ICMC-SMC Conference 2014, Athens*, 2014.

[4] M. Mandanici, L. Amico, A. Rodà, and S. Canazza, "Conoscere l'armonia tonale nell'ambiente interattivo "harmonic walk"," *XX Colloquio di Informatica Musicale, Roma*, Ottobre 2014.

[5] M. Mandanici, A. Rodà, and S. Canazza, "The harmonic walk: an interactive physical environment to learn tonal melody accompaniment." July 2015, submitted.

[6] ——, "A conceptual framework for motion based music applications," *2nd Workshop on Sonic Interactions in Virtual Environments, Arles*, March 2015.

[7] S. Holland, *Learning about harmony with Harmony Space: an overview*. Springer, 1994.

[8] S. Holland, P. Marshall, J. Bird, and al., "Running up Blueberry Hill: Prototyping whole body interaction in harmony space." *Proceedings of the 3rd international Conference on Tangible and Embedded interaction*, 2009.

[9] T. Bergstrom, K. Karahalios, and J. C. Hart, "Isochords: visualizing structure in music," in *Proceedings of Graphics Interface 2007*. ACM, 2007, pp. 297–304.

[10] mDecks Music, "Mapping tonal harmony," November 2012. [Online]. Available: http://mdecks.com/mapharmony.html

[11] L. Bigo, J. Garcia, A. Spicher, W. E. Mackay *et al.*, "Papertonnetz: Music composition with interactive paper," in *Sound and Music Computing*, 2012.

[12] B. Manaris, D. Johnson, and Y. Vassilandonakis, "Harmonic navigator: A gesture-driven, corpus-based approach to music analysis, composition, and performance," in *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

[13] M. C. Johnson-Glenberg, D. Birchfield, P. Savvides, and C. Megowan-Romanowicz, "Semi-virtual embodied learning-real world stem assessment," in *Serious Educational Game Assessment*. Springer, 2011, pp. 241–257.

[14] M. Santos, A. Chen, T. Taketomi, G. Yamamoto, J. Miyazaki, and H. Kato, "Augmented reality learning experiences: Survey of prototype design and evaluation," *Learning Technologies, IEEE Transactions on*, vol. 7, no. 1, pp. 38–56, Jan 2014.

[15] M. Wright, "Open sound control-a new protocol for communicationg with sound synthesizers," in *Proceedings of the 1997 International Computer Music Conference*, 1997, pp. 101–104.

Distributing Music Scores to Mobile Platforms and to the Internet using INScore

D. Fober **G. Gouilloux** **Y. Orlarey** **S. Letz**
GRAME
Centre national de création musicale
Lyon - Fr
fober@grame.fr

ABSTRACT

Music notation is facing new musical forms such as electronic and/or interactive music, live coding, hybridizations with dance, design, multimedia. It is also facing the migration of musical instruments to gestural and mobile platforms, which poses the question of new scores usages on devices that mostly lack the necessary graphic space to display the music in a traditional setting and approach. Music scores distributed and shared on the Internet start also to be the support of innovative musical practices, which raises other issues, notably regarding dynamic and collaborative music scores. This paper introduces some of the perspectives opened by the migration of music scores to mobile platforms and to the Internet and it presents the approach adopted with INScore, an environment for the design of augmented, interactive music scores.

1. INTRODUCTION

When you search *"music score app"* on the Internet, you'll likely get more than 39,000,000 matching pages when associated to *"android"*, more than 12,000,000 when associated to *"iOS"* and over 29,000,000 with *"iPad"* as keyword. Adding *"Web"* or *"Internet"* to the query results in an explosion of matching pages, while support for historical operating systems tends to be bygone (figure 1). These figures indicate clearly a significant evolution and a clear migration of the support for musical scores to mobile devices but above all, to the Internet.

From a technical viewpoint, this change represents a move from one operating system [OS] to another. Web browsers may be viewed as a kind of OS on top of an abstract machine: they are integrating step by step all the services of an OS, up to audio services with the recent Web Audio API [1]. But actually the change is far more than this simple move:

- mobile platforms have adopted a fundamentaly different approach from user point of view: no traditionnal input device (keyboard, mouse), embedded sensors that may be used as controlers, a reduced graphic space especially for smartphones, a step back on multitask aspects but an wave of services composition and integration.
- web applications differ also due to their natural way to agglutinate distributed resources and to share content between several users. Theoretically, they can be deployed on all the platform previously mentioned.

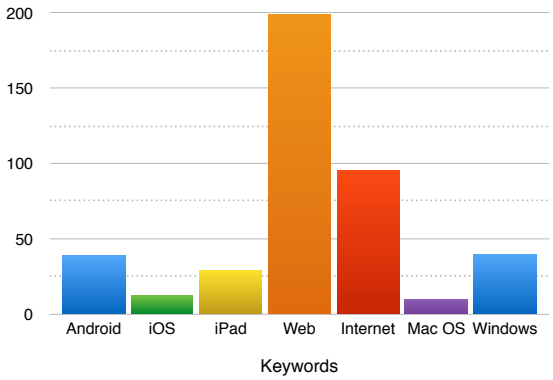


Figure 1. Results of a search using Google with *"music score app"* associated to different keywords. The numbers are in million hits.

The approach adopted by almost all music notation applications available on mobile platforms is rather classical: you can find a plethora of music score readers and players, based on the common music notation, more or less sophisticated. Music score edition is also supported by these applications but they have to re-think the user interface due to the lack of input device: handwritten recognition is one of the explored solutions [1] [2]. On smartphones, the screen size limitation is not really handled, apart with messages that inform the user that the application may not be fully functional. The more innovative approaches generally come from artistic uses [2].

On the web side, you can find online music notation editors, online score sharing systems [3], or JIT compilation services like those proposed for years by the GUIDO Engine [3]. More recently, music notation services have been made available to developers and users under the form of

Copyright: ©2015 D. Fober 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] NotateMe http://www.neuratron.com/notateme.html
[2] StaffPad http://www.staffpad.net/
[3] MuseScore https://musescore.org/en/handbook/share-scores-online-0

a RESTFUL web service [4]. Solutions for score layout and redering can also be embedded in a web page: this is the case for the GUIDO Engine [5] that is now available as a Javascript library. Applications for music practising are now moving to the Internet (e.g. Weezic [4]) but based on already existing strategies [6] [7]. Distributed and collaborative scores are appearing like the Flat music score editor [5] and also tools for network improvisation [8].

Whether running on the web or on mobile platforms, the approach to music notation adopted by almost applications looks quite classical. Innovation generally comes from artistic approaches and are based on specific tools. However, the context of mobile platforms or of the Internet could lead to new and original uses, and we think that in this regard, an adequate support is missing from tools for music notation.

This paper proposes several use cases that are specific to the context described above. These use cases may be implemented using INScore, an environment for the design of augmented interactive music scores that has been extended to support distributed scores and collaborative design, and that runs on all the major platforms (MacOS, Linux, Windows, Android and iOS). The paper starts with a brief reminder of the INScore environment. Next it presents the network extensions. Various use cases are then considered and a concrete realization of a dynamic score published over Internet is presented.

2. INSCORE

INScore is an environment for the design of augmented, interactive music score [9] that is entirely controled by an Open Sound Control [OSC] API [9]. It supports arbitrary graphic resources (symbolic music notation, text, images, vectorial graphics...) and displays the time relationships of the score components by the way of a simple synchronization mechanism. INScore supports performance representation, viewed as audio or gestural signal, as well as interaction process representation also viewed as signals [10]. It includes an event based interaction mechanism [11] that provides a simple and homogeneous way to describe interactions in the graphic or the temporal space.

INScore input language is a textual version of OSC messages extended to support variables, extended OSC adresses and Javascript sections. A Javascript engine is embedded in each score and may be remotely triggered via OSC messages.

The script below shows the example of a rectangle synchronized on a symbolic score (described using the Guido Music Notation format [GMN] [12]) i.e. its graphic position is computed frm it's time location. The result is illustrated in figure 2.

[4] `http://www.weezic.com/`
[5] `https://flat.io/`

EXAMPLE 1:

```
/ITL/scene/score set gmn '[g e c a f]';
/ITL/scene/rect set rect 0.05 0.3;
/ITL/scene/rect color 0 0 240 120;
/ITL/scene/sync rect score;
/ITL/scene/rect date 3 4;
```

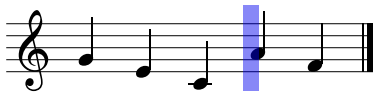


Figure 2. A rectangle synchronized on a score.

INScore has been used in many artistic projects, the last one being an implementation of Earl Brown's December Variation by Richard Hoadley [13].

3. INSCORE WEB SUPPORT

INScore has been extended to support aggregation of distributed resources over Internet, as well as publication of a score via the HTTP and the WebSocket protocols.

3.1 Distributed score components

Most of the components of a score can be specified in a litteral way or using a file. The example below will produce the same object, provided that the '`score.gmn`' file contains the `[g e c a f]` code.

EXAMPLE 2:

```
/ITL/scene/score set gmn '[g e c a f]';
  is similar to
/ITL/scene/score set gmnf 'score.gmn';
```

All the file based resources can be specified as a simple file path using absolute or relative path, or as an HTTP url. When using the relative path form, a file absolute path is built using the score current path, that may be set to arbitrary location using the `rootPath` message. This current path can be also be set to an arbitrary HTTP url, so that further use of a relative path will also result in an url.

The example below refers the same '`score.gmn`' file on `host.domain.org`.

EXAMPLE 3:

```
/ITL/scene/score set gmnf
    'http:://host.domain.org/score.gmn';
  is equivalent to
/ITL/scene rootPath
    'http:://host.domain.org/';
/ITL/scene/score set gmnf 'score.gmn';
```

This mechanism allows to mix local and remote resources in the same music score, but also to express local and remote scores in a similar way, using just a `rootPath` change.

3.2 HTTP and WebSocket components

A music score can be published on the Internet using the HTTP or the WebSocket protocols. Specific components can be embedded in a music score in order to make this score available to remote clients:

- an HTTP server, which INScore type is `httpd` and that takes a listening port number as argument,
- a WebSocket server, which type is `websocket` and that takes a listening port number and a maximum rate for clients notification as arguments.

The WebSocket server allows bi-directional communication between the server and the client. It sends notifications of score changes each time the graphic appearance of the score is modified, provided that the notification rate is lower than the maximum rate set at server creation.

The example below creates an HTTP server that responds on the port 8000 and a WebSocket server that responds on the port 8100 and sends notifications at a maximum rate of 200 ms.

EXAMPLE 4:
```
/ITL/scene/http set httpd 8000;
/ITL/scene/ws set websocket 8100 200;
```

The communication scheme between a client and an INScore server relies on a reduced set of messages. These messages are protocol independent and can be equally supported over HTTP or WebSocket. Table 1 gives an overview of the client server communication scheme:

- the `get` message requests an image of the score. It is similar to an `export` message adressed to INScore, which result is sent over HTTP or WebSocket.
- the `version` message requests the current version of the score. The server answers with an integer value that is increased each time the score is modified. This message is intended to allow clients to keep an up-to-date image of the score. Note that the WebSocket server automatically sends changes notifications with versionning information.
- the `post` message is intended to send an INScore script to the server. The server answers with a status message which is between `OK` or `ERROR`. In case of error, details on the failure reason are provided. In case of success, the score may be modified and its current version number is increased.
- the `click` message is intended to allow remote mouse interaction with the score. The associated data should be a position in an image previously retrieved with a `get` message.

3.3 Messages forwarding

Message forwarding is another mechanism provided to distribute scores over a network. It is applied at application and score levels. It consists in a list of destination hosts specified using a host name or an IP number, and suffixed with a port number. All the OSC messages may be forwarded to the indicated hosts on the corresponding port number, provided they are not filtered out (figure 3). The filtering strategy is based on OSC adresses and/or on INScore methods (i.e. messages adressing specific objects attributes).

Request	Data	Answer	Side effect
`get`	*none*	an image	*none*
`version`	*none*	version num	*none*
`post`	INScore script	status	new score
`click`	x, y position	*none*	new score

Table 1. INScore server Web API.

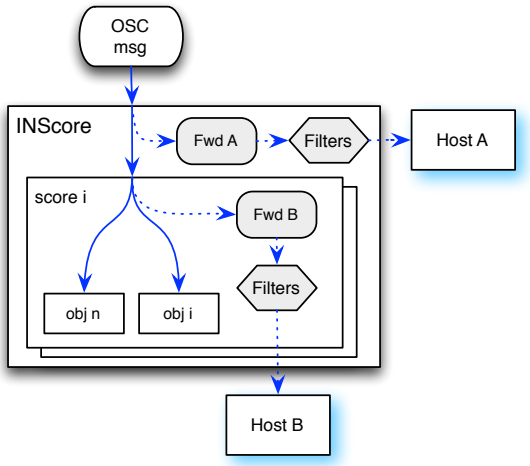


Figure 3. Message forwarding mechanism.

The next example installs a forwarding mechanism at application level: all the incoming messages may be forwarded to a host specified by IP number on the UDP port 7000. Next the filter is configured so that `clock` and `date` messages will not be forwarded.

EXAMPLE 5:
```
/ITL forward 192.168.1.27:7000;
/ITL/filter reject clock date;
```

4. USE CASES

4.1 Groupware technologies

Groupware technologies as described in [7] may be easily deployed using INScore and the forwarding mechanism. Let's say that we have a teacher score on a station T and students on 3 stations $S1$, $S2$, $S3$.

The settings illustrated in figure 4 can be implemented with the following messages:

A) on T : `/ITL forward S1 S2 S3;`

B) on T : `/ITL forward S1;`
on S1 : `/ITL forward T;`

C) on T : `/ITL forward S1 S2 S3;`
on S1 : `/ITL forward T;`

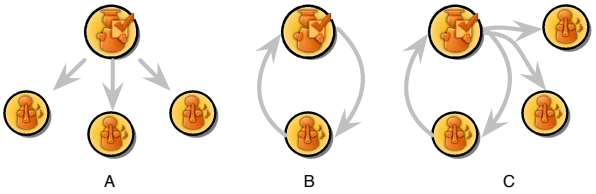


Figure 4. Use cases in a pedagogic setting: A)the teacher score is published to the students, B) the teacher and the student interact with the same score, C) the setting is similar to B) but the score is published to the other students that can look at the interaction.

4.2 Collaborative score design

Collaborative score design could be implemented with any number of participants, i.e. all the participants can interact with a score that is available to all the others, also in read/write mode. We assume that one station is the central point of messages distribution, then the forwarding scheme illustrated in figure 5 is describe below:

EXAMPLE 6:
```
on A: /ITL forward B C;
on B: /ITL forward A;
on C: /ITL forward A;
```

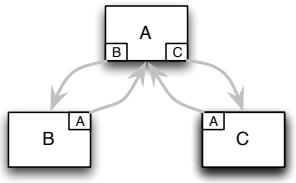


Figure 5. Collaborative score design.

Note that the forwarding scheme could be setup from the same computer using INScore extended OSC addresses (e.g. `B:/ITL/forward A`)

Note also that the forwarding mechanism prevents messages to be forwarded to the sender and thus, avoids direct loops (but not indirect loops e.g. A → B → C → A).

4.3 Shared score over Internet

Although the forwarding scheme above is basically intended to run on a local network, it could be implemented over the Internet as well, but since the underlying communication protocol is UDP, it may face significant packets losses, depending on the network conditions.

A secure solution to collaborative design may use the HTTP or WebSocket servers. The example below implements a score that displays the local score and includes a remote score as illustrated in figure 6.

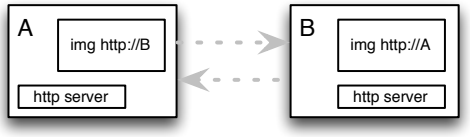


Figure 6. Scores shared over the Internet.

EXAMPLE 7:
```
On each station:
    /ITL/scene/http set httpd 8000;
    /ITL/scene/remote set img
              'http://remote.address';
```

Note that using a `websocket` object instead of `httpd` could make the remote view refreshment transparent. To do so, the `ws://` protocol has to be implemented for file based resources. That's one of the future directions.

4.4 Audience score based interaction

The INScore internet protocols support UI interactions and notably, relay the user clicks or touch screen interactions to the server. It is thus easy to imagine a concert setting where the music score is published (e.g. using the `websocket` server) and where the audience could get the score on a mobile phone and interact with it in real-time using the event based interaction mechanism of INScore, modifying the course of the music piece.

4.5 Flux Aeterna

Flux Æterna has been composed by Vincent Carinola in 2014. The piece has been designed for the Internet[6]. It comes under the form of an endless audio stream. The listening conditions are similar to those of a web radio but here, the listener can influence the future of the piece by providing its own sound files.

A dynamic score of the piece has been designed using INScore. The piece is using Max/MSP that sends modules and events information to INScore in real-time via OSC. This information is converted into a graphic information that reflects the piece structure (figure 7) using the embedded Javascript engine.

The score has been initially designed for a local display, in the context of an exhibition. Adding a simple `httpd` or `websocket` object to the score allows to make it public over Internet, as illustrated with the HTTP example below:

EXAMPLE 8:
```
/ITL/scene/server set httpd 8000;
  makes the score available at
http://thehost.thedomain.org:8000
```

In addition, the score may be distributed in real-time to any INScore viewer connected to the local network using the forwarding mechanism mentionned in section 3.3. The script below forwards the messages to any INScoreViewer

[6] `http://vr.carinola.free.fr/fluxaeterna/`

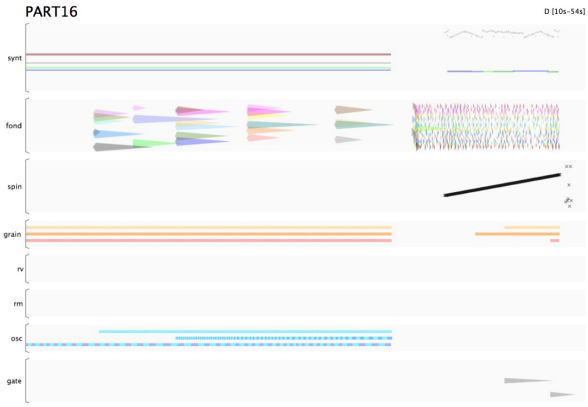


Figure 7. One page of Flux Æterna.

running on a device connected to the local network. Messages addressed to the Javascript engine are filtered in order to only forward the result of their evaluation.

EXAMPLE 9:

```
/ITL forward 192.168.1.255;
/ITL/filter reject '/ITL/scene/javascript';
```

5. CONCLUSIONS

Applications for music notation are moving to mobile platforms and to the web, following the general stream of computing migration. Most of these applications are reproducing the existing approaches to music notation although their deployment on the web and/or mobile platforms could take advantage of the technological context to create innovative uses. Actually, innovation exists but it is restricted to specific applications, mostly designed in artistic projects. With its network extensions and its support for Android and iOS, INScore provides a set of solutions for distributed score design and interaction. The approach tends to make network support as transparent as possible in a score description. Future extensions should make remote resources available using the `WebSocket` protocol, which should make remote files refreshment transparent and allow additional use cases in the domain of shared and collaborative score design.

Acknowledgments

INScore research and development has been funded by the French National Research Agency [ANR].
INScore is an open source project hosted on SourceForge (`http://inscore.sf.net`)

6. REFERENCES

[1] *Web Audio API*, W3C, April 2015. [Online]. Available: http://webaudio.github.io/web-audio-api/

[2] P. Timothy and M. Brad, "Engravings for prepared snare drum, ipad, and computer," in *Proceedings of the Conference on New Interfaces for Musical Expression*, ser. NIME'14, 2014, pp. 82–83.

[3] R. K. and H. Hoos, "A Web-based Approach to Music Notation Using GUIDO," in *Proceedings of the International Computer Music Conference*. ICMA, 1998, pp. 455–458.

[4] M. Solomon, D. Fober, Y. Orlarey, and S. Letz, "Providing music notation services over internet," in *Proceedings of the Linux Audio Conference*, Karlsruhe, Allemagne, 2014, pp. 91–96. [Online]. Available: solomon14a.pdf

[5] C. Daudin, D. Fober, S. Letz, and Y. Orlarey, "The guido engine – a toolbox for music scores rendering," in *Proceedings of Linux Audio Conference 2009*, LAC, Ed., 2009, pp. 105–111. [Online]. Available: lac2009.pdf

[6] S. Raptis, A. Chalamandaris, A. Baxevanis, A. Askenfeld, E. Schoonderwaldt, K. F. Hansen, D. Fober, S. Letz, and Y. Orlarey, "Imutus - an effective practicing environment for music tuition." in *Proceedings of the International Computer Music Conference*. ICMA, 2005, pp. 383–386. [Online]. Available: Barcelona

[7] D. Fober, S. Letz, and Y. Orlarey, "Vemus - feedback and groupware technologies for music instrument learning," in *Proceedings of the 4th Sound and Music Computing Conference SMC'07 - Lefkada, Greece*, 2007, pp. 117–123.

[8] R. Canning, "Interactive parallax scrolling score interface for composed networked improvisation," in *Proceedings of the Conference on New Interfaces for Musical Expression*, ser. NIME'14, 2014, pp. 144–146.

[9] D. Fober, Y. Orlarey, and S. Letz, "Inscore – an environment for the design of live music scores," in *Proceedings of the Linux Audio Conference – LAC 2012*, 2012, pp. 47–54.

[10] ——, "Representation of musical computer processes," in *Proceedings of International Computer Music Conference*, 2014.

[11] D. Fober, S. Letz, Y. Orlarey, and F. Bevilacqua, "Programming interactive music scores with inscore," in *Proceedings of the Sound and Music Computing conference – SMC'13*, 2013, pp. 185–190. [Online]. Available: fober-smc2013-final.pdf

[12] H. Hoos, K. Hamel, K. Renz, and J. Kilian, "The GUIDO Music Notation Format - a Novel Approach for Adequately Representing Score-level Music." in *Proceedings of the International Computer Music Conference*. ICMA, 1998, pp. 451–454.

[13] R. Hoadley, "December variation (on a theme by earle brown)," in *Proceedings of the ICMC/SMC 2014*, 2014, pp. 115–120.

SOUND MY VISION: REAL-TIME VIDEO ANALYSIS ON MOBILE PLATFORMS FOR CONTROLLING MULTIMEDIA PERFORMANCES

Miranda Kreković
School of Computer and Communication Sciences (EPFL), Lausanne, Switzerland
miranda.krekovic@epfl.ch

Franco Grbac
Independent researcher
franco.grbac@gmail.com

Gordan Kreković
Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia
gordan.krekovic@fer.hr

ABSTRACT

This paper presents Sound My Vision, an Android application for controlling music expression and multimedia projects. Unlike other similar applications which collect data only from sensors and input devices, Sound My Vision also analyses input video in real time and extracts low-level video features. Such a versatile controller can be used in various scenarios from entertainment and experimentation to live music performances, installations and multimedia projects. The application can replace complex setups that are usually required for capturing and analyzing a video signal in live performances. Additionally, mobility of smartphones allows perspective changes in sense that the performer can become either an object or a subject involved in controlling the expression. The most important contributions of this paper are selection of general and low-level video feature and the technical solution for seamless real-time video extraction on the Android platform.

INTRODUCTION

In the context of new interfaces for musical expression, mobile devices such as smartphones and tablets take an important place. They are multipurpose and omnipresent devices powerful enough for real-time digital signal processing. What makes them different from pocket computers are hardware prerequisites like touch screens, microphones, cameras, and various sensors which can serve as a foundation for building musical interfaces. For those reasons, mobile devices have become an interesting platform for computer music research [1-3] and for developing practical applications [4, 5].

The research conducted by Kell Thor and Marcelo M. Wanderley at the beginning of 2014 showed that there were more than 5000 iOS applications for making music available on the official app store [2]. The variety and versatility of those applications ensure their usage in different scenarios – from entertainment and experimentation to music production and live performances. While some applications are capable of producing sounds by themselves, others are designed to serve as controllers, so they only send parameter data to other devices.

Musical expression can be observed within several dimensions such as pitch range, pitch style and tuning, dynamic range, timbre style and process, articulation, ornamentation, number of parts, and spatial dimension. [6]. In order to achieve expressibility in those dimensions, most mobile applications primarily rely on inputs from the touch screen. Using the graphical representations, a variety of metaphors for controlling musical expression can be employed. The metaphors such as keyboards, dials, strings, pads, and sliders are intuitive to use because they inherently inform users how to interact with the application and what sonic results of their actions they may expect [7].

In addition to a touch screen, mobile devices are usually equipped with various peripherals and sensors including gyroscope, accelerometers, Global Positioning System (GPS) module, microphone, and cameras. Most of the mentioned input devices and sensors have been used in mobile applications for making music and controlling musical performances. However, this is not the case with cameras.

We assume that the first reason for not using the input image for controlling musical expression is the lack of the systematic research of video features which can be generally used for such purposes. The second reason is that just until a few years ago mobile devices were not capable of complex video analysis in real time.

Video analysis represents a challenging part of many interactive motion sensing systems for installations, enhanced dance choreographies, and other multimedia projects [8]. Real-time video feature extraction on mobile devices can make technology more accessible and convenient for musicians and performance artists. Using a smartphone or tablet, they can easily control an interactive system by dance movements or any other visual information. For such a purpose musicians and performing artists would otherwise need a complex hardware setup with cameras, computers, and lots of cables. Therefore, using widely present and relatively affordable devices they could be higher motivated to explore multimodalities of interaction between physical movements, musical expression, and multimedia.

Another important benefit of using live video input from mobile devices is related to the philosophical perspective. While in usual setups cameras are either fixed in one place or have limited trajectories, mobile devices can move in space without limitations and this allows artists to change a perspective. Instead of being an object observed by the camera, a performer can sometimes become a subject who carries a mobile device to capture the audi-

Copyright: © 2015 Miranda Kreković et al. 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.

ence or a part of scenography which is not visible from other angles. Such a changed perspective is an interesting philosophical foundation of exploring interactivity between the performer and a multimedia system.

Use of mobile applications could open even more interesting opportunities. For instance, by applying the principles of crowdsourcing, the audience can capture the scene by their smartphones and thereby contribute to the interaction with the multimedia system. Their multiple perspectives can be combined when using mobile devices from different places in the audience.

Since there is a lot of practical benefits and new opportunities for performance artists, the goal of our research was to design and develop a mobile application for controlling interactive and multimedia system which also supports real-time video analysis. As the result, we created Sound My Vision, an Android application which captures data from sensors and the touch screen, but additionally extracts video features form the camera input. The mobile application sends collected data over the wireless network so that the other device with a server side application can use those parameters for controlling a multimedia project. The communication is based on the Open Sound Control (OSC) protocol for integrating multimedia equipment and software.

Two most important contributions of this research are (1) selection of video features convenient in general cases of controlling multimedia systems using the input video and (2) a technical solution which allows seamless real time video analysis on the Android operating system. Sound My Vision can serve as a versatile and generic controller intended for multimedia artists, musicians, lighting designers, contemporary dancers, and other performance artists.

SOUND MY VISION

As an OSC controller, Sound My Vision is comparable to existing applications such as andOSC, Kontrolleur, and OSCdroid. These applications are intended to send control parameters over the wireless network and they are not capable of producing sounds by themselves. The concept of mobile controllers usually implies a certain level of flexibility in sense that users can define parameter names and ranges.

Unlike the mentioned applications, Sound My Vision additionally supports real-time extraction of video features from the input camera. Video features are treated in the same ways as other parameters collected from sensors and from the touch screen. The general data flow is illustrated in Figure 1.



Figure 1. Sound My Vision as a versatile controller for interactive and multimedia works.

The application collects the following parameters from sensors, buttons, and the touch screen: (1) physical movements of the device (orientation, linear acceleration, and radial acceleration), (2) proximity of near objects, (3) geographic coordinates, (4) audio volume level controlled by the side buttons, and (5) coordinates of the point where the user touched the screen.

On the other hand, the video features extracted in real-time are selected to describe the movements in the scene and the general characteristic of the input image as described later in more details. For that reason, the following features are extracted: (1) amount of movement, coordinates, size and inclination of the moving object, (2) level of details in the current image, and (3) level of brightness of the current image.

The user can choose which of these parameters will be calculated and sent over the network as shown in Figure 2. Additionally, the application provides a possibility to define arbitrary OSC names and value ranges for each of the parameters. This feature allows users to adapt the output format in order to simplify the server-side implementation.



Figure 2. Screenshots from the application: home screen (left) and configuration of controls (right).

The parameters calculated from raw sensor data and features from the input video were selected based on modalities of interaction which they can provide in certain use cases. Additionally, we have also taken into consideration availability of sensors in popular Android devices. The rest of this section explains design decisions regarding selection and calculation of sensor data and video features.

Sensor data

The Android operating system supports three general categories of sensors: (1) motion sensors which measure acceleration forces and rotational forces along three axes, (2) environmental sensors which measure air temperature and pressure, illumination, and humidity, and (3) position sensors which measure physical position of a device.

Mobile applications for controlling musical expression usually employ movement and orientation sensors in order to translate physical state and movement of the device to the musical content. One such example is a

system for sound synthesis on microstructure level based on the movement [9]. It receives raw data captured from the movement sensors, extracts relevant statistical features, and maps them to parameters of a dynamic stochastic synthesizer using fuzzy logic.

In order to provide raw data which can be used either directly or for extracting higher-level features, Sound My Vision reads data from gyroscope, linear accelerometer, and the rotation sensor.

Location data can be successfully used in various musical applications such as interactive compositions based on the user's location [10] and for navigation purposes [11]. To obtain the geographic data, Sound My Vision employs both location mechanisms supported in the Android platform – location based on the GPS signal and the network location provider estimated from WiFi and cell tower signals. Such an approach ensures the balance between accuracy, frequency of updates, and efficiency of the battery usage.

The proximity sensor provides a measure of how far away an object is from the device. It can be used for controlling sound modulations and effects via hand movements interacting with an infrared beam of light. Besides being a part of some commercial sound synthesizer, proximity sensors are used in the do-it-yourself community, and discussed in the academic literature [12].

In order to exploit all hardware sensors and inputs available in most Android devices, Sound My Vision also reads events from the volume buttons which are usually located on the side of the device. These buttons serve as a metaphor for increasing and decreasing a value of any parameter defined by the mappings on the server side.

Touchpad

If it is enabled by the user, the touchpad is located on the home screen. When the user taps on the touchpad, a small circle appears indicating the position where the user tapped. Any change of the position updates the coordinates X and Y of the circle which are sent to the server side. This way, the user can simultaneously control two parameters using an intuitive touch gestures. Similar two-dimensional controllers can be found in some commercial MIDI controllers and other mobile applications.

Video features

The research on the connection between movement and music languages is continuously yielding interesting results, insights, and technical tools for almost three decades [13, 14]. Important and widely represented set of techniques in that field is computer vision based motion capture. While some computer vision systems were developed to recognize specific groups of physical gestures such as hand movements [15], other serve as generic frameworks for further development and experiments.

The purpose of Sound My Vision is to provide lower-level features of the visual data captured by the camera. Those features are intended to be directly mapped to parameters of an interactive system, or used for higher-level analysis such as recognizing more complex gestures or changes in the scene.

In various usage cases such as interactive installations and dance performances, a moving object is usually the most interesting part of the scene. For that reason, Sound My Vision calculates a group of video features which identifies the position, size, and orientation of the moving object.

General image characteristics such as brightness and level of details may also be useful in applications which rely on analyzing synthesized graphic or complex scenography. Therefore, besides the mentioned movement features, Sound My Vision also extracts general image features of the current video frame. On the server side such features can be either mapped directly to control parameters (e.g. darker image causes the lower sound volume) or they can be observed in longer time frames to extract higher-level parameters (e.g. frequency of the blinking light on the stage).

When selecting the convenient video features, the following aspects were taken into account: (1) relevance in possible use cases, (2) abstractness of the feature, (3) dimensionality, and (4) feasibility of feature extraction on mobile devices. Regarding abstractness, the goal was to find features which are obviously and intuitively related to the current image or sequence. A gradual change in the observed scene should cause a similar change in the appropriate feature. On the other hand, the desirable characteristic was generality so that the selected features can be used to calculate other higher-level features on the server side.

The feature dimensionality was also an important factor, since vectors of numbers and multidimensional data structures would be highly inappropriate for being used as control parameters. For that reason, only scalar features were preferred in the selection.

Moving Object

Separation of the moving object from the background is achieved with two different methods. The user can choose which one is more convenient for the intended usage. The first method is based on background mixture models as initially proposed by Stauffer and Grimson [16]. This is an adaptive technique which relies on the assumption that every pixel's intensity in the video can be modeled using a Gaussian mixture model. Following a simple heuristic it can be determined which intensities most probably belong to the background.

The second method is based on a simple frame differencing. The application allows user to choose the image which represents the background at any moment by tapping the button on the screen. The moving object is determined by calculating the difference between every frame and the reference background selected by the user.

The advantage of the first method is that it does not require any user actions, while the advantage of the second method is somewhat faster detection of moving objects. If there are multiple moving objects in the scene, the application tracks the largest of them. The following features are extracted: (1) amount of movement, (2) coordinates of the moving objects, (3) its orientation, and (4) dimensions (height and width). In order to calculate coordinates, orientation, and dimensions, the algorithm

approximates the object's silhouette with an ellipse and calculates its center, angle, and length of axes.

The server-side system can use these parameters to recognize specific situations on the scene or calculate higher-level features of the movement. For instance, based on the coordinates of the moving objects, it would be possible to reconstruct its trajectory and thereby recognize a moving pattern or gesture. Similarly, based on the aspect ratio of the moving object, the system could identify the moment when a dancer spreads the arms.

Image Characteristics

In order to provide parameters which quantify some general characteristics of the image, we selected two simple scalar features. The first one is the average brightness calculated by averaging intensity values of all pixels in the image. It is larger if the image is brighter, so it is related to the amount and position of lights on the scene in relation to the camera. Therefore, the average brightness can be used to synchronize the scene lightning with the music or other multimedia modalities.

The second feature is the level of details. It indicates the textural quality of the image and can be used for differencing between scenes with various numbers of visual elements in it. The level of details is estimated by considering presence of edges in the image, so the number of pixels classified as edges is normalized with the image size. In order to detect edges, we used the Canny edge detection algorithm [17].

Visualization

The main screen of the mobile application by default shows the image from the camera. In addition, the user can turn on additional visual information which could help in understating how the feature extraction works. In particular, besides the original camera image, the application can display any combination of these visualizations: (1) detected moving objects in the image, (2) the ellipse which fits the largest moving object, and (3) detected edges in the image. These visualizations can be overlapped so that the user can see multiple visualizations at the same time together with the original image if it is enabled. Figure 3 shows several such examples.

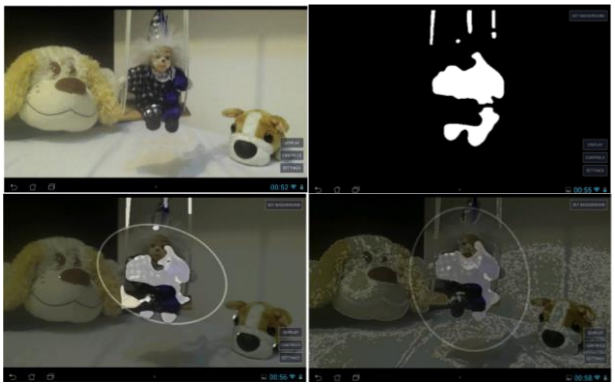


Figure 3. (1) The original image, (2) All moving objects, (3) All of the previous plus the ellipse fitting the largest object, (4) All of the previous plus edges.

TECHNICAL IMPLEMENTATION

The application Sound My Vision is developed both for smartphones and tablets with Android operating systems. It comprises of five modules: user interface, data processing module, OSC module, OpenCV Integrator, and the local database as shown in Figure 4.

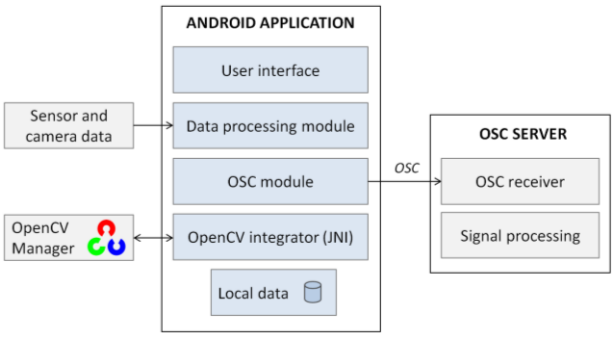


Figure 4. Software architecture.

The user interface adjusts adaptively to different device characteristics (orientation, resolution and dimension of the screen).

The local phone database stores the main properties of the application (OSC names and range values of the parameters, as well as the IP address and port of the OSC server) in order to provide their fast storage and retrieval.

Data processing module collects data from sensors, while OpenCV Integrator enables extracting features from the video signal captured by a camera. All functionalities related to image processing were implemented in C/C++ using the OpenCV library [18]. In order to load and integrate the library with an Android application, it is necessary to install OpenCV Manager on the device. It is an Android service targeted to manage OpenCV library binaries on end users devices. It allows sharing the OpenCV dynamic libraries between different applications on the same device.

The parameters obtained from sensors and camera are sent to the OSC server using the OSC protocol. OSC is an open, transport-independent, message-based protocol developed for communication among computers, sound synthesizers, and other multimedia devices. OSC server listens at the certain port and accepts messages.

OpenCV integration

All algorithms for video analysis in the application are implemented in the C++ programming language using the OpenCV library. The integration of the C++ code with the rest of the mobile application developed in Java has been made using the Java Native Interface (JNI).

OSC integration

The OSC integration is based on the JavaOSC library [19]. As networking is not supported on the main thread in Android applications, communication with the server is realized by using special background thread. Every supported parameter is sent to the server using unique

path and a single value, for example `/orientation-coordinateX 0.481`.

EXPERIMENTS AND USE CASES

In order to evaluate and demonstrate how the application works in different scenarios, we conducted several experiments some of which are described in this section. The server side was implemented using Pure Data, a visual programming environment for music and multimedia projects. The purpose of the server side was to read parameters received from the mobile application and use those parameters to evaluate whether they match expected values and to demonstrate how to control generated sounds or graphics.

Case 1: Playing notes using the touchpad

The first use case shows how the touchpad can serve as a simple interface for playing notes. One dimension is used for controlling the pitch of the note, while the other is mapped to the perceived loudness. Instead of allowing continues frequencies, in this experiment we selected discrete pitches which harmonically match the background music. Figure 5 shows data captured during one experiment and shows the relations between inputs from the touchpad and the acoustical qualities of the synthesized sound.

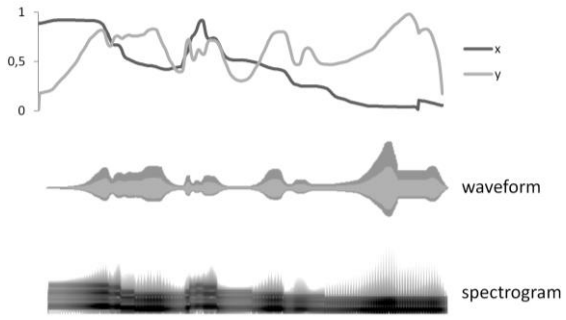


Figure 5. Data captured during a time frame of 12 seconds. The top chart shows values of the x and y coordinates where the user touched the touchpad, while the bottom images represent the waveform and the spectrogram of the synthesized sound.

Case 2: Controlling the 3D graphics and music

The second use case refers to controlling the rendered 3D graphics and music using the orientation of the mobile device. The orientation of the mobile device in the horizontal plane reflects to the orientation of the 3D object in the scene rendered by GEM for Pure Data [20]. If the user tilts the device down, than a violin appears in the scene and violins become prominent in the generated music. The loudness of the violins and the size of the violin object in the scene are determined by the amount of tilt. Similarly, when the user tilts the device to the side, a drum appears in the scene and the drum beat becomes audible. Figure 6 shows several snapshot captured during the experiment.



Figure 6. Dependence of the device orientation and the rendered 3D graphics.

Case 3: Demonstration of video features

In order to evaluate the accuracy of the video feature extractor, we conducted several experiments and compared the results with the expected values. Figures 7 do 9 show snapshot captured during the experiments. Below each image there are values of observed video features.

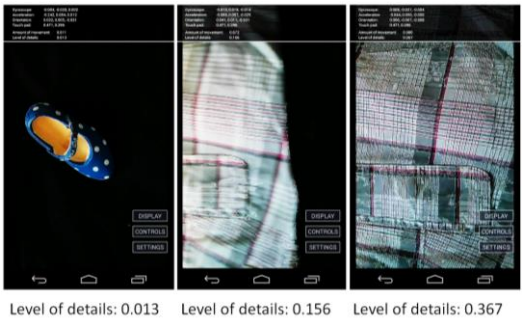


Figure 7. Correlation between the image texture and the level of details. Original image is here displayed with detected edges.

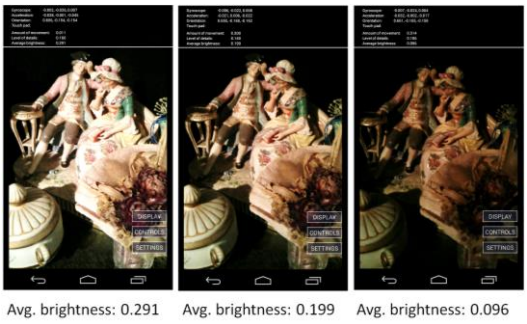


Figure 8. The average brightness in the image.

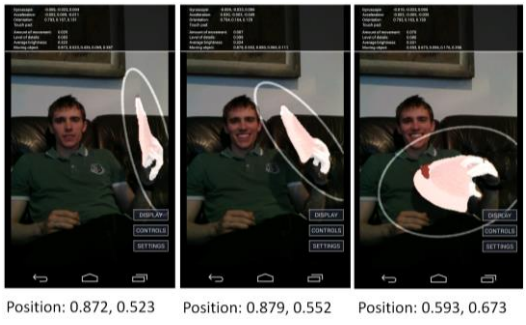


Figure 9. Moving object.

CONCLUSIONS

While most of existing mobile controllers of music and multimedia rely exclusively on the data from input devices and sensors, Sound My Vision additionally extracts video features from the camera image. Having a camera with real-time video analysis in a smartphone provides a lot of benefits starting from accessibility, affordability, and convenience to philosophical and artistic implications in sense of new opportunities of controlling music and multimedia.

In order to ensure usage of the application in various scenarios, the video features were selected to be general, low-level, and low-dimensional.

Based on the conducted experiments, we believe that Sound My Vision can contribute to the popularization of using multimedia in performance arts and for new possibilities in various modalities of artistic expression.

REFERENCES

[1] A. Tanaka, A. Parkinson, Z. Settel, and K. Tahiroglu, "A survey and thematic analysis approach as input to the design of mobile music GUIs," In Proceedings of the International Conference on New Interfaces for Musical Expression, Michigan, 2012.

[2] T. Kell and M. M. Wanderley, "A high-level review of mappings in musical iOS applications," In Proceedings of the International Computer Music Conference joint with Sound and Music Computing Conference, Athens, 2014, pp. 565–572.

[3] I. Neuman, C. Okpala, and C. E. Bonezzi, "Mapping motion to timbre: orientation, FM synthesis and spectral filtering," In Proceedings of the International Computer Music Conference joint with Sound and Music Computing Conference, Athens, 2014, pp. 671–677.

[4] G. Wang, "Designing Smule's iPhone ocarina," In Proceedings of the International Conference on New Interfaces for Musical Expression, Pittsburgh, 2009.

[5] T. Stool, "SoundScapeTK: a platform for mobile soundscapes," In Proceedings of the International Computer Music Conference joint with Sound and Music Computing Conference, Athens, 2014, pp. 1731–1735.

[6] J. Cannon and S. Favilla, "The investment of play: expression and affordances in digital musical instrument design," In Proceedings of the International Computer Music Conference, Ljubljana, 2012, pp. 459–466.

[7] S. Fels, A. Gadd, and A. Mulder, "Mapping transparency through metaphor: towards more expressive musical instruments," Organised Sound, vol. 7, no. 2, 2002, pp. 109–126.

[8] R. Wechsler, W. Frieder, and P. Dowling, "Eyecon: a motion sensing tool for creating interactive dance, music and video projections," In Proceedings of the Society of the Study of Artificial Intelligence and Simulation of Behaviour convention, 2004.

[9] G. Kreković and A. Pošćić, "Shaping microsound using physical gestures," In Proceedings of the International Conference on Computation, Communication, Aesthetics and X, Glasgow, 2015.

[10] J. C. Schacher, "davos soundscape, a location based interactive composition," In Proceedings of the In Conference on New Interfaces for Musical Expression, Genova, 2008.

[11] H. Zaho et al., "Interactive sonification of choropleth maps: Design and evaluation," IEEE multimedia, Special issue on Interactive Sonification, vol. 12, no. 2, 2005, pp. 26–35.

[12] E. R. Miranda and M. M. Wanderley, *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*, A-R Editions, 2006.

[13] A. Camurri, C. Canepa, F. Orlich, and R. Zaccaria, "Interactions between music and movement: a system for music generation from 3D animations," In Proceedings of the International Conference on Event Perception and Action, Trieste, 1987.

[14] R. Rowe, *Interactive Music Systems*, Cambridge (MA): MIT Press, 1993.

[15] O. Nieto and D. Shasha, "Hand gesture recognition in mobile devices: enhancing the musical experience," In Proceedings of the International Symposium on Computer Music Multidisciplinary Research, Marseille, 2013.

[16] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 1999, pp. 246–252.

[17] J. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, No. 8, Vol. 6, 1986, pp. 679–698.

[18] K. Pulli et al., "Real-time computer vision with OpenCV," Communications of the ACM, vol. 55, no. 6, 2012, pp. 61–69.

[19] C. Ramakrishnan, "JavaOSC" [*Online*]. Available: http://www.illposed.com/software/javaosc.html Accessed: April, 2015

[20] M. Danks, "Real-time image and video processing in GEM," In Proceedings of the International Computer Music Conference, Thessaloniki, 1997, pp. 220–223.

Addressing Tempo Estimation Octave Errors in Electronic Music by Incorporating Style Information Extracted from Wikipedia

Florian Hörschläger, Richard Vogl, Sebastian Böck, Peter Knees
Dept. of Computational Perception, Johannes Kepler University Linz, Austria
florian.hoerschlaeger@jku.at, richard.vogl@jku.at,
sebastian.boeck@jku.at, peter.knees@jku.at

ABSTRACT

A frequently occurring problem of state-of-the-art tempo estimation algorithms is that the predicted tempo for a piece of music is a whole-number multiple or fraction of the tempo as perceived by humans (tempo octave errors). While often this is simply caused by shortcomings of the used algorithms, in certain cases, this problem can be attributed to the fact that the actual number of beats per minute (BPM) within a piece is not a listener's only criterion to consider it being "fast" or "slow". Indeed, it can be argued that the perceived style of music sets an expectation of tempo and therefore influences its perception.

In this paper, we address the issue of tempo octave errors in the context of electronic music styles. We propose to incorporate stylistic information by means of probability density functions that represent tempo expectations for the individual music styles. In combination with a style classifier those probability density functions are used to choose the most probable BPM estimate for a sample. Our evaluation shows a considerable improvement of tempo estimation accuracy on the test dataset.

1. INTRODUCTION

A well-known problem of tempo estimation algorithms is the so called *tempo octave error*, i.e., the tempo as predicted by the algorithm is a whole-number multiple or fraction of the actual tempo as perceived by humans. Since these errors on the metrical level are not always clearly agreed on by humans, evaluations performed in the literature discount octave tempo errors by introducing *secondary accuracy* values (i.e. accuracy2) which also consider double, triple, half, and third of the ground truth tempo as a correct prediction. In average, these values exceed the primary accuracy values based only on exact matches by about 20 percentage points, cf. [1, 2].

While for tasks such as automatic tempo alignment for DJ mixes this can be a tolerable mistake, for making accurate predictions of the semantic category of musical "speed," i.e., whether a piece of music is considered "fast" or "slow," this discrepancy shows that there is still a need for improvement. To this end, several approaches have directly addressed the octave error problem, either by incorporating a-priori knowledge of tempo distributions [3], spectral and rhythmic similarity [1, 2], source separation [4, 5], or classification into speed categories based on audio [6, 7] and user-generated meta-data [8, 9].

The importance of stylistic context for the task of beat tracking has been stated before [10]. Similarly, in this work, we argue that there is a connection between the style of the music and its perceived tempo (which is strongly related to the perception of the beat). More precisely, we assume that human listeners take not only rhythmic information (onsets, percussive elements) but also stylistic cues (such as instrumentation or loudness) into account when estimating tempo. [1] Therefore, when including knowledge on the style of the music, tempo estimation accuracy should improve. Consider this simple example: If we knew that an audio sample is a *drum and bass* track, it would be unreasonable to estimate a tempo below 160 BPM. Yet our findings show that uninformed estimators can produce such an output. Therefore we propose to incorporate stylistic information into the tempo estimation process by means of a music style classifier trained on audio data. In addition to predicting multiple hypotheses on the tempo of a piece of music using a state-of-the-art tempo estimation algorithm, we determine its style using the classifier and choose the tempo hypothesis being most likely in the context of the determined style. For this, we utilize probability density functions (PDF) constructed from data extracted from Wikipedia articles (i.e., BPM ranges or values as well as tempo relationships).

The remainder of this paper is organized as follows. Section 2 covers a representative selection of related work. In section 3 the proposed system is presented. This includes our strategy to extract style information from Wikipedia, in particular information on style-specific tempo ranges. In section 4 we evaluate our approach using a new data set for tempo estimation in electronic music. The paper concludes with a short discussion and ideas for future work in section 5.

Copyright: ©2015 Florian Hörschläger, Richard Vogl, Sebastian Böck, Peter Knees 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] This assumption is supported by psychological evidence that identification of pieces as well as recognition of styles and emotions can be performed by humans within 400 msecs [11]. This information can therefore prime the assessment of rhythm and tempo which requires more context.

2. RELATED WORK

Gouyon et al. compare and discuss 11 tempo estimation algorithms submitted to the ISMIR'04 tempo induction contest [12]. Their paper shows that all submitted algorithms perform much better if tempo octave errors are considered as correctly estimated tempos. By ignoring this kind of error it was already possible to reach accuracies beyond 80%. A more recent comparison of state-of-the art tempo estimation algorithms is given by Zapata and Gómez [13]. Again the 11 algorithms compared in [12] are discussed along with 12 new approaches. In this comparison, again, the algorithm presented by Klapuri et al. [3] performs best, if tempo octave errors are ignored.

The tempo estimation algorithm described in [3] uses a bank of comb filters similarly to the approach by Scheirer [14]. One important difference is that while Scheirer uses only five frequency subbands to calculate the input signal for the comb filters, Klapuri et al. use 36 frequency subbands which are combined into 4 so-called "accent bands". This was done with the goal that changes in narrower frequency bands are also detected while maintaining the ability to detect more global changes which was already the case in [14].

The two approaches presented by Seyerlehner et al. [1] are based on two periodicity sensitive features (the autocorrelation function and fluctuation patterns) which are each used to train a k-Nearest-Neighbour classifier. The results obtained with this algorithm is at least comparable to the best results found in [12].

Peeters [2] uses a frequency domain analysis of an onset-energy function to extract so called spectral templates. During training, reference spectral patterns are created used in two different approaches. First in an unsupervised approach where clustering of similar spectral templates is done via a fuzzy k-means algorithm and second in an supervised variant where the 8 genres of the training dataset (ballroom) are used. Viterbi decoding is then used to determine the two hidden variables (tempo and rhythmical pattern) of the spectra templates to estimate the tempo of an audio track.

Gkiokas et al. [7] and Eronen and Klapuri [6] use machine learning approaches to further improve tempo estimation results. While [7] uses support vector machines to classify additional tempo cues in combination with the periodicity vectors, [6] uses k-nearest-neighbour regression in combination with the autocorrelation function of the accent signal. In [5], Elowsson et al. try to improve the tempo estimation results by separating percussive and harmonic sound sources and extracting different features on the resulting signals. Gärtner [15] proposes tempo detection based on non-negative matrix factorization and reports good results on a dataset comprised of urban club music.

Other approaches that aim at classifying music speed use external meta-data. Hockman and Fujinaga learn to classify music pieces into the categories "fast" and "slow" based on user tags found on YouTube [8]. Using simple frame-level features, they could reach a classification accuracy of 96%. Following a similar argumentation, Levy claims that the tempo octave error rate can be reduced by utilizing user tags of "fast" and "slow" [9]. Independent of a specific method for tempo estimation, Moelants and McKinney investigate the factors of a piece being perceived as fast, slow, or temporally ambiguous [16]. In this work, we focus on predicting the correct *beats per minute (bpm)* for a music piece rather than directly classifying music into speed categories.

3. METHOD

Our approach consists of a two stage tempo estimation process (visualized in figure 1). First, the *Tempo Estimator* generates $n = 10$ tempo estimates using a state-of-the-art tempo estimation approach. Second, the *Style Estimator* classifies the audio file into a style. Finally, the *Tempo Ranker* chooses the most probable tempo in the context of the classified style. In the following, we describe the used tempo induction approach (that also serves as a reference baseline for our evaluations), the construction of the style classifier and the strategy for picking the most probable tempo estimate. In the context of this work we also present an approach for extraction of music style specific tempo information from Wikipedia articles and how it is used within the described scheme.

3.1 Baseline Tempo Estimator

In the tempo estimation stage, any state-of-the-art tempo induction algorithm that can provide more than one tempo hypothesis can be used. In this work, we make use of the beat detection method introduced by Böck in [17]. The algorithm is based on *bidirectional long short-term memory (BLSTM)* recurrent neural networks. As network input, six variations of *short time fourier transform (STFT)* spectrograms transformed to the Mel-scale (20 bands) are used. The six variations consist of three spectrograms which are calculated using window sizes of 1024, 2048, and 4096 samples. For this process, audio data with a sampling rate of 44.1kHz is used which results in windows lengths of 23.2ms, 46.4ms and 92.8ms, respectively. In addition to these three spectrograms, the positive first order difference to the median of the last 0.41s of every spectrogram is used as input. The neural networks are randomly initialized and trained using manually annotated ground truth for beats from the *ballroom dataset*.[2] The trained neural network produces beat activation functions which are then used to calculate an autocorrelation function. The peaks in the smoothed autocorrelation function represent the tempo candidates expressed in *beats per minute (BPM)*. The height of the peaks corresponds to the probability of being the dominant tempo of the track.

3.2 Style Classification

The construction of the style classifier is based on the approach proposed by Seyerlehner et al. [18] that consistently yielded top-ranked results in the recent MIREX tasks on similarity estimation and genre and tag classification. As described in [19], this genre classification algorithm uses

[2] http://mtg.upf.edu/ismir2004/contest/tempoContest/node5.html

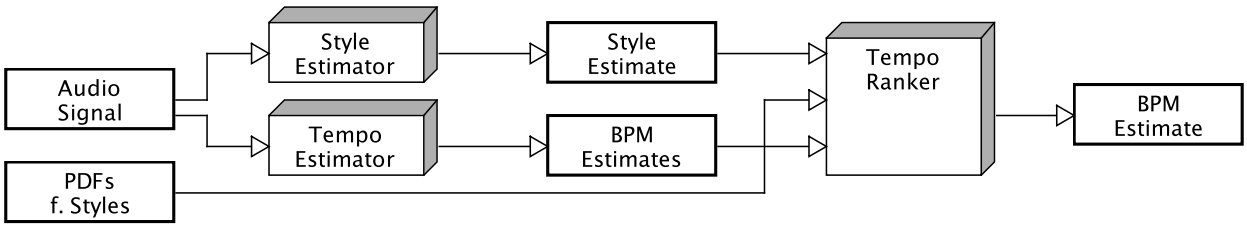


Figure 1. Schematic overview of the proposed system. The audio signal is used to derive multiple tempo estimations as well as a style estimation. The Tempo Ranker utilizes the style estimate, the tempo estimations and a set of predefined probability density functions (PDFs) to chose the most probable tempo in the context of the estimated style.

six block-level feature types, namely *spectral pattern*, *delta spectral pattern*, *variance delta spectral pattern*, *logarithmic fluctuation pattern*, *correlation pattern*, and *spectral contrast pattern*. Furthermore we add another vector, containing ten preferred BPM estimates produced by the tempo induction algorithm described in subsection 3.1. Directly concatenating the six block-level feature vectors would result in a combined feature vector with a length of 9,448 dimensions. To speed up training and reduce redundancy we first normalize those individual feature vectors and perform a separate *principal component analysis (PCA)* on each of the six feature types over the whole training set. For each feature type, we then take the first l dimensions of the PCA-transformed feature vectors which have a cumulative sum of their latent values (eigenvalues of covariance matrix) of more than 80%. The final feature vector is then obtained by concatenation of the reduced feature vectors as well as the vector containing the BPM estimates. Note that we do not transform the vector containing the tempo estimates. On the used dataset, this results in vectors with a length of 113 dimensions. Using this feature vectors we train a *random forest classifier* with 500 trees on a dataset containing 23,000 random samples downloaded from the *Beatport*® website. Those samples are almost equally distributed among the 23 different styles.

Samples are classified by first computing the block-level features and tempo estimates individually for each file. The six block-level feature vectors are individually (i) normalized by reusing the averages obtained from normalizing the training data (ii) transformed by reusing the coefficient matrices obtained by the PCA of the training data (iii) reduced by only taking the first l dimensions, where l are the same numbers as for the training vectors. The final vector is then composed of the six resulting vectors as well as the vector containing the BPM estimates and classified by the random forest.

Although not our primary interest in this work, we test the quality of the music style classification component by conducting 8-fold cross-validation on the training dataset. The average accuracy on the individual folds is 52.3 percent while the standard deviation is 1.0 percent. The overall accuracy on the *GiantSteps* tempo data set, that is used for the tempo estimation experiments, is 56.3 percent.

Style	from	to	slower than
chill-out	80	160	house
funk-r-and-b	80	160	
house	115	130	trance*
minimal	125	130	
electro-house	128	130	
glitch-hop	128	130	
hip-hop	124	135	
breaks	110	150	
indie-dance-nu-disco	120	140	
progressive-house	110	150	
pop-rock	130	140	
techno	120	150	psy-trance*
dubstep	130	142	
reggae-dub	130	142	
deep-house	110	170	
trance	120	160	
hard-dance	140	150	
psy-trance	140	150	
electronica	119	180	
drum-and-bass	130	180	
hardcore-hard-techno	160	200	
tech-house	180	220	

Table 1. Tempo ranges and tempo relationships for the *GiantSteps* dataset styles extracted from Wikipedia articles. Relationships marked with an asterisk where converted from *faster than* to *slower than* relationships.

3.3 Incorporating Style Information

To make use of the classified style in terms of tempo estimation, we need a suitable way to add tempo restrictions to each style. We do this by linking the different styles in our dataset to probability density functions, which are used to rank the different estimates. This section describes how the ranking works, the tempo information was extracted from Wikipedia and the probability density functions were modeled given the tempo information.

3.3.1 Deriving tempo information from Wikipedia articles

In this work we focused on deriving two different kinds of tempo information from Wikipedia articles:

- tempo annotations, which can either be BPM ranges or BPM values
- tempo relationships, which model whether one style is faster than another one (or vice verca)

This is achieved by a combination of heuristics and regular expression patterns, which were hand-crafted after reviewing a considerable amount of examples. Experience has shown that this task offers some major challenges: (i) multiple tempo annotations for styles, (ii) tempo annotations that need to be associated with other styles and (iii) the usage of synonyms and the complexity of the natural language.

For the purpose of this experiment we crawled a Wikipedia dump using JWPL [20] and Sweble [21] and used a hybrid strategy to decide if an article is about a music genre/style or not. If an article contains an instance of the *infobox music genre* [3] we assume it is indeed about a genre (infobox data is due to it's high quality utilized in many projects e.g. [22]). If this is not the case a WEKA [23] classifier is used. This classifier was trained using tf-idf weights, as well as some features based on infobox availability and the Wikipedia category and article graph [24] (e.g. number of referenced artists or the minimal category-graph distance to the root category of music genres). The training dataset was constructed by utilizing instances of the infobox music genre: articles containing the infobox as well as those referred as subgenres were added as positive training examples, articles referred as instruments and cultural origins were added as negative training examples. Given the resulting genres sub- and supergenre relationships were extracted by using data available in music genre infoboxes. Using this approach we were able to extract 775 genres (with a precision of 96.6 %) as well as 2.217 sub- and supergenre relationships from the Wikipedia snapshot created on 2nd May 2014.

In order to extract tempo information the derived genre-graph is traversed and all article texts are processed. The article texts are scanned for relevant sentences (e.g., containing a notation for BPM). Those sentences are matched with the hand-crafted patterns. Due to the fact that there are some genre articles that contain tempo annotations relevant to other genres it was necessary to perform a sanity check: For each match, the current section's title is cross-checked with a blacklist made up of the names and synonyms associated with the related genres (either sub- or supergenres of the current genre) as well as some words indicating that the content is about another genre (e.g. subgenres, related, influences). Whenever the section title matches an entry of the blacklist the system tries to associate the tempo annotation with the correct genre. This is done by matching names or synonyms of related genres within the current sentence or section title. All annotations that cannot be associated with a unique genre are dropped. To give an impression of what a sentence and a pattern might look like we provide the following example [4] (containing booth a range and a single value):

- "The average tempo of a minimal techno track is between 125 and 130 beats per minute. Richie Hawtin suggests 128 BPM as the perfect tempo."
  - between([^\d]{0,5}|)(**d**{**2,4**})([^\d]{0,5}|) (and)([^\d]{0,5}|)(**d**{**2,4**})
  - (**d**{**2,4**})[^\d]{0,5}BPM

Depending on the case, the bold face printed regions are then used as upper or lower boundary of a BPM range or a single BPM value. Tempo relationships are derived in a similar manner. Again all irrelevant sentences are dropped. Identified genre's names (and synonyms) are masked in order to make matching easier. Whenever a regular expression matches, a relationship is instantiated between the identified genres. This approach works reasonably well for extracting tempo annotations (precision = 90.2%) while the tempo relationship extraction (precision = 81.7%) would probably benefit from more sophisticated natural language processing techniques. Using this simple technique we were able to extract 94 tempo annotations - most of them associated with electronic music genres. Furthermore we were able to extract 38 tempo relationships.

3.3.2 Ranking of BPM estimates

In our approach the baseline estimator computes *ten* BPM estimates, those estimates are ranked by the Tempo Ranker. The Tempo Ranker uses predefined probability density functions (PDFs) to choose the most likely BPM estimate given by tempo estimation algorithm. To further formalize the behavior of the ranker, let S be the set of music styles, PDF_s be the probability density function (a function assigning a probability to BPM value) for the individual style estimate s and $E = \{e_0, e_1, .., e_9\}$ be the set of computed tempo estimates. Then the ranker chooses the tempo estimate $e_{max} \in E$ that maximizes the result of PDF_s (see equation 1).

$$e_{max} = \arg\max_{e \in E} PDF_s(e) \tag{1}$$

In the concrete test setup a PDF_s was derived for each style of the *GiantSteps* tempo dataset $s \in S$ except *dj-tools*. [5] In order to provide PDFs for the different styles of the *GiantSteps* dataset the first step was to create a mapping from Wikipedia genres to *GiantSteps* dataset styles. Since this step strongly depends on the dataset, it needs to be carried out manually. Given these mappings a BPM range $r_s = (min_s, max_s)$ is extracted for each style (the corresponding ranges are given in table 1). Also tempo relationships between the styles are considered. For the cases where the style s is not perceived to be slower than one of the other styles, PDF_s is defined analogous to the PDF of a normal distribution (see equation 2). With $\mu_s =$

[3] http://en.wikipedia.org/wiki/Template:Infobox_music_genre

[4] The syntax of regular expressions is conform to the Java Pattern class, for details see https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html

[5] No corresponding Wikipedia article could be found. Since there is no PDF for the style *dj-tools*, the Tempo Ranker skips ranking and chooses the first estimate in such cases.

$\frac{min_s+max_s}{2}$, $\sigma_s = \frac{\mu_s - \min_s}{3}$ and $\sigma_s \geq 3$ the PDF is basically a normal distribution with its center and standard deviation defined by the values of the range.

$$PDF_s(x) = \frac{1}{\sigma_s\sqrt{2\pi}} \cdot e^{-\frac{1}{2}(\frac{x-\mu_s}{\sigma_s})^2} \quad (2)$$

For styles k that are (according to the Wikipedia tempo relationships) slower than another style $s \in S$, the PDF is modeled analogous to the PDF of a gamma distribution (see equation 3). With $\gamma = 3$ (the shape parameter), $beta_k = 0.4 - \lfloor \frac{(max_k - min_k) - 15}{15} \rfloor 0.05$ (the decay parameter - this parameterization of β_k defines the PDF's decay wrt. the difference of min_k and max_k, i.e., the larger the difference, the lower the decay and therefore the PDF is smoother fading towards max_k) and $\mu_k = min_k - \frac{\gamma}{\beta_k} - \frac{max_k - min_k}{4}$ (the position parameter trying to position the PDF in a way that the PDF reflects the *slower* relationship). The resulting PDFs are visualized in figure 2.

$$PDF_k(x) = \frac{(\frac{x-\mu_k}{\beta_k})^{\gamma-1} exp(-\frac{x-\mu_k}{\beta_k})}{\beta_k \int_0^\infty t^{\gamma-1}e^{-t}dt} \quad (3)$$

4. EVALUATION

In this section we describe the conducted experiments, introduce the used dataset and discuss the results. All experiments were conducted using the *GiantSteps* tempo dataset. For every algorithm we provide accuracy1 and accuracy2 within a $\pm 4\%$ tolerance window. Accuracy1 considers an estimate to be correct if it is within $\pm 4\%$ of the true tempo. Accuracy2 also considers an estimate to be correct if it is within $\pm 4\%$ of either a third, half, double or triple of the true tempo.

4.1 Experiments

For the evaluation of our approach we conducted two experiments. In the first experiment we test the composition of style classification and tempo ranking, as visualized in figure 1 (for details see section 3). First the style estimation is carried out and ten tempo estimates are computed. Given those tempo and style estimates the Tempo Ranker chooses the most probable tempo based on the probability density function of the estimated style. This experiment therefore tests the impact of the overall composition of style estimation and ranking based on the data obtained from Wikipedia on tempo estimation accuracy. For simplicity this experiment is referred to as *wikidata-1*.
In the second experiment we test tempo ranking with respect to a known style, hence this shows how well the ranking itself performs given correct style assumptions. The style estimation step is skipped and the ranking algorithm is provided with the correct style. The experiment therefore tests the actual impact of the ranking procedure on the tempo estimates. For simplicity this experiment is referred to as *wikidata-2*.

We compare our results with tempo estimators, that are shipped with popular DJ tools. Namely *Cross DJ Free* [6], *Deckadance v2 (trail)* [7] and *Traktor 2 PRO*. [8] We argue that those estimators are tailored for electronic music and therefore should be able to perform well on the dataset.
Each of the products enables the user to choose some parameters for BPM prediction. *Deckadance* offers to choose among a predefined set of lower bounds, based on the ranges extracted from Wikipedia we decided to use 80 BPM. In the *Traktor* option pane the user can choose between a predefined set of tempo ranges, we decided to evaluate two ranges: 88-175 BPM (*TraktorA*) and 60-200 BPM (*TraktorB*). *CrossDJ* also provides a predefined set of tempo ranges, we chose 75-150 BPM for evaluation. In order to perform the evaluation we imported the audio files in the individual tools and analyzed them, the predicted values were later obtained from XML files (Deckadance, CrossDJ) or via ID3 tags that were instantiated during the analysis (Traktor).

4.2 The *GiantSteps* tempo dataset

The *GiantSteps* tempo dataset created in the course of the *GiantSteps* project [9] was obtained using the *Beatport*® website. [10] It contains tempo and stylistic ground truth for 664 samples. *Beatport*® is an online music store targeting producers and DJs of electronic music. For each track available on the website a preview sample can be downloaded. Furthermore, a variety of annotations for the tracks are provided – among them are *music style* presumably assigned by the composer out of the 23 maintained styles and the *tempo in BPM*. Since the BPM annotations might be calculated by an undisclosed algorithm they cannot be used as tempo ground truth data. However customers were encouraged to report false BPM annotations within a discussion forum. Typically users posted a reference to the track plus the correct BPM annotation. This discussion was crawled for comments containing a link to a track, the term BPM and a two or three digits long number. The association of a BPM annotation and a track was then conducted by putting this three pieces of information together. In cases where unambiguous information could be derived, the corresponding samples were downloaded. It can therefore be argued that this approach derives a human annotated dataset appropriate for evaluating tempo estimations. Unfortunately *Beatport*® recently abandoned this discussion forum.

As can be seen in table 3 this dataset has a strong bias towards the style *drum-and-bass*, 20% of the samples are within this style. This bias is most likely triggered by the fact that tempo estimation of drum-and-bass tracks is frequently affected by the tempo octave error, which implies many reports of incorrect tempi for this style.

4.3 Results and Discussion

Table 2 gives an overview of the obtained accuracy1 and accuracy2 values on the *GiantSteps* tempo dataset. In table

[6] http://www.mixvibes.com/products/cross

[7] http://www.image-line.com/deckadance/

[8] http://www.native-instruments.com/en/products/traktor/dj-software/traktor-pro-2/

[9] http://www.giantsteps-project.eu

[10] http://www.beatport.com

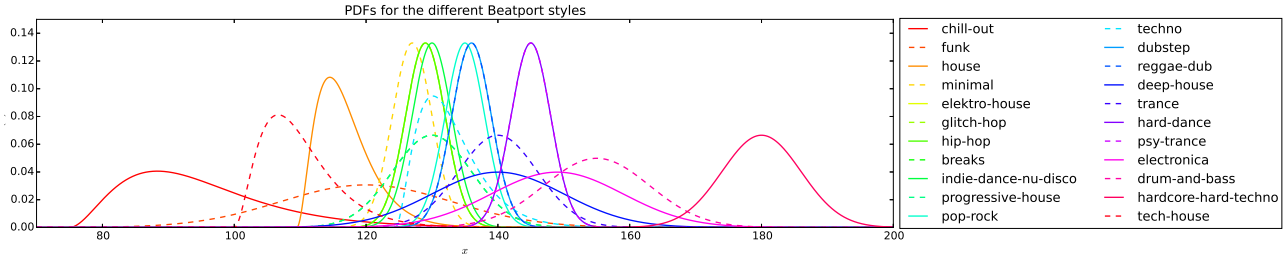


Figure 2. Probability density functions for the *GiantSteps* tempo dataset styles based on data extracted from Wikipedia articles.

	baseline	wikidata-1	wikidata-2	TraktorA	TraktorB	Deckadance	Cross DJ
accuracy1	45.33%	74.85 %	72.74%	76.81 %	64.46%	57.53%	63.25%
accuracy2	72.89 %	82.68%	80.72 %	88.55%	88.70%	81.48%	90.06%

Table 2. Tempo estimation accuracies for the different algorithms on the *GiantSteps* tempo dataset within a $\pm 4\%$ tolerance window. Apart from TraktorA (tempo range 88-175 BPM) the proposed approach clearly outperforms others and considerably improves the baseline performance.

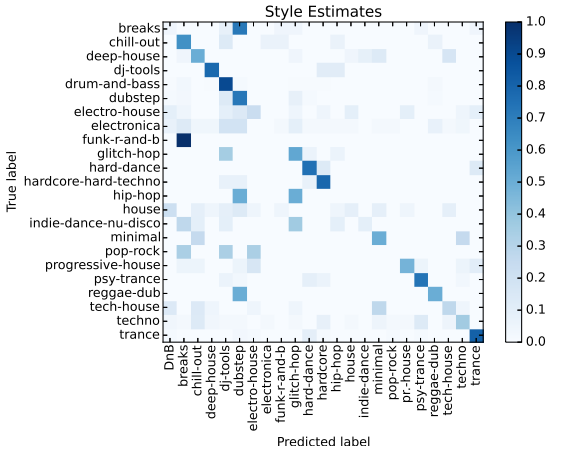


Figure 3. Confusion matrix for the style estimation task on the *GiantSteps* tempo dataset.

3 we provide detailed, per-style accuracies for the baseline, wikidata-1 and wikidata-2. Considering the accuracy1 values, which punish octave errors, it is apparent that the proper boundaries for the different styles help to increase tempo estimation accuracy. Compared to the performance of the baseline, we were able to increase the accuracy1 by 29 percentage points in scenario wikidata-1 and 27 percentage points in scenario wikidata-2. After having a detailed look on the per-style accuracy values in table 3 we noticed that especially for drum-and-bass, which makes up 20% of the dataset, we were able to considerably increase estimation accuracy (i.e. decrease the influence of the octave error). The baseline estimator only got 7.19 % right while the ranking boosts this value to 78.42 %. Overall we can report an increase of tempo estimation accuracy for most of the styles. A particularly interesting finding is, that despite the bad style classification performance wikidata-1 slightly outperforms wikidata-2, the tempo estimation accuracy still improves. Having a look at the confusion matrix in figure 3 reveals that styles which are hard to distinguish have similar tempo ranges (see table 1 and figure 2). This applies for instance to breaks and dubstep or reggae-dub and dubstep. Therefore is not too surprising that the ranking approach is able to chose a proper tempo estimate. For those styles (e.g. chill-out) for which the ranking decreases performance we assume that the extracted ranges do not properly represent the style. Except for the Traktor algorithm (TraktorA) our approach outperforms others in terms of accuracy1, we argue that despite reaching a higher accuracy the Traktor algorithm very much depends on the selected BMP range in order to reduce octave errors. Apart from that the algorithm is is highly tailored for tempo estimations of electronic music styles. In contrast to that our approach can (given proper input ranges for the styles of interest) be used for a wide range of music styles and does not enforce estimations within certain, predefined ranges. Note that we were not able to apply our approach on top of the audio-based tempo estimations given by Traktor, as our proposed method builds upon a tempo estimator that outputs multiple hypotheses. Also, it is not possible to set arbitrary tempo output ranges in Traktor, which would be another possibility of using external stylistic information.

In terms of accuracy2 other algorithms outperform our approach. Nevertheless it is apparent that wikidata-1 and wikidata-2 do only benefit by a small magnitude from the simpler task. While the other algorithms are able to increase their accuracy by between 12 and 27 percentage points our approaches only increase by about 8 percentage points. This means that there is only a small fraction of octave errors produced by the baseline that could not be corrected by the ranking procedure.

5. CONCLUSION

In this paper we have presented and evaluated a novel approach to further improve tempo estimation results of state-

style	#	baseline	wikidata-1	wikidata-2
drum-and-bass	139	7.19	78.42	83.45
dubstep	76	42.11	76.32	73.68
trance	74	75.68	97.30	98.65
techno	61	44.26	65.57	60.66
electronica	54	42.59	53.70	53.70
psy-trance	34	76.47	85.29	85.29
breaks	25	72.00	96.00	84.00
deep-house	24	75.00	75.00	83.33
house	23	47.83	65.22	73.91
tech-house	22	54.55	72.73	9.09
electro-house	22	63.64	77.27	68.18
progressive-house	19	57.89	89.47	94.74
glitch-hop	17	47.06	41.18	47.06
chill-out	16	62.50	43.75	37.50
hardcore-hard-techno	14	14.29	85.71	92.86
indie-dance-nu-disco	11	63.64	63.64	45.45
dj-tools	9	44.44	55.56	44.44
minimal	8	75.00	75.00	75.00
hard-dance	8	37.50	62.50	62.50
pop-rock	3	33.33	66.67	33.33
reggae-dub	2	0.00	0.00	0.00
hip-hop	2	50.00	50.00	50.00
funk-r-and-b	1	100.00	100.00	100.00
Weighted Average	664	45.33	74.85	72.74

Table 3. Primary tempo estimation accuracy (within 4% tolerance) of baseline estimator, wikidata-1 and wikidata-2 for the individual styles of the *GiantSteps* dataset.

of-the-art tempo induction algorithms. From the presented experiments we can see that using additional stylistic information of music can be beneficial for the task of tempo estimation. We were able to considerably reduce the amount of octave errors made by our baseline estimator and boost it's performance to be comparable with the performance of tempo estimators shipped with popular DJ tools. The facts that these (i) heavily depend on the predefined BPM output ranges / boundaries and (ii) do strictly enforce this parameters can be considered as drawbacks. Due to the use of probability density functions our approach does not come with this drawbacks. However, finding and assigning the right information in order to describe the styles is not trivial. To this end we proposed a strategy to extract tempo information from Wikipedia. The majority of tempo annotations derived from Wikipedia belong to electronic styles – this implies a limitation to the domain of electronic music. In our experiments, using information extracted from Wikipedia gives advantages over the uninformed baseline approach. The results we have obtained show the potential of tapping external and contextual information – unfortunately they are still rather inconsistent. The fact that some styles do not benefit from the extra knowledge suggests that we need to invest some more effort in the tempo range extraction strategy. Apart from that one could utilize more data sources or different tempo estimators. As mentioned before, the majority of the tempo annotations extracted from Wikipedia belong to electronic music styles, hence we were forced to carry out the experiments on an appropriate dataset. It is hard to predict how well the introduced method would perform on a dataset containing music from genres with wide tempo ranges like "classical," "metal," or "pop", cf. [25]. In these examples, tempo estimation would benefit only little if the chosen styles can not be linked to specific tempo ranges – which might be the case for e.g. "classical". On the other hand, if there are diverse subgenres with specific tempo ranges (consider "speed metal" vs. "rock ballad") the challenge is finding an appropriate set of styles and tempo ranges. For electronic music styles, where tempo can be one of the major factors that determine the style, we could show that the introduced tempo estimation approach improves results of state-of-the-art algorithms – mainly by preventing tempo octave errors.

6. ACKNOWLEDGEMENTS

The research leading to these results was performed in the *GiantSteps* project, which has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement no. 610591.

7. REFERENCES

[1] K. Seyerlehner, G. Widmer, and D. Schnitzer, "From rhythm patterns to perceived tempo," in *Proceedings of the 8th International Society for Music Information Retrieval Conference (ISMIR 2007)*, Vienna, Austria, Sept 2007.

[2] G. Peeters, "Template-based estimation of tempo: Using unsupervised or supervised learning to create better spectral templates," in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept 2010.

[3] A. Klapuri, A. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 342–355, Jan 2006.

[4] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stafylakis, "Music tempo estimation and beat tracking by applying source separation and metrical relations," in *Proceedings of the 37th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, Kyoto, Japan, Mar 2012, pp. 421–424.

[5] A. Elowsson, A. Friberg, G. Madison, and J. Paulin, "Modelling the speed of music using features from harmonic/percussive separated audio," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, Curitiba, Brazil, Nov 2013.

[6] A. Eronen and A. Klapuri, "Music tempo estimation with k-nn regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 1, pp. 50–57, Jan 2010.

[7] A. Gkiokas, V. Katsouros, and G. Carayannis, "Reducing tempo octave errors by periodicity vector coding and svm learning," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, Porto, Portugal, Oct 2012, pp. 301–306.

[8] J. Hockman and I. Fujinaga, "Fast vs slow: Learning tempo octaves from user data," in *Proceedings of the*

11th International Society for Music Information Retrieval Conference (ISMIR 2010), Utrecht, the Netherlands, 2010, pp. 231–236.

[9] M. Levy, "Improving perceptual tempo estimation with crowd-sourced annotations." in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, Miami, FL, USA, 2011, pp. 317–322.

[10] N. Collins, "Towards a style-specific basis for computational beat tracking," in *Proceedings of the 9th International Conference on Music Perception and Cognition (ICMPC9) and 6th Triennial Conference of the European Society for the Cognitive Sciences of Music (ESCOM)*, Bologna, Italy, 2006.

[11] C. Krumhansl, "Plink: "Thin Slices" of Music," *Music Perception: An Interdisciplinary Journal*, vol. 27, no. 5, pp. 337–354, June 2010.

[12] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, Sept 2006.

[13] J. Zapata and E. Gómez, "Comparative evaluation and combination of audio tempo estimation approaches," in *AES 42nd International Conference*, Ilmenau, Germany, July 2011.

[14] E. Scheirer, "Tempo and beat analysis of acoustic musical signals," *The Journal of the Acoustical Society of America*, vol. 103, no. 1, pp. 588–601, 1998.

[15] D. Gärtner, "Tempo detection of urban music using tatum grid non negative matrix factorization," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, Curitiba, Brazil, Nov 2013.

[16] D. Moelants and M. F. McKinney, "Tempo perception and musical content: What makes a piece fast, slow or temporally ambiguous?" in *In Proceedings of the 8th International Conference on Music Perception and Cognition (ICMPC8)*, Evanston, USA, August 2004.

[17] S. Böck and M. Schedl, "Enhanced Beat Tracking with Context-Aware Neural Networks," in *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, Paris, France, Sept 2011, pp. 135–139.

[18] K. Seyerlehner, M. Schedl, T. Pohle, and P. Knees, "Using block-level features for genre classification, tag classification and music similarity estimation," in *online Proceedings of the 6th Annual Music Information Retrieval Evaluation eXchange (MIREX-2010)*, Utrecht, the Netherlands, Aug 2010.

[19] K. Seyerlehner, G. Widmer, and T. Pohle, "Fusing block-level features for music similarity estimation," in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept 2010.

[20] T. Zesch, C. Müller, and I. Gurevych, "Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary," in *Proceedings of the Conference on Language Resources and Evaluation (LREC), electronic proceedings*. Ubiquitious Knowledge Processing, Universität Darmstadt, Mai 2008.

[21] H. Dohrn and D. Riehle, "Design and implementation of the Sweble Wikitext parser: unlocking the structured data of Wikipedia." in *Int. Sym. Wikis*, F. Ortega and A. Forte, Eds. ACM, 2011, pp. 72–81.

[22] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia," *Semantic Web Journal*, 2014.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[24] T. Zesch and I. Gurevych, "Analysis of the Wikipedia Category Graph for NLP Applications," in *Proceedings of the Second Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing*. Rochester, NY, USA: Association for Computational Linguistics, 2007, pp. 1–8.

[25] F. Pachet and D. Cazaly, "A Taxonomy of Musical Genre," in *Proceedings of Content-Based Multimedia Information Access (RIAO) Conference*, Paris, France, Apr 2000.

Web Audio Modules

Jari Kleimola
Dept. of Computer Science
Aalto University
Espoo, Finland
jari.kleimola@alumni.aalto.fi

Oliver Larkin
Music Department
University of York
York, UK
oliver.larkin@york.ac.uk

ABSTRACT

This paper introduces Web Audio Modules (WAMs), which are high-level audio processing/synthesis units that represent the equivalent of Digital Audio Workstation (DAW) plug-ins in the browser. Unlike traditional browser plugins WAMs load from the open web with the rest of the page content without manual installation. We propose the WAM API – which integrates into the existing Web Audio API – and provide its implementation for JavaScript and C++ bindings. Two proof-of-concept WAM virtual instruments were implemented in Emscripten, and evaluated in terms of latency and performance. We found that the performance is sufficient for reasonable polyphony, depending on the complexity of the processing algorithms. Latency is higher than in native DAW environments, but we expect that the forthcoming W3C standard AudioWorkerNode as well as browser developments will reduce it.

1. INTRODUCTION

Digital Audio Workstations (DAWs) have evolved from simple MIDI sequencers into professional quality music production environments. Contemporary DAWs equip home studios with multi-track audio and MIDI recording/editing capabilities, which enable musicians to turn their compositions into publication-ready master tracks. DAWs are standalone native applications, whose functionality may be extended using plug-ins. Plug-ins implement custom virtual instruments and effects processing devices, which co-operate inside the DAW host environment through host-specific application programming interfaces (APIs).

The primary application scope of a DAW is limited to music making in a local single user environment. In this sense, web browsers may be regarded as functional opposites of DAWs: browsers target a wide range of use cases, focus on networked connectivity between many users, and provide remote resource access in a global scope. Like DAWs, browsers have also matured in time from simple document viewing applications into rich interactive multimedia platforms. Moreover, the ever-increasing number of standardized web APIs and open source third party libraries continues to expand their scope of applicability. For instance, the recent Web Audio API extends the browser sandbox to fit in musical applications such as those presented in this work (see Figure 1). Browser functionality may be further increased with novel secure extension formats such as Emscripten and Portable Native Client (PNaCl) that are running close to native speeds, and without manual installation.

Copyright: © 2015 Jari Kleimola and Oliver Larkin. This is an open-access article dis- tributed 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.

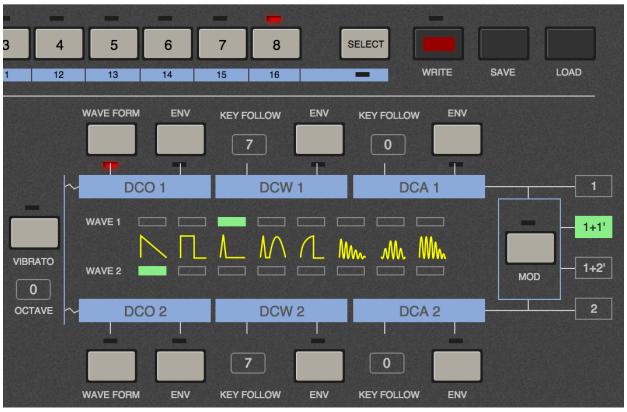


Figure 1. Detail of webCZ-101 user interface.

With these things in mind, we argue that enabling DAW-style virtual instruments and effects processors in web browsers – and integrating them with existing web APIs – introduces novel use cases that go beyond standalone DAW host scenarios. We give examples of four categories.

First, the Internet infrastructure may be utilized in direct distribution of software synthesizers, effects devices and their presets. Online DAW plug-ins may also be used in demoing native plug-ins without installation. More elaborate use cases include collaborative music making and live coding performances. *Second*, seamless integration with online web pages and strong support for multimedia suggests use cases for plug-in tutorials, interactive documentation, music theory lessons, online musical score rendering, audiovisual installations, and parameterized audio assets for online games. *Third*, wireless networking and support for various local communication protocols afford new interaction paradigms for software synthesizer control. Browsers also provide versatile tools for traditional graphical user interface (GUI) implementations. *Fourth*, the direct development approach – based on JavaScript (JS), HTML and CSS – encourages prototyping and exploration of novel audio synthesis and processing algorithms. *Finally*, when conforming to existing

web standards, these four categories are available in cross-platform and cross-device manner without plug-in host vendor control. We envision that many more scenarios will emerge.

This work introduces Web Audio Modules (WAMs), which are DAW-style plug-ins optimized for web browsers. Unlike most existing online synthesizer and audio effects implementations that build *on top* of Web Audio API, WAMs integrate *into* the Web Audio API via its script-based backend node: each WAM thus implements a full-blown software synthesizer or effects device inside a single Web Audio API node. The proposed WAM API strives to make these nodes reusable a) by standardizing how the enclosing web page loads and controls them, and b) by standardizing their DSP interface. The former enables development of novel application scenarios enlisted in the previous paragraph, while the latter enables audio algorithm development in JavaScript or cross-compiled C/C++. This, in turn, affords single code base for native and web audio plug-in implementation. However, in contrast to traditional web plugins, the cross-compiled WAMs load directly from the open web without manual installation (hence the term "module" instead of "plug-in"). The primary contribution of this work consists of:

- a proposal for a streamlined API that enables DAW-style virtual instruments and effects devices in web browsers.
- an implementation of the API in JS and C/C++.
- a minimal WAM example and two proof-of-concept WAM virtual instruments conforming to the API.
- a web service to aggregate WAMs and their presets.

This work also explores how WAMs integrate with existing and emerging APIs such as Web MIDI, WebGL, and Web Components.

The remainder of this paper is structured as follows. Section 2 reviews related work in native and web platforms. Section 3 details the API and its architecture, while section 4 explains how web pages and applications may embed WAMs, and describes their proof-of-concept implementations. Section 5 evaluates the implementations in terms of latency and performance, and finally, Section 6 concludes.

Source code, documentation, demos, and the web service are available via links at the accompanying website[1]

https://mediatech.aalto.fi/publications/webservices/wams

2. RELATED WORK

2.1 Native Plugin APIs

In 1996 Steinberg introduced Virtual Studio Technology (VST) and started a trend towards "in the box" audio production, where hardware effects and instruments could be replaced by native software equivalents running inside a DAW on personal computers. A publicly available C++ SDK[2] allowed developers to create their products as plug-ins – dynamic libraries conforming to a specific API, to be loaded by a "Host" application, which would typically be a DAW. A small industry had developed around the technology by the early-2000s with companies adopting the format along with a hobbyist community. Some host vendors, such as Apple and AVID created competing APIs allowing them a tighter control of the market and better integration with their platform. Although it represents a small fraction of musical instrument retail sales, the industry is still growing (at least in the USA) as can be seen in the NAMM 2014 global report[3].

Several plug-in APIs have prevailed and are used widely at the time of writing, including Steinberg's own VST2.4 and VST3, Apple's AudioUnit and AVID's AAX. In the open source community LADSPA and LV2 (LADSPA version 2) have been widely adopted. In the commercial arena the success and adoption of a particular API is often dictated by the host vendors and the market share they control, more than the merits of the API itself. VST plug-ins are supported by many hosts on the Mac and PC platforms, although the VST3 format, which is substantially different from VST2.4, has not yet seen widespread support outside of Steinberg's DAWs. The AudioUnit format only runs on the Mac platform and is the only format supported by Apple's popular Logic DAW. Access to the AAX SDK is controlled by AVID and only AVID can produce AAX hosting applications (such as ProTools). LV2 is platform agnostic and entirely open source with the most liberal license, but to date uptake has been mainly on Linux.

In 2003 a working group of the MIDI Manufacturer's Association (MMA) was set up to develop a non vendor controlled plug-in API Generalized Music Plug-in Interface (GMPI). Although this API never materialized, a draft of a list of requirements was produced based on the members' discussions[4]. This has informed the LV2 plug-in specification[5] and serves as a useful reference for the design of audio plug-in APIs.

Audio plug-in developers who want to make their software compatible with a variety of hosts and platforms and reach a wide market have to support multiple APIs and the complexity is increased by the need to provide cross platform GUI and file system features. For this reason many developers use an intermediate C++ framework such as JUCE[6] or IPlug[7] (or a proprietary solution) in order to develop an abstracted version of the plug-in which can then be compiled to multiple formats, platforms and architectures, saving development time.

Although we aim to introduce the functionality offered by the concept of native audio plug-ins to the web, the differences of the environment require a different approach to the API design, and the development of a new

[1] https://mediatech.aalto.fi/publications/webservices/wams

[2] http://www.steinberg.net/en/company/developers.html

[3] https://www.namm.org/files/ihdp-viewer/global-report-2014/

[4] http://retropaganda.info/archives/gmpi/gmpi-requirements-2005-04-05-final-draft.html

[5] http://lv2plug.in/gmpi.html

[6] http://www.juce.com/

[7] https://github.com/olilarkin/wdl-ol

API for WAMs provides an opportunity to improve upon some aspects of native APIs. Criticisms of existing audio plug-in APIs would include single vendor-control, unnecessary complexity/verbosity, ambiguity of operation (which thread calls which method and when), synchronization of user interface and DSP processing state, and multifarious preset formats which lead many plug-in developers to create their own preset format, thus increasing the problem.

2.2 Web Audio

The Web Audio API [1] is a W3C standard for enabling realtime audio synthesis and processing in web browsers. The API models audio algorithms as interconnected node graphs. The current node set includes 18 native nodes as general building blocks (e.g., classic waveform oscillators and filters), and a generic script node that enables arbitrary DSP algorithm implementations using JS. The Web Audio API is still in development, and the current ScriptProcessorNode (SPN) – which resides entirely in the main thread – will eventually be deprecated in favor of AudioWorkerNode (AWN). AWN splits its functionality between main and audio threads for reduced latency and increased performance. Web MIDI API [2] complements Web Audio API by offering access to local MIDI devices for control-oriented tasks.

The Web Audio API extension framework (WAAX) [3] abstracts Web Audio API node graphs as units, which may be parameterized and interconnected with other WAAX units and Web Audio API nodes. Its latest version turns units into more functional plug-ins, and provides two-way data binding between plug-in parameters and GUI elements. Plug-in parameters abstract Web Audio API AudioParams, and are therefore sample accurate and may be modulated at audio rates. WAAX targets only native Web Audio API nodes, and does not support scripted DSP algorithms. The WAM concept introduced in this work thus complements WAAX.

Web Audio Components[8] (WACs) are interoperable and reusable custom DSP units similar to WAAX plug-ins. WACs define a JavaScript Object Notation (JSON) manifest and publish themselves in a centralized registry. Each WAC also implements a constructor, metadata for parameter space description, and set of instance properties for interconnecting with other nodes. The current WAC registry has a RESTful[9] API, and contains eight components that operate as building blocks of larger DSP pipelines.

WebMidiLink[10] defines a simple textual language for transmitting MIDI and patch dump messages between a hosting web application and a conforming web synthesizer. The service maintains a list of synthesizer descriptors in JSONP format. A conforming synthesizer is loaded from the URL into an `iframe`, which enables cross-domain control using `window.postMessage()` function calls. At the time of writing, WebMidiLink registry contains 16 conforming web synthesizers. WAMs do not require an `iframe` container, but a simple wrapper can make them WebMidiLink conformant.

Emscripten [4] is a toolchain and virtual machine that enables cross-compilation of C/C++ code into high performance JS subset called asm.js [5]. Since asm.js *is* JavaScript, Emscripten modules (such as WAMs) work in all modern browsers without manual installation. PNaCl [6] loads LLVM bitcode (which is cross-compiled from C/C++) into the browser, and compiles that into native sandboxed code ahead of runtime. Recently in [7], native DAW plug-ins were ported to web environments as Emscripten and PNaCl modules. The work concluded that porting is feasible, and that web browsers are capable of running ported plug-ins without audible artifacts. The latencies were found to be higher than in native implementations, but expected to improve with AWNs.

As stated above, conforming to native DAW plug-in format such as VST carries unnecessary complexity that is irrelevant in web platform. Instead of porting native plug-in formats, the present work proposes a streamlined API that is optimized for web browsers and the AWN node. However, since AWNs are not yet supported by browser engines, the current WAM version uses SPN to emulate the AWN approach.

We also address a few Web Audio API shortcomings. Although the current node set exposes common and often used building blocks for various DSP implementations, the number of native node types (18) is insufficient to cover general DAW-style virtual instrument realizations. Parameterization and granularity of nodes raises further issues: for instance, Web Audio API oscillator nodes do not expose phase signals for external manipulation, and the filter nodes are simply textbook biquads. Single sample feedback connections between nodes are also problematic. The WAM API proposal relies therefore on script nodes that do not pose similar restrictions. The following section describes the proposed WAM architecture and API in detail.

3. PROPOSED API

3.1 Goals and Restrictions

The goal of the WAM proposal is to specify a streamlined API that enables DAW-style virtual instruments and effects processors in web browsers. The API needs to be simple, extensible, and strive for minimal latency and maximum performance. WAMs should load straight from the open web without manual installation, and they should integrate seamlessly with existing W3C APIs. WAMs may be developed in vanilla JavaScript, or cross-compiled C/C++ using the Emscripten or PNaCl toolchains.

The browser sandbox and W3C APIs pose specific restrictions to WAM implementations. The most prominent are: A) access to native operating system services and resources such as the local file system is restricted, and

[8] http://component.fm/#about

[9] https://en.wikipedia.org/wiki/Representational_state_transfer

[10] http://www.g200kg.com/en/docs/webmidilink/

B) custom DSP needs to run in a separate audio thread, while the rest of the WAM (e.g., GUI) resides in the main thread. Inter-thread communication is asynchronous.

3.2 Architecture

A WAM consists of Controller and Processor parts as shown in Figure 2. The **Controller** exposes the JS developer API, interfaces with other web APIs, and optionally provides the GUI. The **Processor** implements signal processing algorithms in JS or cross-compiled C/C++. Controller and Processor run in separate threads, and communicate through a "datachannel" using asynchronous events. In most cases, the events flow in a single direction (from Controller to the Processor), and asynchronous request-reply communication is only required during the initialization phase. The events are parsed and translated into method invocations at the Processor side in the **wrapper API**, which is exposed as a JS prototype or C/C++ header file. There is no traditional plug-in host concept in the API. Instead, the Controller hosts the Processor directly, and all interaction with the WAM and the web application code happens through Controller. This resolves the ambiguity of operation and synchronization issues present in some native plug-in APIs.

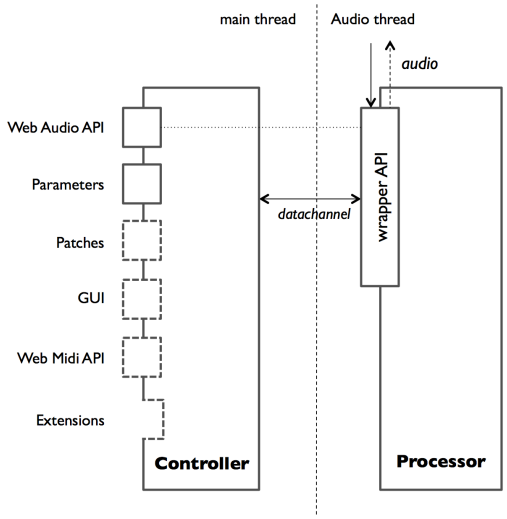


Figure 2. WAM architecture. Solid squares denote mandatory functionality that all WAMs need to implement, dashed ones are optional.

The division of functionality between the two WAM parts is as follows. The Controller holds the state (e.g., parameter values, loading and saving them from/into patches), while Processor implements the DSP (reflecting the parameter values as properly scaled synthesis parameters). Audio buffers are passed directly from/to the audio rendering pipeline, and they are thus not transferred between Processor and Controller. The parameter space and audio/event I/O configuration is declared as a JSON descriptor during initialization time, either at the Controller side or the Processor side (latter preferred). GUIs are outside the scope of this proposal, although they attach to the Controller using the functions defined in the API. Handling of the remaining optional functionality, i.e., Web MIDI API integration and patch handling is at the discretion of each individual WAM implementation.

3.3 Controller API

The Controller is implemented in vanilla JS and it runs in the web application's main thread. The mandatory functionality consists of lifecycle management (discussed later), Web Audio API integration (AWN-based implementation will move this to the Processor side), and parameter handling. The full Controller prototype, including optional functionality, is shown in Listing 1. Custom Controllers are derived from `WAM.Controller` using prototypal inheritance, and they decide which optional functionality to support.

```
WAM.Controller = function () { … }
WAM.Controller.prototype = {
  setup: function (actx,bufsize,desc,proc) { … },
  terminate: function () { … },
  connect: function (destnode, port) { … },
  disconnect: function (destnode, port) { … },
  getParam: function (id) { … },
  setParam: function (id, value) { … },
  setPatch: function (data) { … },
  postMidi: function (msg) { … },
  postMessage: function (verb,resource,data) { … },
  onMessage: function (verb,resource,data) { … } };
```

Listing 1. Controller prototype.

A WAM is exposed as a virtual Web Audio API AudioNode instance, which may be inserted into the node graph like any other real AudioNode. The JSON descriptor, which is thus formed either in Controller or Processor side during initialization time, serves as a contract between control and patch handling and the DSP implementation. The descriptor contains audio, MIDI, and data I/O configuration, as well as parameter space definition. WAM may contain any number of *audio* input and output buses, each with variable number of channels (thus enabling side-chaining). *MIDI ports* are bidirectional and *data ports* are provided for non-MIDI control streams, such as Open Sound Control (OSC). Finally, *parameter* definitions are optionally organized into a tree-like structure, which permits URL-like parameter addresses familiar from OSC. Each parameter is defined with id, name, datatype, min/max/default/step values, and modulation rate (control or audio). Parameter types include int32, double, enum, string, bool, and opaque chunk (void* + length).

3.4 Processor API

The Processor implements the realtime DSP algorithms conforming to the wrapper API, which operates as a bridge between the Controller and the Processor (see Figure 2). The wrapper API has bindings for JS and C/C++. Custom Processor implementations inherit `WAM::Processor` class, whose C++ interface is shown in Listing 2.

```
class Processor {
// -- lifecycle
public:
Processor() {}
virtual const char* init(uint32_t bufsize, uint32_t sr,
   char* descriptor);
virtual void terminate() {}
// -- audio and data streams + patches
virtual void onProcess(AudioBus* audio, void* data) = 0;
virtual void onParam(uint32_t idparam, double value) {}
virtual void onMidi(byte* msg, uint32_t size) {}
virtual void onMessage(char* verb, char* res, void* data,
   uint32_t size) {}
virtual void onPatch(void* data, uint32_t size) {}
// -- controller interface
protected:
void postMessage(const char* verb, const char* resource,
   void* data, uint32_t size) {}
uint32_t m_bufsize,m_sr; };
```

Listing 2. Processor interface.

3.5 WAM Lifecycle

Figure 3 shows WAM lifecycle as a sequence diagram. `WAM.Controller.setup()` first loads the processor script (which contains the implementation of the DSP code in vanilla JS or in Emscripten/asm.js), and calls the `createProcessor()` entry point at Processor side to create a new custom Processor instance. The Controller then initializes the Processor by passing buffer size, sample rate, and an optional descriptor as parameters. The Processor may choose to return a new descriptor as a JSON string instead of conforming to the Controller suggested parameter space (if any). WAM then enters runtime stage, which is aborted by invoking `terminate()`. Terminate disconnects the virtual AudioNode from the Web Audio API node graph, disconnects MIDI ports, and blocks the data-channel between Controller and Processor. Controller and Processor are eventually disposed by garbage collection.

Runtime control is available via `set/getParam`, `setPatch` `postMidi`, and `postMessage` functions, which are routed to the Processor side `onParam`, `onPatch`, `onMidi` and `onMessage` handlers. `Processor.postMessage()` is routed to the opposite direction.

During runtime, Web Audio API requests periodically a new block of samples. The request is dispatched to the `Processor.onProcess()` function, passing audio input and output buffers, as well as AudioParams (containing parameter automation and audiorate parameter modulation signals) in the first argument. The sample size is 32 bit float, normalized to unity range [-1,1]. Audio is non-interleaved, i.e., there is one buffer per channel. Custom Processor implementations may perform internal processing using double precision, although the Web Audio API input and output buffers are restricted to 32 bit floats.

Sample-accurate MIDI and data events are passed in the second argument, which holds a pointer to an ordered event queue. The queue entries are timestamped with sample offsets from the start of the current audio buffer. Since processing sample-accurate events produces overhead, the Processor needs to request them in the JSON descriptor. A detailed description of the optional functionality for data, MIDI, patches, and GUIs is available at the accompanying website[1] of this paper.

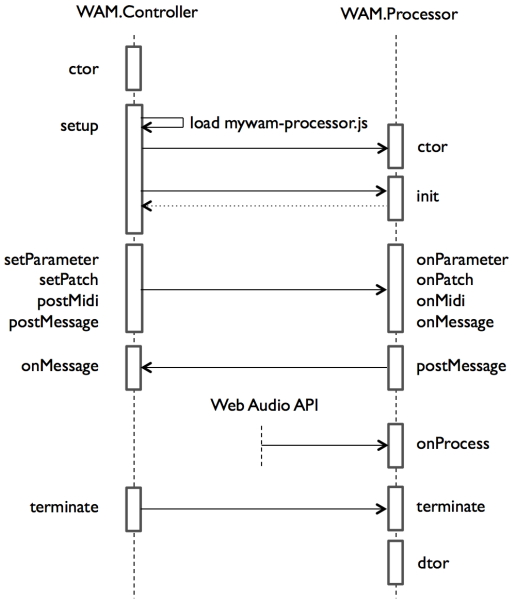


Figure 3. WAM lifecycle.

4. IMPLEMENTATIONS

4.1 WAM Usage

A Web page may embed a WAM by loading the supporting framework and the custom Controller implementation (lines 1-2 in Listing 3), and initializing the WAM in lines 4-9. The initialization script creates new Web Audio API AudioContext and the custom WAM in lines 4-5. Line 6 initializes the WAM instance by passing audio context and buffer size as arguments. The initialization function loads the Processor script asynchronously, and therefore returns a JS Promise that resolves in line eight. Line eight simply connects the custom WAM into the default AudioContext sink (i.e., the speakers). Another implementation might extend line eight into a more elaborate audio graph, for instance, by connecting the WAM into a convolution reverb node. Naturally, the audio graph may also chain WAMs together.

```
1 <script src="wam.min.js"></script>
2 <script src="sinsynth.js"></script>
3 <script>
4    var actx = new AudioContext();
5    var sinsyn = new SinSynth();
6    sinsyn.init(actx, 256).then(function ()
7    {
8       sinsyn.connect(actx.destination);
9    });
A </script>
```

Listing 3. WAM usage.

As stated previously, GUIs are outside the scope of WAM proposal. However, like WAAX [3], we have found Web Components and Polymer[12] useful for GUI implementation. Listing 4 shows an example. Line 1 loads the Polymer framework, while line two uses HTML imports to include the custom GUI. Line four

inserts the GUI into the web page, which may of course contain other HTML5 and WAM GUIs as appropriate. The `controller` attribute links the GUI with the custom WAM embedded in Listing 3.

```
1 <script src="polymer.min.js"></script>
2 <link rel="import" href="wam-sinsynth.html">
3 <body>
4   <wam-sinsynth controller="sinsyn"/>
5 </body>
```

Listing 4. Embedding WAM GUI into a webpage.

Listing 5 shows a minimal Controller implementation. The key to brevity is line five, which delegates most of the functionality to the WAM framework. The `setup` call in line three initializes the Processor. The third parameter denoting the descriptor is set to null, indicating that the Processor side is free to define its parameter space.

```
1 var SinSynth = function () {
2 self.init = function (ctx, bufsiz) {
3   return self.setup(ctx,bufsiz,null,"sinproc.js");
4   }};
5 SinSynth.prototype = new WAM.Controller();
```

Listing 5. Minimal WAM Controller (sinsynth.js).

Listing 6 shows the related minimal Processor implementation in vanilla JS (parts omitted for brevity). Line one is the entry point creating a new Processor instance. Line four returns a descriptor to define the number of audio input/output ports and parameter space. Lines 5-B implement the DSP algorithm for a simple monophonic sinusoidal synthesizer. Line six indicates silence, while line B indicates data in the output buffer. Line C receives MIDI input to update `voiceActive` and phase increment `phinc` member variables according to received status code and note number. Line D receives a parameter from the GUI to update the `gain` parameter, and finally, line E ties the implementation into the WAM framework.

```
1 function createProcessor() { return new SinProc; }
2 var SinProc = function () {
3 this.init = function (bufsize,sr,desc) {
4   return { … }}
5 this.onProcess = function (audio,data) {
6   if (!voiceActive) return false;
7   var out = audio.outputs.getChannelData(0);
8   for (var n=0; n<out.length; n++) {
9     out[n] = gain*Math.sin(phase*2*Math.PI);
A     phase = (phase + phinc) % 1; }
B   return true; }
C this.onMidi  = function (msg) { … }
D this.onParam = function (id,value) { … }};
E SinProc.prototype = new WAM.Processor();
```

Listing 6. Simple Processor implementation (sinproc.js)

WAM JS bindings enable rapid audio algorithm prototyping, since code changes are reflected by simply refreshing the browser window. WAM bindings also provide additional prototyping boost with a generic polyphonic synthesizer framework[1].

4.2 webCZ-101

webCZ-101 is an emulation of the Casio CZ101 Phase Distortion synthesizer, based on the DSP engine of Larkin's VirtualCZ plug-in[11] with a new user interface developed using Web Components/Polymer[12] (see Figure 1). VirtualCZ is implemented using the IPlug C++ framework, which the authors were able to extend to export the processor part of the WAM. This demonstrates how a closed-source plug-in, written in C++ can be ported to the WAM API, and that an existing cross platform plug-in framework can be adapted for WAMs. For some use cases (such as a web demo of a native plug-in, or interactive documentation) it would clearly be desirable to use the same C++ GUI code in the Web version, rather than rewriting it with a web-oriented GUI, but this is out of the scope of this work. In the current situation, where a different web GUI is necessary, porting a native plug-in is made much easier if the code for the DSP of the synthesiser or effect is clearly separated from the existing GUI code, which is something that is encouraged by modern plug-in APIs such as Steinberg's VST3. If GUI or native specific code is interleaved with the DSP, it can usually be easily excluded from compilation via the C preprocessor.

webCZ-101 implements five public methods from the WAM processor C++ interface.

- The init() method specifies the parameter space and I/O of the WAM as a JSON description as well as initializing the DSP with the sample rate and block size.
- The onProcess() method simply calls the DSP's block process method.
- The onMidi() method adds incoming MIDI messages to the DSP's internal MIDI message queue.
- The onPatch() method handles an opaque data chunk that is delivered from the controller after a patch is loaded in the GUI. The chunk is parsed and DSP parameters are updated.
- The onParam() method is called whenever a parameter change occurs in the GUI, and the DSP is updated accordingly.

The webCZ-101 controller side is written entirely in JS and handles the loading of CZ System Exclusive (sysex) files and GUI interaction. Since the source code for the processor part of the WAM is compiled to JS via Emscripten, the code is obfuscated to a degree and cannot be easily reverse engineered, however the JS and supporting files could potentially be used elsewhere, much like any other elements of a web page can be extracted.

4.3 webDX7

The Yamaha DX7 was the first affordable digital synthesizer, and it still remains the most sold hardware synthesizer to date. Since the theory of FM synthesis is well known, several virtual DX7 implementations exist both in open and closed source form. webDX7 uses the open source msfa[13] synthesis engine, which was initially de-

[11] http://www.olilarkin.co.uk/index.php?p=virtualcz

[12] https://www.polymer-project.org

[13] https://code.google.com/p/music-synthesizer-for-android/

veloped for Android OS and later encapsulated as a native VST plug-in[14] and its PNaCl port [7]. In the present work, the msfa DSP engine was wrapped inside the `WAM.Processor` class, and cross-compiled into an Emscripten module. The C++ implementation was then interfaced with a basic JS Controller class.

Although DX7 sounds were notoriously difficult to program, a large amount of patches are available on the Internet due to its commercial success. For instance, the collection at Kronos site[15] contains more than 200,000 patches gathered from the web. The collection contains many duplicates, but still offers a large corpus of presets that are usable with the WAM metadata preset format specification. We started exploring the metadata concept using *set4* from the Kronos collection. After removing duplicates, set4 contained 10236 unique patches preorganized into instrument categories and their subcategories. This provided a base for keyword-based classification. Each patch's data was then analyzed into a set of continuous-range perceptual features such as attack speed, duration, and DX7 algorithm. The analysis phase takes less than 100 ms for the 10236 patch corpus, which itself takes 437 kB when compressed.

We then explored their visualization to find out how WAMs integrate with other web APIs such as webGL. Each white particle in Figure 4 represents a DX7 patch. The initial screen (Figure 4a) shows a disorganized set of patches, in which particles stray across the screen space with random velocities and bounce from the screen bounds. The user is able to position a "magnet" over the patch cloud and move it around. Depending on the attributes of the magnet, it either attracts or repels patch particles based on their qualities. It is also possible to use multiple magnets with different attributes simultaneously. Particles are either orbiting a single magnet, or moving along a path between several of them. The PatchCloud implementation is based on a force-driven physical model [8] and implemented in three.js[16].

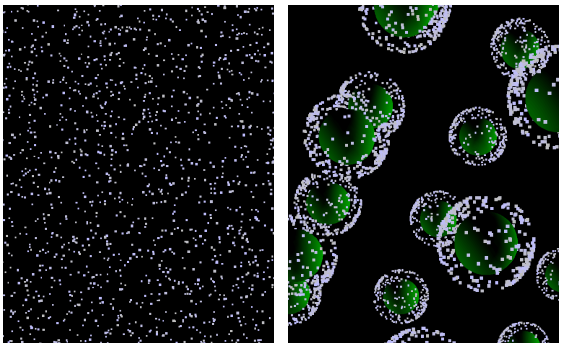


Figure 4. PatchCloud. (a) disorganized set, (b) arranged by DX7 algorithm.

Once a desired magnet constellation has been set up, the user may switch into patch audition mode. A cursor picks a single particle to send the associated patch to the webDX7 instance for audio rendering. Figure 4b shows a snapshot where patches have been organized into an evolving 3D setup based on their algorithms.

4.4 Web Service

We are aggregating WAM implementations and their patches in a public web service. The service maintains a central WAM registry, and exposes a RESTful API for querying and accessing the WAMs. It will thus operate as a distributed cloud VST folder for web applications, such as web DAW hosts. The web service has endpoints for headless WAMs (e.g., sinsynth.js in Listing 3), optional GUI implementations (wam-sinsynth.html in Listing 4), and standalone versions that are embedded inside an `iframe` using a WebMidiLink[10] manifest. For example, a web application may issue a request `"GET /synths /subtractive"` to get a list of all virtual analog WAM synthesizers in the registry, or access one directly by issuing `"GET /synths/subtractive/mysynth.js"`. A similar patch URL enables preset download and upload. Link to the service is available at the accompanying website[1].

5. EVALUATION AND DISCUSSION

5.1 Latency

WAM implementations were evaluated in terms of latency and performance (OSX Mavericks, MPB 2.2 GHz Intel Core i7, 256 sample buffer size, 44.1 kHz sample rate, Chrome v43, Emscripten optimization level –O2). The end-to-end latency was measured by connecting an external MIDI keyboard to a laptop via USB, and using the embedded microphone to capture the mechanical MIDI key click and the output sound of the WAM.

The latency measured 40-48 ms in all implementations as shown in column 2 of Table 1. On the average, this is ~32 ms higher than the theoretical 2×256/44100 = 11.6 ms SPN latency. To find out the cause for the increase, we implemented the baseline algorithm of Listing 6 directly in Web Audio API's `SPN.onaudioprocess()` handler, which gave 39 ms latency on the average. Comparing this to baseline WAM SinSynth, we note that WAM framework overhead is only 1 ms. Latency must therefore be related to the browser, operating system, and mechanical delay in the external MIDI keyboard.

In Chrome, browser-induced latency may be reduced by defining its buffer size with a command-line parameter. We found that buffer size of 32 samples gave lowest latency, as listed in column 3 of Table 1. Considering that the AWN node will remove the 11.6 ms SPN overhead, the latencies have potential to drop below 20 ms.

WAM	default	buffer = 32
SinSynth SPN	39	24
SinSynth WAM	40	28
webCZ-101	48	33
webDX7	45	31

Table 1. Latency in milliseconds.

[14] https://github.com/asb2m10/dexed
[15] http://korgpatches.com/patches/kronos/dx7_200k_collection
[16] http://threejs.org

5.2 Performance

Performance was evaluated in terms of polyphony, i.e. maximum number of simultaneous voices that still produce artifact free sound output. The results are shown in Table 2. The second column lists the number of voices in the WAM implementation, while the third column shows the performance in alternative implementations. webCZ-101 was compared against native standalone version (factory preset BRASS ENS. 1), and webDX7 against the Dexed PNaCl port from [7] (factory preset EPiano1). Baseline was provided by the minimal WAM synthesizer of Section 4.1 and its PNaCl version [7].

WAM	JavaScript	Native / PNaCl
webCZ-101	60	200
webDX7	17	128
SinSynth	280	350

Table 2. Performance in number of voices.

As expected, JS performance was lower than in native and PNaCl targets. webCZ-101 reached 30% of the native standalone VirtualCZ polyphony, which is acceptable. However, webDX7 achieved only 13.2% of the PNaCl polyphony, which suggests that its rather complex processing algorithm does not optimize well for JIT compilation. The performance can however be improved with larger buffer sizes.

5.3 Commercial Concerns

Although the web is based on open standards and web developers are accustomed to the fact that client-side code is easily viewable, there may be concerns relating to copy protection and monetization that could prevent companies from releasing their products as WAMs. The audio software industry is notoriously concerned with piracy, with many companies using hardware based copy protection systems for their products. It would be a significant challenge to reverse engineer the Emscripten-compiled asm.js into a readable and useable native form, but relatively easy to extract a WAM's entire code and use it elsewhere. Until pro audio on the web has matured and is seen to rival native platforms this is probably not a significant issue, and by that time attitudes may have changed and solutions may exist to protect and monetize products of this nature.

6. CONCLUSION

This paper introduced Web Audio Modules (WAMs), which are DAW-style virtual instruments and effects processors for web browsers. A streamlined API which is optimized for the forthcoming Web Audio API AudioWorkerNode was proposed, and two proof-of-concept WAMs were implemented. We found that it is trivial to add a degree of support for the WAM format to existing plug-in abstraction frameworks, and that JS/HTML/CSS provides a rapid prototyping environment for virtual instrument development. The implementations were evaluated in terms of latency and performance. The results show that although the default latency is relatively high, it has potential to fall below 20 ms with proper buffer size adjustments and the introduction of AWNs. Performance of SPN-backed JS modules is sufficient for multitimbral compositions, albeit not yet on par with corresponding native and PNaCl implementations.

We also explored WAM integration with other web APIs. Web Components were found useful in GUI implementation, while WebGL has clear potential in visualizing and browsing large preset libraries. The RESTful web service API for WAMs and their preset dissemination scales well for metadata-based patch queries and even accessing each preset with a unique URL.

Our future work will add support for PNaCl targets and AWN implementation once available. We shall also provide more WAM implementations and investigate how to allow a single code base to be used for both the web and native versions of an instrument or effect.

Finally, we would like to stress out that the API presented in this work is a proposal for community feedback. We welcome comments and contributions to make the API as usable as possible.

7. REFERENCES

[1] P. Adenot, C. Wilson, and C. Rogers, "Web Audio API," W3C Working Draft, Oct 10, 2013 and W3C Editor's Draft, April 03, 2015. Available online at http://www.w3.org/TR/webaudio/ and http://webaudio.github.io/web-audio-api/

[2] J. Kalliokoski and C. Wilson, "Web Midi API," W3C Working Draft, March 17, 2015 and W3C Editor's Draft, April 24, 2015. Available online at http://www.w3.org/TR/webmidi/ and http://webaudio.github.com/web-midi-api/

[3] H. Choi and J. Berger, "WAAX: Web Audio API eXtension," in Proc. Int. Conf. New Interfaces for Musical Expression (NIME'13), Daejeon, Korea, 2013, pp. 499–502.

[4] A. Zakai, "Emscripten: an LLVM-to-JavaScript compiler," In Proc. ACM Int. Conf. companion on Object oriented programming systems languages and applications (OOPSLA '11). New York, 2011, pp. 301-312.

[5] D. Herman, L. Wagner, and A. Zakai, "asm.js," Working Draft, Aug. 18, 2014. Available online at http://asmjs.org/spec/latest/.

[6] A. Donovan, R. Muth, B. Chen, and D. Sehr, "PNaCl: Portable Native Client Executables," White paper, Feb. 22, 2010.

[7] J. Kleimola, "DAW Plugins for Web Browsers," in Proc. 1st Web Audio Conference (WAC-15), Paris, 2015.

[8] D. Shiffman, The Nature of Code, 2012. Available online at http://natureofcode.com

TEMPO CURVING AS A FRAMEWORK FOR INTERACTIVE COMPUTER-AIDED COMPOSITION

Jean Bresson
UMR STMS: IRCAM-CNRS-UPMC, Paris
bresson@ircam.fr

John MacCallum
CNMAT - University of California, Berkeley
john@cnmat.berkeley.edu

ABSTRACT

We present computer-aided composition experiments related to the notions of polyrhythmic structures and variable tempo curves. We propose a formal context and some tools that enable the generation of complex polyrhythms with continuously varying tempos integrated in compositional processes and performance, implemented as algorithms and user interfaces.

1. INTRODUCTION

Tempo variations in musical performances significantly influence musical and rhythmic perception. Expressive musical timing and tempo curves (or *time maps*) are the object of previous studies in the field of computer music [1, 2]. In general the timing of beats and musical events is computed by the integration of tempo curves [3], and the compositional challenges are concentrated on the joint specification of these curves (or other expressive timing controls) and of a certain level of synchrony between simultaneous voices.

As a slightly different approach, we concentrate here on the notion of rhythmic equivalence in the context of time-varying tempos. Rhythms are prescriptive structures denoted by sequences of durations (also called *temporal patterns* [4]) when associated with a given (and possibly varying) tempo. Intuitively, it is possible to imagine that two different rhythms produce an equivalent temporal pattern if played following adequate tempo curves. Considering a rhythm as the convolution of another rhythm and a tempo curve, or as a superimposition of other rhythms and tempo curves, can be attractive musically as different representations of the same musical material can be suggestive of different interpretations in performance. In this paper, we explore and actualize this idea through computer-aided composition tools and techniques.

2. PRELIMINARY DEFINITIONS

In this section we introduce simple conventions that will be used throughout this paper. Our intention is not to discuss or overlap with the rich literature on rhythm theory and formalisation, but to provide keys for the reading and understanding of the subsequent parts.

Copyright: ©2015 Jean Bresson 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.

2.1 Rhythmic Figures / Durations / Tempo

It is important to first distinguish the notated/compositional form of a rhythmic structure (e.g. 𝅗𝅥 ♩. ♪) from the corresponding "phenomenological" rhythm, that is, the resulting sequence of durations or temporal pattern (e.g. 2s, 1.5s, 0.5s). These two forms are related functionally by a tempo value (in the previous examples, ♩ = 60, i.e. a quarter note corresponds to 1s).

We will use the upper-case character R to identify notated rhythms and lower-case r for the rendered temporal patterns. We note $\otimes$ the rendering operation associating a tempo τ to a rhythm R, yielding to $r : R \otimes \tau = r$.

2.2 Equivalence

We call *equivalent* two rhythms yielding equal temporal patterns and note this equivalence $R_1 \equiv R_2$. In other words:

$$\forall \tau, R_1 \equiv R_2 \Leftrightarrow R_1 \otimes \tau = R_2 \otimes \tau. \quad (1)$$

Recent works have delved into formalisms that allow one to search for equivalent notations for a given rhythm R, that is, $R_i \neq R$ such that $R_i \equiv R$ [5]. Most of the time, the tempo τ is used to convert rhythmic figures into actual durations, or conversely, to guide the search for the rhythmic notation that will best match a given temporal pattern (rhythmic quantification [6]). In both cases, it is the same on the two sides of the equality (as in Eq. 1).

In order to integrate the tempo as a variable parameter, we will group the rhythms and tempos in pairs (R, τ) and now call *equivalent* two pairs (R_i, τ_i) and (R_j, τ_j) which verify $R_i \otimes \tau_i = R_j \otimes \tau_j$. We also note this equivalence $(R_i, \tau_i) \equiv (R_j, \tau_j)$. Given a pair (R_i, τ_i), a limited numbers of rhythms $R_j \neq R_i$ will verify $(R_i, \tau_i) \equiv (R_j, \tau_j)$ if $\tau_j \neq \tau_i$. [1]

2.3 Polyphony

In order to work with polyphonic rhythmic structures, we also introduce the operator $\oplus$ which perceptually merges several rhythms or temporal patterns into a single one. We will use it for instance to compose complex rhythmic lines from simpler ones, or conversely to find sets of rhythms $\{R_1 ... R_n\}$ which verify: $\oplus_{i=1}^{n} R_i \equiv R_T$ (where R_T is called a "target" rhythm). [2]

[1] Rhythms verifying this property are equivalent rhythms ($R_j \equiv R_i$) modulo a "speed factor" τ_i / τ_j (e.g. ♩ ♩ 𝅗𝅥 and ♪♪♩).

[2] We simplify the notation here using R_i for expressing rhythms in general, that is either (R_i, τ_i) pairs or r_i.

Note that the $\oplus$ operator is hard to define and implement in the notation domain (see Figure 1a), but it is trivial in the "real" time domain, where there is a direct mapping between the clock time and the notes' onsets and durations (see Figure 1b).

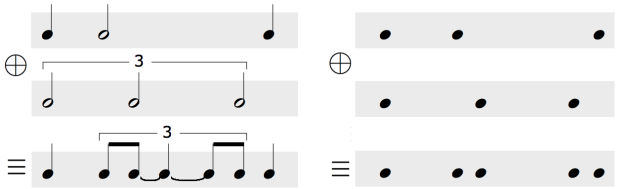


Figure 1: Merging rhythms a) in the notation domain and b) in the time/durations domain.

2.4 Varying Tempo

We now note $\tau_i(t)$ the function giving the value of the tempo τ at time t.

Considering the tempo as a variable function of time, we can assume that for any pair (R, τ) there exist an infinity of $(R_i, \tau_i(t))$ which verify $(R_i, \tau_i(t)) \equiv (R, \tau)$, and as a corollary, that for any given rhythms R_1 and R_2 and for any tempo τ_1 there exist a tempo function $\tau_2(t)$ such that $(R_1, \tau_1) \equiv (R_2, \tau_2(t))$. Conversely, given a *target* rhythm $r_T = (R_T, \tau_T)$ and a tempo curve $\tau(t)$ there must exist a rhythm R which verifies $(R, \tau(t)) \equiv (R_T, \tau_T)$.

Finding $\tau(t)$ or R here, or a combination of rhythms and tempos (R_i, τ_i) such that $\oplus_{i=1}^{n}(R_i, \tau_i) \equiv (R, \tau)$, is an appealing challenge from a musical point of view: it will allow us to render predetermined target rhythms (R_T, τ_T) using poly-temporal structures computed from time-varying tempo curves (see next section).

This problem is hard to solve with purely formal or algorithmic methods, and the search gets even more complex when combinations of rhythms are involved. As we will see below, supervised/interactive tools and heuristics provide interesting opportunities for compositional exploration.

3. COMPOSITIONAL CONTEXT

Musical polytemporality has been explored by many composers throughout the 20th and 21st centuries, however, the challenges involved in the composition and representation of polytemporal music have prevented many from progressing beyond experimentation to the development of a praxis. Gérard Grisey (*Tempus ex machina*), Iannis Xenakis (*Persephassa*), György Ligeti (*Kammerkonzert, 3rd mvmt.*), and Conlon Nancarrow (*Studies for Player Piano*) all produced works that explored the textures that become available to a composer when the clock that unifies performers is removed. However, the limited number of polytemporal works produced by these composers is representative of the challenges of constructing compositional systems and intuition in this domain. Dobrian [7] recently published a survey of compositional and technical issues related to polytemporal composition. In this section, we present recent works by John MacCallum that serve as a case study highlighting the need for a set of compositional tools that facilitate experimentation, exploration, and situated action.

3.1 Motivation

The conceptual motivation behind the works described below is predicated on the idea that synchrony between multiple musicians is a fictional construct of music notation. The concept of a musical "now" is not an infinitesimal, but rather, a small window, the width of which varies continuously between the imperceptibly small and the unacceptably large. As performers use the visual and auditory cues around them to negotiate and approximate a common tempo, they construct a system that is not *syn*chronous, but rather, *plesio*chronous in nature, i.e., *nearly* together, or close enough for the intended purpose. One's attention is rarely drawn to this fundamental feature of performance, except in the most extreme moments when the system begins to diverge from plesiochrony and approach true synchrony or diverge off to asynchrony. The works below foreground the human aspect of performance, albeit in a representative way, and push Platonic ideals inherent in music into the background.

3.2 Virtual and Emergent Tempos

In MacCallum's recent works, performers listen to a clicktracks that vary smoothly in tempo over time, independent of one another. The compositional challenge in these works is to construct musical material that unifies the different parts that are no longer bound by a common clock. *aberration* for percussion trio, [3] is an investigation into the use of composite rhythm and the emergence of a "virtual tempo" as a means of producing coherent ensemble material. In this work, tempo curves $\tau_i(t)$ were chosen using a random process and, in many sections of the piece, the rhythms R_i are chosen using a simulated annealing algorithm with the goal of producing $\oplus_{i=1}^{3}(R_i, \tau_i(t)) \equiv (R_T, \tau_T(t))$ where R_T represents a sequence of $1/16^{th}$ notes in a tempo $\tau_T(t)$ that can be imagined as the ideal tempo continuously "running in the background" that the musicians are trying to approximate.

The form of *aberration* was constructed largely independently of $\tau_i(t)$, and the material itself was algorithmically derived and then altered to draw the listener's attention to certain features of the relationship between the tempos at a given moment. The result is a complex rhythmic texture with a number of emergent properties unforeseen by the composer at the time of its creation. It is largely these underdeveloped textures that became the focus of a more intuitive and less systematic/process-oriented exploration in *Delicate Texture of Time* for eight players.

3.3 Methods

The composition of *aberration* relied heavily on a carefully constructed plan that was designed to project a small number of textures of interest to the composer who had, at

[3] http://john-maccallum.com/index.php?page=./compositions

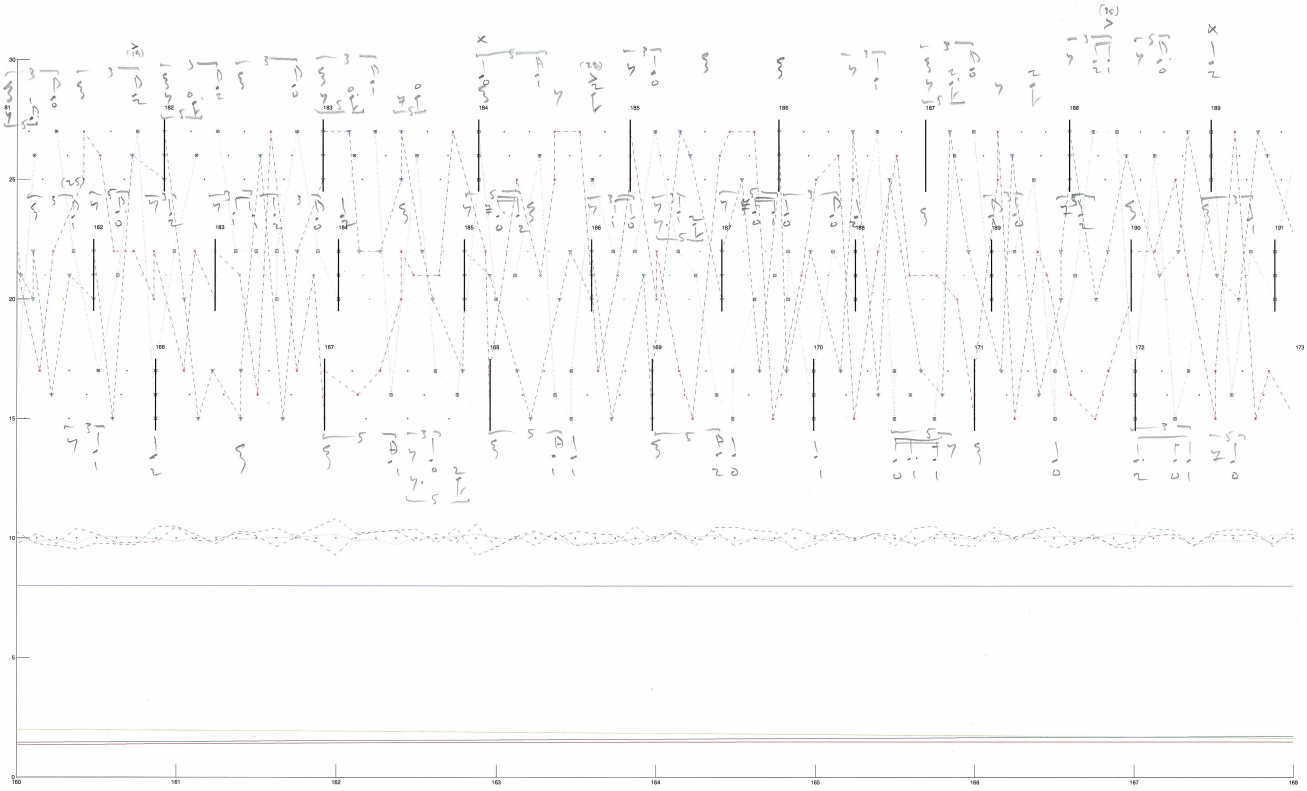


Figure 2: Compositional sketch of MacCallum's *aberration*.

that time, no intuitive sense of how they would be enacted in performance. The piece was constructed as follows:

1. $\tau_i(t)$ were created using a random process designed to produce curves that generally oscillate between a maximum and minimum tempo and change direction with some average frequency.
2. The time of every beat and subdivision (triplets, sixteenth notes, and quintuplets in this case) was computed for every voice from $\tau_i(t)$ and written to a file.
3. A simulated annealing algorithm was run to find R_i such that $\oplus_{i=1}^{3}(R_i, \tau_i(t)) \approx (R_T, \tau_T(t))$.
4. Matlab was used to create a template showing the position of every bar, beat, and subdivision for all voices, along with lines overlayed to show different outputs of step 3.
5. Steps 2–4 were repeated until the simulated annealing algorithm produced output with a high degree of voice exchange without too many instances of polyrhythms containing more than two subdivisions in half a beat.
6. Simple compositional sketches would be made to investigate the features of $(R_i, \tau_i(t))$.
7. Steps 1–6 were repeated until a suitably interesting output was produced.
8. The composition was then done directly on top of the template in pencil (Figure 2), and the results transcribed using Sibelius for the parts and OmniGraffle for the score (Figure 3 – see also Section 4.4).

There are a number of difficulties inherent in the steps listed above:

- The distance of the approximation $\oplus_{i=1}^{3}(R_i, \tau_i(t)) \approx (R_T, \tau_T(t))$ is directly dependent on $\tau_i(t)$ which were chosen *a priori* using a random process rather than being treated as free variables. This is not necessarily a problem, indeed, in the case of *aberration*, this was a feature of the work and a point of compositional investigation.
- Steps 1–7 offer little in the way of compositional intervention. When the simulated annealing algorithm produced output that was deemed unusable for one reason or another, it was difficult to apply constraints to have it avoid similar conditions during future execution.
- Step 8 is extremely cumbersome, error-prone, and forces the composer to commit to a given output once the composition of material has begun. If changes to any of the $\tau_i(t)$ need to be made at a later time, any material created must be discarded.
- Step 8 must be completed with little or no audition of material during the compositional process, preventing the composer from experimenting with material.

Delicate Texture of Time was produced in a manner similar to the method listed above, with the exception that the tools had become easier to use and more robust, and Adobe Illustrator was used in place of OmniGraffle, however the problems listed above remained present in the process.

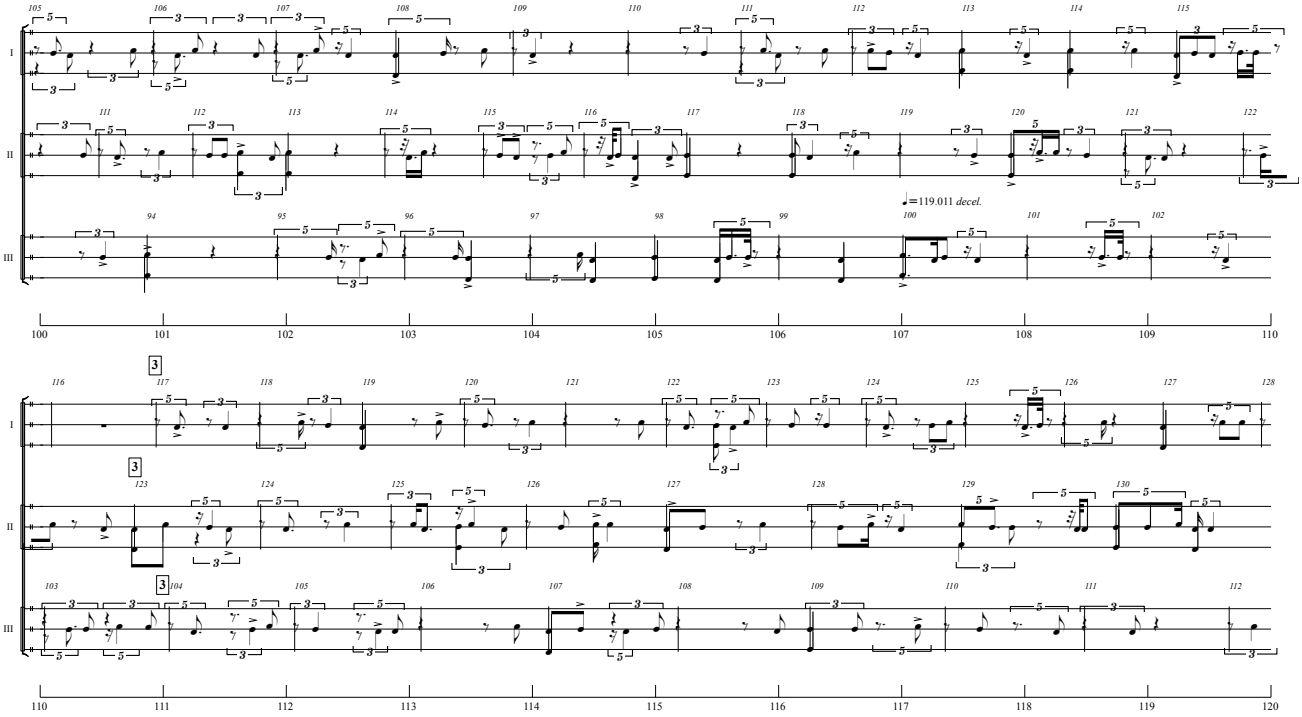


Figure 3: Score of MacCallum's *aberration*.

4. COMPUTER-AIDED COMPOSITIONAL PROCESS

In this section we present an implementation of the procedure described previously aided by the *timewarp*$\sim$ external for Max/MSP [8] and computer-aided composition processes implemented in the OpenMusic environment [9].

The compositional objective driving our discussion is the determination of n rhythmic lines (corresponding to n instrumental performers), each following a given tempo $\tau_i(t)$ $(i = 1...n)$ and merging to produce a target rhythm (R_T, τ_T).[4] The target rhythm can come from previous compositional processes or material, or it can be arbitrarily decided and specified by the composer. It can be expressed with rhythmic notation (R_T, τ_T) or (equivalently) directly as a sequence of durations (r_T).

Our problem is therefore to find R_i $(i \in \{1...n\})$ given r_T and $\tau_i(t)$, such that:

$$\oplus_{i=1}^{n}(R_i, \tau_i(t)) \equiv r_T$$

The combination of n lines exponentially increases the search space of this problem, which makes it slightly more complex than the equivalence issues mentioned in Section 2.4. As a first step, we will consider that $n = 1$ (and eventually drop the subscripts i to simplify the notation). As we will see, the presented methods easily scale to greater numbers of rhythmic lines.

4.1 Resolution with One Voice ($n = 1$)

As in Step 2 (Section 3.3), a first simplification we make to the problem is to preliminarily choose a number of possible pulse subdivisions for each voice R_i. Each subdivision (S) yields a regular rhythmic pattern (notated R_i^S) which will be used to compose a "quantification grid". Given these patterns R^{S_j} and the tempo curve $\tau(t)$ a sequence of duration $r^{S_j} = R^{S_j} \otimes \tau(t)$ can be computed for each subdivision (see Figure 4). Currently this part of the process is performed in Max/MSP using the *timewarp*$\sim$ external as described in [8]. The results (r^{S_j}) are communicated to OpenMusic through a simple file export/import protocol.

Note that at this point, if the subdivision is known and the tempo curve $\tau_i(t)$ does not change, the conversion of r^S back into R is relatively straightforward.

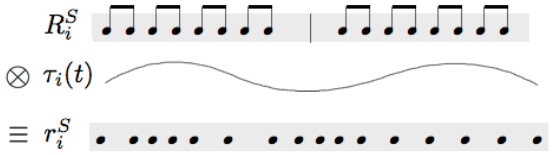


Figure 4: Generating a sequence of durations r_i^S starting from a beat subdivision S and a tempo curve $\tau_i(t)$ $(S = 2)$.

The same procedure is applied for different values of S $(S_1, S_2, ...)$ yielding a superimposition of lines (r^{S_j}) following the same tempo curve $\tau(t)$ (Figure 5).

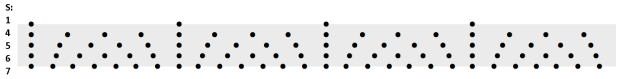


Figure 5: Superimposition of r^{S_j} $(S_j = \{1, 4, 5, 6, 7\})$.

[4] We suppose — especially in the case of complex tempo variations — that each performer will be assisted, typically by a click-track.

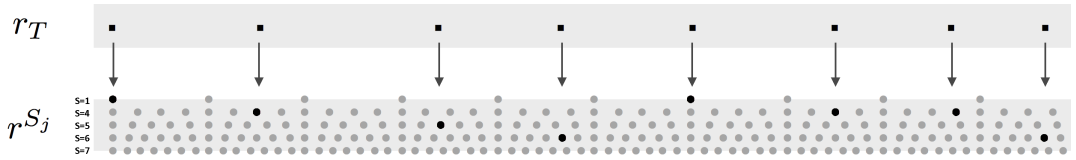


Figure 6: Finding elements of r_T in r^{S_j}.

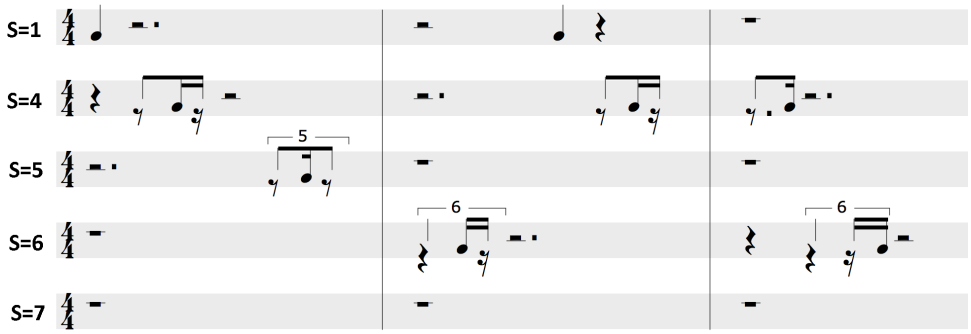


Figure 7: Reconstitution of R^{S_j} for each subdivision S according to the selection in Figure 6. (Note that $R^{S_7} = \emptyset$.)

The search procedure then consists in finding and marking an element in one of the different r^{S_j} which best matches each of the elements in the target r_T (see Figure 6). Constraints that govern acceptable combinations of subdivisions can also be applied to this search procedure to offer a degree of control over the general polyrhythmic complexity of the output.

From these marks it is easy to reconstitute a simple rhythm R^{S_j} for each value of S, containing one single rhythmic figure or subdivision (S_j) and considering every marked element in r^{S_j} as a played note, and every non-marked element as a silence (see Figure 7). An "abstract" rhythm R is then created regardless of the tempo curve $\tau(t)$, which will be equivalent to r if played back following $\tau(t)$.

A delicate part in the process is the merging of the different lines R^{S_j} back into a single voice R. As the tempo curve has been abstracted (it is the same for every R^{S_j}), some OpenMusic tools such as the *merger* function can be used for this operation [10]. Another solution is to perform a "local" quantification of the sequence r obtained from the combination of $r^{S_j} = R^{S_j} \otimes \tau_\alpha$, where τ_α is an arbitrary value of the tempo [6]. This quantification process is generally straightforward and reliable using existing tools (e.g. *omquantify*[5]), given the known tempo τ_α and the limited set of allowed subdivisions corresponding to the different S_j. Figure 8 shows the rhythm $R_0 = R^{S_1} \oplus R^{S_4} \oplus R^{S_5} \oplus R^{S_6} \oplus R^{S_7}$ merging the lines R^{S_j} from Figure 7. This rhythm corresponds to the target sequence r_T from Figure 6, if played following the initial tempo curve $\tau(t)$: $R_0 \otimes \tau(t) \equiv r_T$.

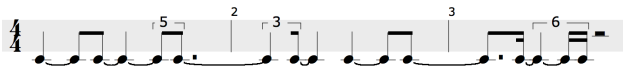


Figure 8: Rhythm merging the R^{S_j} from Figure 7.

[5] http://support.ircam.fr/docs/om/om6-manual/co/Quantification.html

4.2 Resolution with n Voices

The previous procedure is easily adapted to more than one voice. Considering our initial problem of finding R_i ($i \in \{1...n\}$) such that $\oplus_{i=1}^{n}(R_i, \tau_i(t)) \equiv r_T$, we just need to reproduce n times the process of generating the lines $r_i^{S_j}$ as in Figure 5.

The search is then extended so as to look up in the n different voices for an element of $r_i^{S_j}$ matching each element of r_T. According to the selection, separate sets of rhythms $R_i^{S_j}$ will be generated for each voice, merged into R_i and gathered in a polyphony as a result of the overall process.

Here as well, the graphical representation and alignment of polyphonic rhythmic structures with different, time-varying tempos is a tricky aspect of the polyphonic extension, but stands out of the scope of our present discussion. The OpenMusic *poly* editor allows for the assignment of tempo changes approximating $\tau_i(t)$ at every pulse of the n different voices, and to visualize/play these voices as a single score (see Figure 9).

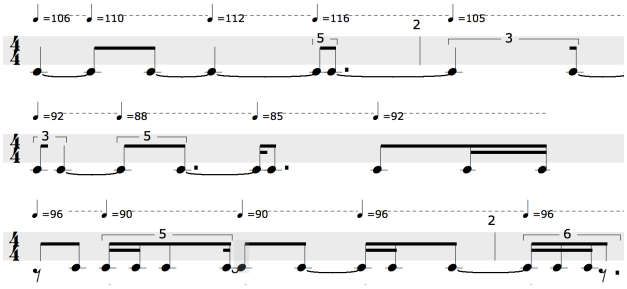


Figure 9: Aligned representation of 3 voices (R_1, R_2, R_3) in the OpenMusic *poly* editor, with tempo changes approximating $\tau_1(t)$, $\tau_2(t)$ and $\tau_3(t)$.

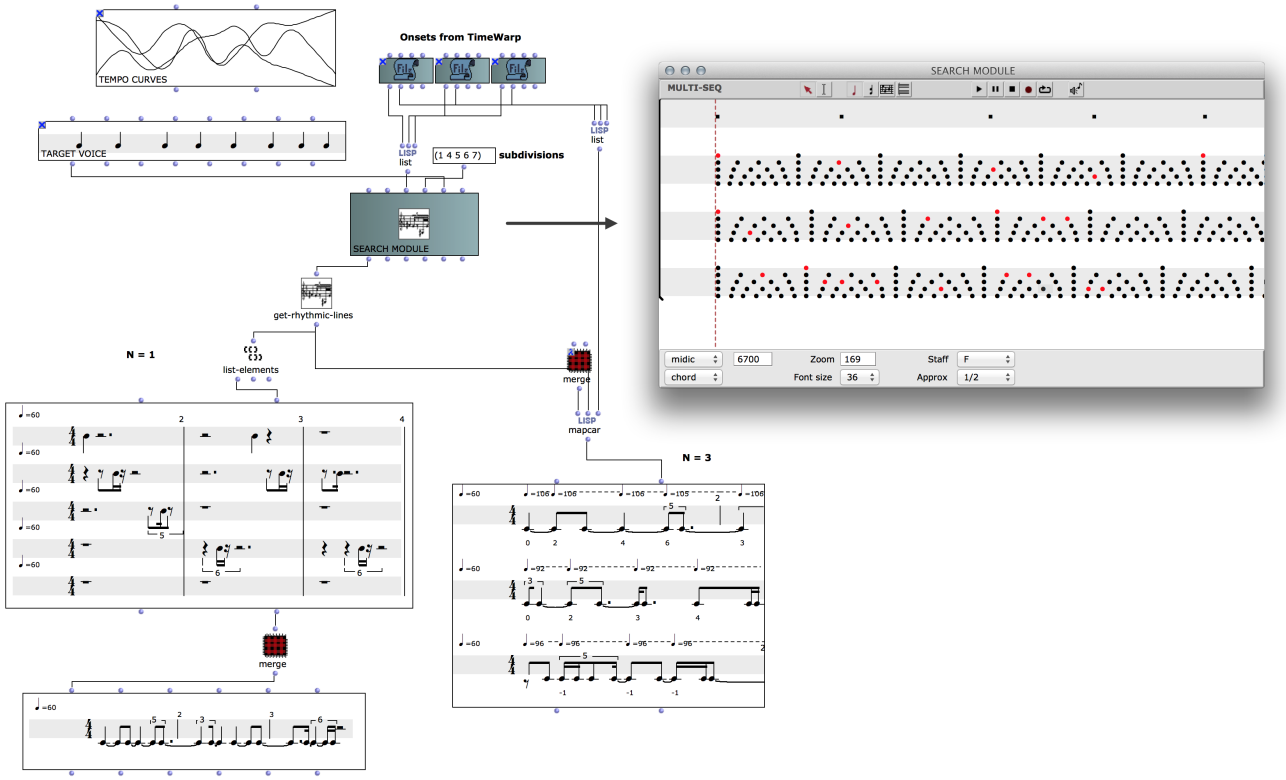


Figure 10: OpenMusic visual program implementing the rhythm matching process. The rhythm matching interface at the right allows to visualize/edit the matching between the target rhythm r_T and the various lines' tempo-varying grids $r_i^{S_j}$.

4.3 User Interface and Supervision of the Process

The process presented in the previous sections in principle could be completely automated and packaged in a black box. The main choices (inputs) for the user are the tempo curves $\tau_i(t)$, and the allowed rhythmic subdivisions S_j in the different voices. Still, visualizing the different steps is crucial to understand the process and eventually tweak these parameters in order to obtain relevant results, hence the choice and importance of a visual programming environment like OpenMusic where each of these steps is materialized by a module, and where all intermediate results can be inspected and edited (see Figure 10).

More importantly, composers' choices can be taken into account in the search part of the process where elements of the different $r_i^{S_j}$ are selected to compose the rhythms R_i. The algorithm may make unfortunate choices in the case of equivalent matches, and sometimes the "best" choice in terms of distance may not be the best in terms of readability, playability of the result, or because of any other compositional reason (e.g. controlling voice exchange or density, see for instance Step 5 in Section 3.3).

The main module of the visual program in Figure 10 therefore comes with a specific user interface (visible at the right on the figure) which extends the traditional *multi-seq* object of OpenMusic and allows the composer to visualize and make the choices of the elements in $r_i^{S_j}$ according to visual judgements or other arbitrary motivations. Computation can therefore temporarily stop here to leave space for manual edition and operations, before proceeding to downstream parts of the data processing.

4.4 A Note on Score Notation

In the works presented above, the parts that the musicians read from are typeset according to standard notational conventions in which spacing between notes is set in order to minimize page turns without sacrificing readability. The score, however, is prepared in such a way that the horizontal distance on the page between two notes N_i and N_j is proportional to the duration of N_i (see for instance on Figure 3). This redundant representation of time (rhythmic *and* proportional) allows one to see clearly the intended temporal relationships between the individual parts and to easily correlate moments in the score with the notation as seen by the performers.

Proportional notation that allows for complete and accurate control over the space between notes is impossible in most environments necessitating the use of graphic design software such as OmniGraffle for *aberration* or Adobe Illustrator for MacCallum's more recent works. The use of two separate environments for the score and parts can lead to differences between the two causing confusion in rehearsal.

Currently, OpenMusic scores represent time proportionally (*chord-seq*) or notationally (*voice*), but not both simultaneously. Recent work has been done to extend the notation objects and provide a hybrid (redundant) representation of time.

5. FROM COMPSOSITION TO PERFORMANCE

One of the goals of the compositions described in Section 3 is the representation of the inherently plesiochronous nature of human musical performance. It is the musical material itself, however, that carries this representation; those pieces do nothing to elicit a performance that would foreground this feature the way, for example, the distribution of the musicians across a large distance in space would. We present in this section two recent projects designed with this performative aspect in mind, and which also problematize the relationship between music notation and its realization.

5.1 Windows of Musical Time

If the "musical now" is a small window of continuously varying width, what would we find if we could construct a performance context that would increase the scale of that window to the point that its contents become musical material and even form? MacCallum's recent work *Hyphos* for alto flute, bass clarinet, viola, and electronics is a compositional study meant to explore this idea. As in *aberration* and *Delicate Texture of Time*, the performers listen to click-tracks to aid them as they perform long nonlinear *accelerandi* and *decelerandi*, however, in *Hyphos*, they are meant to only use the click-tracks in rehearsal and dispense with them in performance. The use of different slow, gradual, continuous changes in tempo for the three performers is designed to defamiliarize the performance context by removing the fictional shared tempo. As the performers attempt to follow their individual temporal trajectories, their vertical relationships vary over time with respect to those prescribed by the score, and the "window of the now", bounded by the two musicians leading and trailing the others, is brought to the foreground.

The compositional challenge here is to construct material that satisfies musical and æsthetic goals despite potentially extreme variation in performance. To this end, a set of tools that reconfigure the score to represent the temporal relationships of the individual parts during a given performance is essential for composers looking to gain a deeper understanding of the nature of the performative variability and develop compositional strategies and intuition that rely on it.

This work highlights the latent dualistic role of the score as providing a set of prescriptive instructions for performance, as well as being a representation of that which was performed. In the case of a score for two musicians, one may be able to follow the prescribed score and mentally reconcile the visual and auditory representations of the composition, however, as the number of parts increases, the complexity becomes unmanageable and the score, as a notational representation of the performance, is no longer of any value. Without a visual aid describing what actually happened, constructive communication with and between performers is hindered.

5.2 External Sources of Temporal Control

Hyphos, described in Section 5.1, was a study in preparation for a collaborative project between MacCallum and choreographer Teoma Naccarato that remains ongoing at the time of this writing. In *Choreography and Composition of Internal Time*, [6] pulses extracted from wireless electrocardiogram (ECG) units worn by dancers serve as click-tracks for musicians in real-time. The musicians render a score, but as in *Hyphos*, the temporal relationships between the different parts is in constant flux as the dancers perform choreography designed to affect change in their cardiac function that approximates the general contour of the precomposed $\tau_i(t)$. The use of biosensors here is intended to foreground the limits and variation of human bodies in performance, as well as to intervene in the compositional and choreographic processes.

6. CONCLUSION

We presented formal and compositional approaches for dealing with poly-temporal rhythmic structures in computer-aided composition. These formalisms and general workflow emphasize both computational and interactive considerations at manipulating musical time and rhythmic notations. Computer-aided composition provides interactive musical representations at the different steps of the process and allows for the combination of systematic/automated procedures with compositional interventions.

The presented framework is suitably general to be used for the generation and manipulation of rhythmic structures. It can, for example, be seen as a supervised rhythmic quantification tool, enabling the production of notated rhythmic approximations of a given sequence of linear onsets, using variable tempo tracks and/or polyrhythmic scores. We have also emphasized, in the discussion of recent compositional projects, how it is likely to be used in more interactive situations such as when the tempo information, for example, becomes a reactive input causing the different steps and views of the corresponding musical representations to update on the fly.

Acknowledgments

This work is part of the French National Research Agency project with reference ANR-13-JS02-0004.

7. REFERENCES

[1] D. Jaffe, "Ensemble Timing in Computer Music," *Computer Music Journal*, vol. 4, no. 9, 1985.

[2] H. Honing, "From Time to Time: The Representation of Timing and Tempo," *Computer Music Journal*, vol. 3, no. 25, 2001.

[3] J. Schacher and M. Neukom, "Where's the Beat? Tools for Dynamic Tempo Calculations," in *Proceedings of the International Computer Music Conference*, Copenhagen, Denmark, 2007.

[6] http://ccinternaltime.wordpress.com

[4] P. Desain and H. Honing, "Tempo Curves Considered Harmful," in *Time in contemporary musical thought*, ser. Contemporary Music Review, J. D. Kramer, Ed., 1993, vol. 2, no. 7.

[5] F. Jacquemard, P. Donat-Bouillud, and J. Bresson, "A Structural Theory of Rhythm Notation based on Tree Representations and Term Rewriting," in *Proc. Mathematics and Computation in Music*, London, 2015.

[6] C. Agon, G. Assayag, J. Fineberg, and C. Rueda, "Kant: a Critique of Pure Quantification," in *Proc. International Computer Music Conference*, Aarhus, Denmark, 1994.

[7] C. Dobrian, "Techniques for Polytemporal Composition," in *Proceedings of Korean Electro-Acoustic Music Society's 2012 Annual Conference (KEAMSAC2012)*, Seoul, Korea, 2012.

[8] J. MacCallum and A. Schmeder, "Timewarp: A Graphical Tool for the Control of Polyphonic Smoothly Varying Tempos." in *Proc. International Computer Music Conference*, New York/Stony Brook, USA, 2010.

[9] J. Bresson, C. Agon, and G. Assayag, "OpenMusic. Visual Programming Environment for Music Composition, Analysis and Research," in *ACM MultiMedia 2011 (OpenSource Software Competition)*, Scottsdale, AZ, USA, 2011.

[10] O. Delerue, G. Assayag, and C. Agon, "Etude et réalisation d'opérateurs rythmiques dans OpenMusic, un environnement de programmation appliqué à la composition musicale," in *Actes des Journées d'Informatique Musicales (JIM)*, La Londe, les Maures, France, 1998.

PERCEIVING AND PREDICTING EXPRESSIVE RHYTHM WITH RECURRENT NEURAL NETWORKS

Andrew J. Lambert
City University London
andrew.lambert.1@city.ac.uk

Tillman Weyde
City University London
t.e.weyde@city.ac.uk

Newton Armstrong
City University London
newton.armstrong.1@city.ac.uk

ABSTRACT

Automatically following rhythms by beat tracking is by no means a solved problem, especially when dealing with varying tempo and expressive timing.

This paper presents a connectionist machine learning approach to expressive rhythm prediction, based on cognitive and neurological models. We detail a multi-layered recurrent neural network combining two complementary network models as hidden layers within one system.

The first layer is a Gradient Frequency Neural Network (GFNN), a network of nonlinear oscillators which acts as an entraining and learning resonant filter to an audio signal. The GFNN resonances are used as inputs to a second layer, a Long Short-term Memory Recurrent Neural Network (LSTM). The LSTM learns the long-term temporal structures present in the GFNN's output, the metrical structure implicit within it. From these inferences, the LSTM predicts when the next rhythmic event is likely to occur.

We train the system on a dataset selected for its expressive timing qualities and evaluate the system on its ability to predict rhythmic events. We show that our GFNN-LSTM model performs as well as state-of-the art beat trackers and has the potential to be used in real-time interactive systems, following and generating expressive rhythmic structures.

1. INTRODUCTION

> "Composition is not a matter of filling or dividing time, but rather of generating time." [1]

The examination of the expressive qualities of music has been ongoing since the Ancient Greeks [2]. For instance, performers have been shown to express the higher metrical structures within a piece of music by tending to slow down at the end of certain phrases [3].

What Roads is alluding to in the above quote is that it is the perception of rhythmic events that provides a subjective experience of time to the listener. As the performer expressively varies the temporal dynamics, metrical dissonances and consonances are formed, affecting our perception of musical time and our expectation of rhythmical events. Our research concerns this interplay of metric *perception*, expectational *prediction* with respect to expressive variations on musical timing.

In order to achieve rhythmic prediction, we need to first overcome the current problem with perceiving expressive timing. Automatically processing an audio signal to determine pulse event onset times (beat tracking) is a mature field, but it is by no means a solved problem. Analysis of beat tracking failures has shown that a big problem for beat trackers is varying tempo and expressive timing [4, 5].

We take a cognitive approach, utilising a neurologically inspired model of rhythm perception known as a Gradient Frequency Neural Network (GFNN) [6]. In a GFNN a network of oscillators are distributed across a frequency spectrum. Internal connections between oscillators in the network can be learned via Hebbian learning. When stimulated by a signal, the GFNN resonates nonlinearly, producing larger amplitude responses at related frequencies along the spectrum. When the frequencies in a GFNN are distributed within a rhythmic range, resonances can occur at integer ratios to the pulse. These resonances can be interpreted as the perception of a hierarchical metrical structure.

GFNNs have shown promise even when dealing with more complex input, such as syncopated rhythms [7] and polyrhythms [8]. The oscillators' entrainment properties make them good candidates for solving the expressive timing problem and so the GFNN forms the basis of our *perception* layer.

In our system the GFNN is coupled with a Long Short-Term Memory Neural Network (LSTM) [9], which is a type of recurrent neural network able to learn long-term dependencies in a time-series. The LSTM takes the role of *prediction* in our system; it reads the GFNN's resonances to make predictions about the expected rhythmic events in the piece.

A future goal of our research is to use the GFNN-LSTM model for expressive rhythmic *production*. That is, the generation of new expressive timing structures based on its own output and/or other music agents' output. This system would be fast enough to operate in real-time.

In this paper, Section 2 details previous work in this area, Section 3 details a rhythm prediction experiment we have conducted with the GFNN-LSTM model and shares its results. Finally, Section 4 offers conclusions and points to future work.

Copyright: ©2015 Andrew J. Lambert 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.

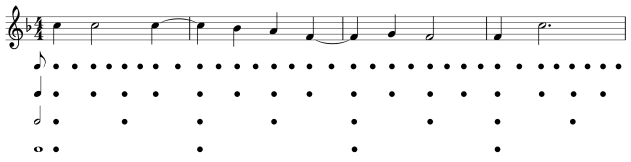


Figure 1. Metrical levels marked with Lerdahl and Jackendoff's 'dot notation'. The pulse level in this score would be at the crotchet (quarter note) level.

2. BACKGROUND

2.1 Pulse and Metre

A central idea in Lerdahl and Jackendoff's *Generative Theory of Tonal Music* (GTTM) is the notion of structures in music which are not present in the music itself, but perceived and constructed by the listener [10].

GTTM presents a detailed grammar of the inferred hierarchies a listener perceives when they listen to and understand a piece of music. Lerdahl and Jackendoff define four such hierarchies in tonal music, however in this paper we focus predominantly on *metrical structure*, considering other grammars only in relation to this.

A natural and often subconscious behaviour when we listen to music is that we tap our feet or nod our heads along to it. By doing so, we are reducing the music we hear into a series of periodic events. These events can sometimes be present in the music, but are often only implied by the rhythm of the music events and are constructed psychologically in the listener's mind. This process is known as *beat induction*; it is still an elusive psychological phenomenon that is under active research [11, 12], and has been claimed to be a fundamental musical trait [13].

When performing beat induction, one listener may tap along at twice the rate of another listener. In fact, there are several ways in which the music can be tapped along to, existing in a hierarchically layered relationship. The layers of beats are referred to in GTTM as 'metrical levels' and together they form a hierarchical metrical structure.

The beats at any given level can be perceived as 'strong' or 'weak'. If a beat on a particular level is perceived as strong, then it also appears in the next highest level, which creates the aforementioned hierarchy of beats. Theoretically, large measures, phrases, periods, and even higher order forms are possible in this hierarchy. Figure 1 illustrates a metrical analysis of a score.

Although tapping along at any metrical level is perfectly valid, humans often choose a common, comfortable period to tap to. Lerdahl and Jackendoff explain this selection process as a *preference rule* [14]. In general, this common period is referred to as the 'beat', but it is a problematic term since a beat can also refer to a singular rhythmic event or a metrically inferred event. Here we use a term that has recently grown in popularity in music theory: 'pulse' [15].

2.2 Nonlinear Resonance

GTTM is a musicological theory beginning with (but not limited to) the musical score as a source for analysing metre. What actually occurs in our brains as we listen to music and perform metre induction is another matter entirely.

Entrainment is the phenomena that occurs when two or more oscillations become synchronised in frequency and phase. It has been studied in a variety of disciplines such as mathematics and chemistry [16–18]. One can observe entrainment in action by placing several metronomes on a connected surface; over time the metronomes will synchronise [19].

Jones was among the first to propose an entrainment theory for the way we perceive, attend and memorise temporal events [20]. Jones posits that rhythmic patterns such as music potentially entrain a hierarchy of oscillations, forming an *attentional rhythm*. These attentional rhythms inform an expectation of when events are likely to occur, by extending the entrained period into the future.

Large takes this idea one step further with the notion of *nonlinear resonance* [6]. He states that musical structures occur at similar time scales to fundamental modes of brain dynamics, causing the nervous system to resonate to the rhythmic patterns. According to this theory, perceptions of pulse and metre perception arise as patterns of nervous system activity.

$$\frac{dz}{dt} = z(\alpha + i\omega + (\beta_1 + i\delta_1)|z|^2 + \frac{(\beta_2 + i\delta_2)\varepsilon|z|^4}{1-\varepsilon|z|^2}) + kP(\epsilon, x(t))A(\epsilon, \bar{z}) \quad (1)$$

Eq. (1) shows the differential equation that defines a Hopf normal form oscillator with its higher order terms fully expanded. This form is referred to as the canonical model, and was derived from a model of neural oscillation in excitatory and inhibitory neural populations [21]. z is a complex valued variable, $\bar{z}$ is its complex conjugate, and ω is the driving frequency in radians per second. α is a linear damping parameter, and β_1, β_2 are amplitude compressing parameters, which increase stability in the model. δ_1, δ_2 are frequency detuning parameters, and ε controls the amount on nonlinearity in the system. $x(t)$ is a time-varying external stimulus, which is also coupled nonlinearly and consists of passive part, $P(\varepsilon, x(t))$, and an active part, $A(\varepsilon, \bar{z})$, controlled by a coupling parameter k.

The α parameter acts as a bifurcation parameter: when $\alpha < 0$ the model behaves as a damped oscillator, and when $\alpha > 0$ the model oscillates spontaneously, obeying a limit-cycle. The gradual dampening of the amplitude allows the oscillator to maintain a long temporal memory of previous stimulation. This oscillator will resonate to an external stimulus that contains frequencies at integer ratio relationships to its natural frequency. Ratios such as 1:1, 2:1, 1:2, 3:1, 1:3, 3:2, and 2:3 are common and even higher order integer ratios are possible.

Optionally, canonical oscillators can be coupled together with a connectivity matrix as is shown in Eq. (2).

$$\frac{dz}{dt} = f(z, x(t)) + \sum_{i \neq j} c_{ij} \frac{z_j}{1-\sqrt{\epsilon} z_j} \cdot \frac{1}{1-\sqrt{\epsilon}\bar{z_i}} \quad (2)$$

Where $f(z, x(t))$ is the right hand side of Eq. (1) and c_{ji}

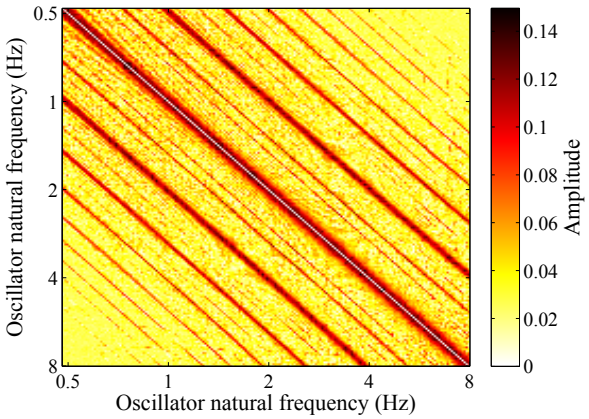


Figure 2. Amplitudes of connectivity matrix. Hebbian parameters are set to the following: $\lambda = .001, \mu_1 = -1, \mu_2 = -50, \epsilon_c = 16, \kappa = 1$, oscillator parameters are set to a limit cycle behaviour. Strong connections have formed at high-order integer ratios.

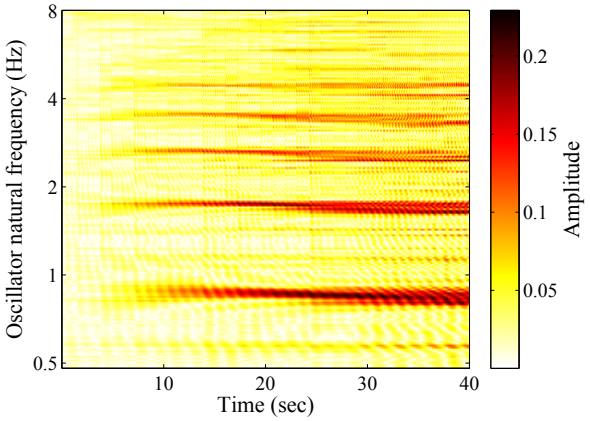


Figure 3. Amplitudes of oscillators over time.

is a complex number representing phase and magnitude of a connection between the ith and j^{th} oscillator.

Hebbian learning can be incorporated on these connections, in a similar way to Hoppensteadt and Izhikevich [22]. This can allow resonance relationships between oscillators to form stronger bonds and is shown in Eq. (3).

$$\frac{dc_{ij}}{dt} = c_{ij}(\lambda + \mu_1|c_{ij}|^2 + \frac{\epsilon_c\mu_2|c_{ij}|^4}{1-\epsilon_c|c_{ij}|^2}) + \kappa\frac{z_i}{1-\sqrt{\epsilon_c}z_i}\cdot\frac{z_j}{1-\sqrt{\epsilon_c}\bar{z_j}}\cdot\frac{1}{1-\sqrt{\epsilon_c}z_j} \quad (3)$$

Here λ, μ_1, μ_2, ϵ_c and κ are all canonical Hebbian learning parameters.

Figure 2 shows a connectivity matrix after Hebbian learning has taken place. In this example the oscillators have learned connections to one another in the absence of any stimulus due to the oscillators operating in their limit cycle behaviour. Connections have been learned at high order integer ratios.

2.3 Gradient Frequency Neural Networks

Connecting several canonical oscillators together with a connection matrix forms a *Gradient Frequency Neural Network* (GFNN) [21]. When the frequencies in a GFNN are distributed within a rhythmic range and stimulated with music, resonances can occur at integer ratios to the pulse.

Velasco and Large connected two GFNN networks together in a pulse detection experiment for syncopated rhythms [7]. The two networks were modelling the sensory and motor cortices respectively. In the first network, the oscillators were set to a bifucation point between damped and spontaneous oscillation ($\alpha = 0, \beta_1 = -1, \beta_2 = -0.25, \delta_1 = \delta_2 = 0$ and $\varepsilon = 1$). The second network was tuned to exhibit double limit cycle bifurcation behaviour ($\alpha = 0.3, \beta_1 = 1, \beta_2 = -1, \delta_1 = \delta_2 = 0$ and $\varepsilon = 1$), allowing for greater memory and threshold properties. The first network was stimulated by a rhythmic stimulus, and the second was driven by the first. The two networks were also internally connected in integer ratio relationships such as 1:3 and 1:2. The results showed that the predictions of the model match human performance, implying that the brain may be adding frequency information to a signal to infer pulse and metre. Other rhythmic studies with GFNNs include rhythm categorisation [23] and polyrhythmic analysis [8].

Figure 3 shows the amplitude response of a GFNN to a rhythmic stimulus over time. Darker areas represent stronger resonances, indicating that that frequency is relevant to the music. A hierarchical structure can be seen to emerge from around 8 seconds, in relation to the pulse which is just below 2Hz in this example. At around 24 seconds, a tempo change occurs, which can be seen by the changing resonances in the figure. These resonances can be interpreted as a perception of the hierarchical metrical structure.

2.4 Beat Tracking

By far the most common form of automatically predicting rhythmic events is that of automatically processing an audio signal to determine pulse event onset times. In Music Information Retrieval (MIR) this is known as *beat tracking*.

Automated beat tracking has a long history of research [24]. The MIR Evaluation eXchange (MIREX) [1] project runs a beat tracking task each year, which evaluates several submitted systems against various datasets. This provides an easy way to discern what the current state-of-the-art is in terms of beat tracking, which lately has been Böck and Schedl's system [25].

State-of-the-art beat trackers do a relativity good job of finding the pulse in music with a strong beat and a steady tempo, yet we are still far from matching the human level of beat induction. Furthermore, despite a recent surge in new beat-tracking systems, there has been little improvement over Klapuri et al.'s system [26].

Grosche et al. [4] have performed an in-depth analysis

[1] http://www.music-ir.org/mirex/

of beat tracking failures on the Chopin Mazurka dataset [2] (MAZ). MAZ is a collection of audio recordings comprising on average 50 performances of each of Chopin's Mazurkas. Grosche et al. found that properties such as expressive timing and ornamental flourishes were contributing to the beat trackers' failures.

Holzapfel et al. [5] have selected 'difficult' excerpts for a new beat tracking dataset by a selective sampling approach. This is now publicly available as the SMC dataset [3] . The SMC excerpts are tagged with a selection of signal property descriptors, which allows for an overview of what contributes to an excerpt's difficulty. Most of the descriptors refer to temporal aspects of the music, such as slow or varying tempo, ornamentation, and syncopation, and over half of the dataset is tagged with the most prominent tag: expressive timing.

From this it is clear that being able to track expressive timing variations in performed music is one area in which there is much room for improvement. This has been attempted in many cases, most notably in the work of Dixon [27] and Dixon and Goebl [28]. However, these systems do not perform well on today's standard datasets, scoring poorly on the SMC dataset in 2014's MIREX results.

2.5 Neural Network Music Models

Todd [29] and Mozer [30] were among the first to utilise a connectionist machine learning approach to music generation. One of the major advantages of this approach is that it replaces rule-based systems, which can be strict, lack novelty, and not deal with unexpected inputs very well. Instead, the structure of existing musical examples are learned by the network and generalisations are made from these learned structures to compose new pieces. Both Todd and Mozer's systems are recurrent networks that are trained to predict melody. They take as input the current musical context as a pitch class and note onset marker and predict the same parameters at the next time step.

Whilst Todd and Mozer were mainly concerned with predicting pitch sequences over time, Gasser et al. [31] have taken a connectionist approach to perceive and produce rhythms that conform to particular metres. Their neural network model *SONOR* is a self-organising network of adaptive oscillators that uses Hebbian learning to prefer patterns similar to those it has been exposed to in a learning phase. A single input/output (IO) node operates in two modes, perception and production. In the perception mode, the IO node is excited by patterns of strong and weak beats, conforming to a specific metre. Hebbian learning is used to create connections and between the oscillators in the network. Once these connections have been learned, the network can be switched to production mode, reproducing patterns that match the metre of the stimuli.

Recurrent neural networks (RNNs) such as the those used in the above systems can be good at learning temporal patterns. However, as noted by Todd [29] and Mozer [30], they often lack global coherence due to the lack of long-term memory. This results in sequences with good local structures, but long-term dependencies are often lost. One way of tackling this problem is to introduce a series of time lags into the network input, so that past values of the input are presented to the network along with the present. Kalos [32] used a model of a similar type to generate music data in symbolic MIDI format. One advantage of this method is that it performs well on polyphonic music, but the time lag method still does not capture long-term structure very successfully.

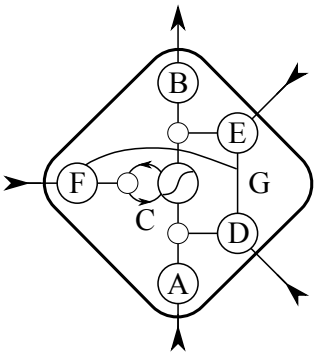


Figure 4. A single LSTM memory block showing (A) input, (B) output, (C) CEC, (D) input gate, (E) output gate, (F) forget gate and (G) peephole connections.

2.6 Long Short-Term Memory

Introduced by Hochreiter and Schmidhuber in 1997, Long Short-Term Memory Neural Networks (LSTMs) were designed to overcome the problem of modelling long term structures. Whilst RNNs can theoretically learn infinitely long patterns, in practice this is difficult due to the 'vanishing gradient problem' [9]. It can take as little as 5 time steps for this problem to occur in an RNN [33]. In an LSTM, a self-connected node known as the Constant Error Carousel (CEC) ensures constant error flow back through time, meaning that LSTMs can bridge time lags in excess of 1000 time steps [9].

A simplified diagram of an LSTM memory block can be seen in Figure 4. The input and output gates control how information flows into and out of the CEC, and the forget gate controls when the CEC is reset. The input, output and forget gates can be connected via 'peepholes'. For a full specification of the LSTM model we refer to [9] and [34].

As time-series predictors, LSTMs perform very well, as is shown by Böck and Schedl's beat tracker [25]. LSTMs have also had some success in generative systems. Eck and Schmidhuber [35] trained LSTMs which were able to improvise chord progressions in the blues and more recently Coca et al. [36] used LSTMs to generate melodies that fit within user specified parameters.

Lambert et al. have combined a GFNN with an LSTM (GFNN-LSTM) as two layers in an RNN chain and used it to predict melodies [37, 38]. Providing nonlinear resonance data from the GFNN helped to improve melody prediction with an LSTM. This is due to the LSTM being

[2] http://www.mazurka.org.uk/

[3] http://smc.inescporto.pt/research/data-2/

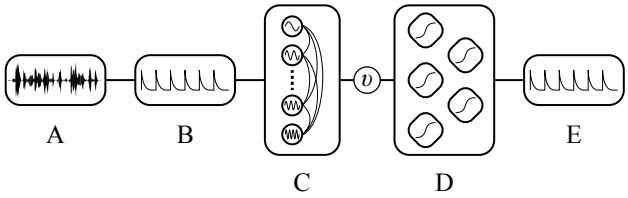


Figure 5. An overview of our GFNN-LSTM system showing (A) audio input, (B) mid-level representation, (C) GFNN, (D) LSTM, and (E) rhythm prediction output. The variable ν can be a mean field function or full connectivity.

able to make use of the relatively long temporal resonance in the GFNN output, and therefore model more coherent long-term structures. Here we take this work further by working with audio data and differing tempos.

3. EXPERIMENTS

We have performed an experiment where we have trained a GFNN-LSTM to predict expressive rhythmic events from audio data. The system takes audio data as input and outputs an event activation function. The system operates in a number of stages which are detailed below. A schematic of the system is provided in Figure 5.

The pieces in the MAZ dataset are expressively performed by various performers and vary in tempo and dynamics throughout the performance. However, the pieces are all within the same genre and are all performed on the piano, making drawing conclusions about the rhythmic aspects more valid. We have collected a subset of 50 excerpts, each 40 seconds long, by randomly choosing the full pieces and slicing 40 seconds worth of data.

When processing audio data for rhythmic events, it is common to first transform the audio signal into a more rhythmically meaningful form from which these events can be inferred. This representation could be extracted note onsets in binary form, or a continuous function that exhibits peaks at likely onset locations [39]. These functions are called *onset detection functions* and their outputs are known as *mid-level representations*.

Since we are dealing with expressively rich audio, we have chosen an onset detection function which is sensitive both to sharp and soft attack events such as those found in the MAZ piano performances. From Bello et al.'s tutorial on onset detection in music signals [40], we have selected the complex spectral difference onset detection function. This is a good general onset detector which works well with a variety of timbres. It is a continuous function that can be converted into binary onset data by using suitable threshold levels for peak picking. A sample rate of 86.025Hz was used, which was recently found to yield accurate detection results [41].

3.1 GFNN layer

The GFNN was implemented in MATLAB using the GrFNN Toolbox [42]. It consisted of 192 oscillators, logarithmically distributed with natural frequencies in a rhythmic range of 0.5Hz to 8Hz. The GFNN was stimulated by rhythmic time-series data in the form of the mid-level representation the audio data.

We have selected two parameter sets for the oscillators themselves, which affect the way the oscillators behave. The first is set to the bifurcation point between damped and spontaneous oscillation. We term this 'critical mode', as the oscillator resonates with input, but the amplitude decays over time in the absence of input: $\alpha = 0, \beta_1 = \beta_2 = -1, \delta_1 = \delta_2 = 0, \epsilon = 1$. By setting $\delta_1 = 1$, we define the second parameter set: 'detune mode'. These parameters allow the oscillator to change its natural frequency more freely, especially in response to strong stimuli. This essentially allows more entrainment to occur, so should allow for greater tracking of tempo changes. We obtained these values from the examples provided with the GrFNN Toolbox.

We have also selected three approaches to performing the Hebbian learning in the GFNN layer. The first approach simply has no connectivity between oscillators and therefore no learning activated at all (None). This is so that we can measure the effect (if any) that learning in the GFNN layer has on the overall predictions of the system.

The second approach is to activate online Hebbian learning with the following parameters: $\lambda = 0, \mu_1 = -1, \mu_2 = -50, \epsilon_c = 4$ and $\kappa = 1$ (Online). Under these parameters, the network should learn connections between related frequencies as they resonate to the stimulus.

The third approach is where generic initial connections have first been set in the network, learned by operating the oscillators in limit cycle mode (InitOnline). In this mode, the internal connections can be learned in the absence of any stimulus and results in a connectivity matrix shown in Figure 2. This provides a much more general state for the connection matrix to be in and potentially overcomes the limitations of the fixed frequency connections learned in online-only mode.

We found in some initial experimentation that during learning phase, the differential equations that drive the connectivity matrix can tend to spiral off to infinity. To ensure greater stability in the system, we have limited the connections in the connectivity matrix to have a magnitude less than $\frac{1}{\sqrt{\epsilon_c}}$ (0.5 in our experiments). We also and rescaled all stimuli to be in the range $0 <= x(t) <= 0.25$.

3.2 LSTM layer

The LSTM was implemented in Python using the PyBrain library [43]. For each variation of the GFNN, we trained two LSTM topologies. The first had 192 linear inputs, one for each oscillator in the GFNN, which took the real part of each oscillator's output. The second topology took only one linear input, which consisted of the mean field of the GFNN. The mean field reduces the dimensionality of the input whilst retaining frequency information within the signal.

All networks used the standard LSTM model with peephole connections enabled. The number of hidden LSTM blocks in the hidden layer was fixed at 10, with full recurrent connections. The number of blocks was chosen based on previous results which found it to provide reasonable

prediction accuracy, whilst minimising the computational complexity of the LSTM [38].

All networks had one single linear output, which serves as a rhythmic event predictor. The target data used was the output of the onset detection algorithm, where the samples were shifted so that the network was predicting what should happen next. The input and target data was normalised before training.

Training was done by backpropagation through time [44] using RProp- [45]. During training we used 5-fold cross-validation [46]. Training stopped when the total error had not improved for 20 epochs, or when this limit was reached, whichever came sooner.

3.3 Evaluation

The two main aims of this experiment were to firstly create a meaningful internal representation of metrical structure, and secondly to create good predictions in terms of the rhythmic structure. Therefore we are evaluating the system on its ability to predicted expressively timed rhythmic events, whilst varying the parameters of the GFNN and connectivity.

The results have been evaluated using the standard information retrieval metrics of precision, recall and F-measure. Events are predicted using a gradient threshold of the output data. The threshold looks for peaks in the signal by tracking gradient changes from positive to negative. When this gradient change occurs, an onset has taken place and is marked as such.

These events were subject to a tolerance window of $\pm$58.1ms. This means that an onset can occur within this time window and still be deemed a true positive. At the sample rate used in this experiment, this equates to 5 samples either side of an event. We also ensured that neither the target nor the output can have onsets faster than a rate of 16Hz, which is largely considered to be the limit of where rhythm starts to be perceived as pitch [6]. These are limitations to our evaluation method, but since we are mainly interested in predicted rhythmic structures and are not explicitly evaluating the production of expressive micro-timing, we believe they are acceptable concessions.

The first 5 seconds of output by the network are ignored, making the evaluation only on the final 35 seconds of predictions.

Table 1 and Table 2 display the results of the experiment, Figure 6 shows an example network output. These numerical metrics and visual figures provide some indication of how well the system is capturing the rhythmic structures. However, this information may be better understood by listening to the predicted rhythms. To this end, the reader is invited to visit this paper's accompanying website [4], where we have assembled a collection of audio examples and further output plots for each network's target and output data.

3.4 Discussion

We can see from the results that the best overall network incorporates detune oscillators, online learning with initial generic connections in the GFNN layer, and mean field connections.

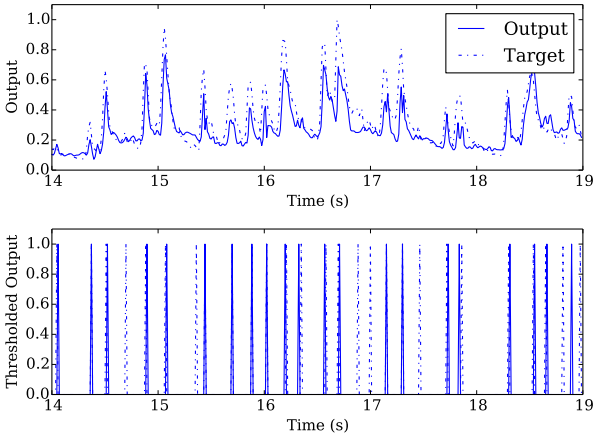


Figure 6. The output of the GFNN-LSTM. The top figure shown the predicted onset likelihood, the bottom figure displays the threshold events.

The mean field networks always outperformed the LSTMs with full connections to the GFNN. This is probably due to the mean field being able to capture the most resonant frequencies, whilst filtering out the noise of some less resonant frequencies. The resulting signal to the LSTM would therefore be more relevant for predicting rhythmic events. However, this may be due to the limited number of LSTM blocks in each network forming a bottleneck in the fully connected networks. Increasing number of hidden LSTM blocks may mitigate this limitation.

Another downside of the mean field networks is shown in the standard deviation figures. Whist performance improved in all cases using the mean field, the standard deviation also increased. This means there was a greater range of performances between the folds and could possibly indicate some networks being trained to local optima. During training we observed that the mean field networks took many more epochs for errors to converge.

The detuning oscillators outperformed the critical oscillators in all cases. This can be attributed to the greater amount of entrainment occurring in the network. Tempo changes can be tracked as an entrainment process between a local population of oscillators in the network. Where there is a local area of strong resonance the oscillators will take on very near frequencies to one another. As the stimulus frequency changes, this local area will be able to follow it, moving the local resonance area along the frequency gradient.

When compared to the results of our previous work on rhythm prediction with the GFNN-LSTM model [38], these results may at first seem a little underwhelming. The best network in our previous experiment achieved a rhythm prediction mean F-measure of 82.2%, compared with the 71.8% mean achieved here. However, this reflects the added difficulty of the task being undertaken here. Our previous work was on symbolic music at a fixed tempo and no expressive variation, whereas this study is undertaken on

[4] http://andyroid.co.uk/research/gfnn_lstm_rhythm_prediction/

Learning	LSTM	Precision	Recall	F-measure
None	Full	0.6114 (0.035)	0.6182 (0.034)	0.6059 (0.021)
None	Mean	0.6878 (0.100)	0.6883 (0.067)	0.6823 (0.081)
Online	Full	0.5637 (0.043)	0.6185 (0.076)	0.5798 (0.042)
Online	Mean	0.6862 (0.039)	0.6401 (0.050)	0.6548 (0.042)
InitOnline	Full	0.5982 (0.055)	0.6230 (0.041)	0.6000 (0.018)
InitOnline	Mean	**0.7032 (0.031)**	**0.6979 (0.041)**	**0.6958 (0.036)**

Table 1. Critical oscillation mode results. These results show the mean results calculated on the validation data. The number in brackets denotes the standard deviation.

Learning	LSTM	Precision	Recall	F-measure
None	Full	0.5972 (0.027)	0.6508 (0.036)	0.6161 (0.027)
None	Mean	0.7208 (0.058)	0.6891 (0.069)	0.6959 (0.057)
Online	Full	0.5831 (0.044)	0.6443 (0.067)	0.6020 (0.015)
Online	Mean	0.6943 (0.028)	0.6911 (0.045)	0.6866 (0.034)
InitOnline	Full	0.5666 (0.023)	0.6787 (0.033)	0.6114 (0.013)
InitOnline	Mean	**0.7239 (0.013)**	**0.7178 (0.061)**	**0.7142 (0.033)**

Table 2. Detune oscillation mode results. These results show the mean results calculated on the validation data. The number in brackets denotes the standard deviation.

audio data performed in expressive way at varying tempos. The overall best single system (Detune oscillators, InitOnline connections, and Mean input) was achieving an F-measure of 77.2%, which is extremely promising.

For comparison with other systems, the best beat tracker performance on MAZ submitted to MIREX in 2014 scored an F-measure of 71.5% (see [47]). Whilst this is not a direct comparison as we are predicting expressive rhythm, not pulse events, we believe it shows our system is at least comparable to state-of-the-art systems.

4. CONCLUSIONS

In this paper we have detailed a multi-layered recurrent neural network model for expressively timed rhythmic perception and prediction. The model consists of a perception layer, provided by a GFNN, and a prediction layer provided by an LSTM. We have evaluated the GFNN-LSTM on a dataset selected for its expressive timing qualities and found it to perform at a compatible standard to a previous experiment undertaken on symbolic data.

Our system's performance is comparable to state-of-the-art beat tracking systems. For the purposes of rhythm generation, we feel that the F-measure results reported here are already in a good range. Greater values may lead to too predictable and repetitive rhythms, lacking in the novelty expected in human expressive music. On the other hand, lower values may make the generated rhythms too random and irregular, so that they may even not be perceived as rhythmic at all. To make any firm conclusions on this, we would need to conduct formal listening tests based on the rhythms we have generated with our system. This is left for future work.

By using an oscillator network to track the metrical structure of expressively timed audio data, we have moved towards real-time processing of audio signals. We intend to extend this initial system for complete use as a MuMe system. Firstly, we will incorporate polyphonic rhythms into the system, instead of outputting a single rhythm output. Secondly, incorporating some melody model as in our previous work would be of use for complete autonomy of the system as a musical agent. Finally, we will close the feedback loop by connecting the system's output to its input. This would allow indefinite generation of new rhythmic structures which can be evaluated for their novelty. In doing so we will have created an expressive, generative, real-time agent.

Acknowledgments

Andrew J. Lambert is supported by a PhD studentship from City University London.

5. REFERENCES

[1] C. Roads, "Rhythmic Processes in Electronic Music," in *Joint 40th International Computer Music Conference and 11th Sound & Music Computing conference*, Athens, Greece, 2014.

[2] A. Gabrielsson and E. Lindström, "The role of structure in the musical expression of emotions," *Handbook of music and emotion: Theory, research, applications*, pp. 367–400, 2010.

[3] E. F. Clarke, "Generative principles in music performance." 1988.

[4] P. Grosche, M. Müller, and C. S. Sapp, "What Makes Beat Tracking Difficult? A Case Study on Chopin Mazurkas," in *in Proceedings of the 11th International Society for Music Information Retrieval Conference, 2010*, 2010, pp. 649–654.

[5] A. Holzapfel, M. E. P. Davies, J. Zapata, J. Oliveira, and F. Gouyon, "Selective Sampling for Beat Tracking Evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2539–2548, Nov. 2012.

[6] E. W. Large, "Neurodynamics of Music," in *Music Perception*, ser. Springer Handbook of Auditory Research, M. R. Jones, R. R. Fay, and A. N. Popper, Eds. Springer New York, Jan. 2010, no. 36, pp. 201–231. [Online]. Available: http://0-link.springer.com.wam.city.ac.uk/chapter/10.1007/978-1-4419-6114-3_7

[7] M. J. Velasco and E. W. Large, "Pulse Detection in Syncopated Rhythms using Neural Oscillators," in *12th International Society for Music Information Retrieval Conference*, Miami, FL, 2011, pp. 185–190.

[8] V. Angelis, S. Holland, P. J. Upton, and M. Clayton, "Testing a Computational Model of Rhythm Perception Using Polyrhythmic Stimuli," *Journal of New Music Research*, vol. 42, no. 1, pp. 47–60, 2013. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/09298215.2012.718791

[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[10] F. Lerdahl and R. Jackendoff, *A generative theory of tonal music*. Cambridge, Mass.: MIT press, 1983.

[11] G. Madison, "An Auditory Illusion of Infinite Tempo Change Based on Multiple Temporal Levels," *PLoS ONE*, vol. 4, no. 12, p. e8151, Dec. 2009. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0008151

[12] J. London, *Hearing in Time: Psychological Aspects of Musical Meter*. Oxford University Press, May 2012. [Online]. Available: http://0-www.oxfordscholarship.com.wam.city.ac.uk/view/10.1093/acprof:oso/9780199744374.001.0001/acprof-9780199744374

[13] H. Honing, "Without it no music: beat induction as a fundamental musical trait," *Annals of the New York Academy of Sciences*, vol. 1252, no. 1, pp. 85–91, 2012.

[14] F. Lerdahl and R. Jackendoff, "An Overview of Hierarchical Structure in Music," *Music Perception: An Interdisciplinary Journal*, vol. 1, no. 2, pp. 229–252, Dec. 1983. [Online]. Available: http://www.jstor.org/stable/40285257

[15] S. Grondin, *Psychology of Time*. Emerald Group Publishing, 2008.

[16] C. Huygens, *Horologium oscillatorium, sive de motu Pendulorum ad Horologia aptato demonstrationes geometricae*. Muguet, 1673.

[17] Y. Kuramoto, *Chemical oscillations, waves and turbulence*. Springer, Berlin, 1984.

[18] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology and chemistry*. Perseus publishing, 2001.

[19] J. Pantaleone, "Synchronization of metronomes," *American Journal of Physics*, vol. 70, no. 10, pp. 992–1000, Oct. 2002. [Online]. Available: http://scitation.aip.org/content/aapt/journal/ajp/70/10/10.1119/1.1501118

[20] M. R. Jones, "Time, our lost dimension: Toward a new theory of perception, attention, and memory," *Psychological Review*, vol. 83, no. 5, pp. 323–355, Sep. 1976. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1977-07367-001&site=ehost-live

[21] E. W. Large, F. V. Almonte, and M. J. Velasco, "A canonical model for gradient frequency neural networks," *Physica D: Nonlinear Phenomena*, vol. 239, no. 12, pp. 905–911, Jun. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167278910000187

[22] F. C. Hoppensteadt and E. M. Izhikevich, "Synaptic organizations and dynamical properties of weakly connected neural oscillators II. Learning phase information," *Biological Cybernetics*, vol. 75, no. 2, pp. 129–135, 1996.

[23] R. Bååth, E. Lagerstedt, and P. Gärdenfors, "An Oscillator Model of Categorical Rhythm Perception," in *Proceedings of the 35th Annual Conference of the Cognitive Science Society*, M. Knauff, M. Pauen, N. Sebanz, and I. Wachsmuth, Eds. Austin, TX: Cognitive Science Society, 2013, pp. 1803–1808.

[24] P. E. Allen and R. B. Dannenberg, "Tracking musical beats in real time," in *Proceedings of the 1990 International Computer Music Conference*, vol. 140143, 1990.

[25] S. Böck and M. Schedl, "Enhanced Beat Tracking with Context-Aware Neural Networks," in *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, Paris, France, 2011.

[26] A. Klapuri, A. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 342–355, Jan. 2006.

[27] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, 2001.

[28] S. Dixon and W. Goebl, "Pinpointing the beat: Tapping to expressive performances," in *Proc. of International Conference on Music Perception and Cognition*, Sydney, Austrailia, 2002, pp. 617–620.

[29] P. M. Todd, "A Connectionist Approach to Algorithmic Composition," *Computer Music Journal*, vol. 13, no. 4, pp. 27–43, Dec. 1989. [Online]. Available: http://www.jstor.org/stable/3679551

[30] M. C. Mozer, "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing," *Connection Science*, vol. 6, no. 2-3, pp. 247–280, 1994.

[31] M. Gasser, D. Eck, and R. Port, "Meter as Mechanism: A Neural Network Model that Learns Metrical Patterns," *Connection Science*, vol. 11, no. 2, pp. 187–216, 1999. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/095400999116331

[32] A. Kalos, "Modeling MIDI Music as Multivariate Time Series," in *IEEE Congress on Evolutionary Computation, 2006. CEC 2006*, 2006, pp. 2058–2064.

[33] F. Gers and J. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, Nov. 2001.

[34] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000. [Online]. Available: http://dx.doi.org/10.1162/089976600300015015

[35] D. Eck and J. Schmidhuber, "Finding temporal structure in music: blues improvisation with LSTM recurrent networks," in *Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing, 2002*, 2002, pp. 747–756.

[36] A. Coca, D. Correa, and L. Zhao, "Computer-aided music composition with LSTM neural network and chaotic inspiration," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug. 2013, pp. 1–7.

[37] A. Lambert, T. Weyde, and N. Armstrong, "Beyond the Beat: Towards Metre, Rhythm and Melody Modelling with Hybrid Oscillator Networks," in *Joint 40th International Computer Music Conference and 11th Sound & Music Computing conference*, Athens, Greece, 2014.

[38] ——, "Studying the Effect of Metre Perception on Rhythm and Melody Modelling with LSTMs," in *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.

[39] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *The Journal of the Acoustical Society of America*, vol. 103, no. 1, pp. 588–601, Jan. 1998. [Online]. Available: http://scitation.aip.org/content/asa/journal/jasa/103/1/10.1121/1.421129

[40] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 1035–1047, 2005.

[41] M. E. P. Davies and M. Plumbley, "Context-Dependent Beat Tracking of Musical Audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1009–1020, Mar. 2007.

[42] E. W. Large, J. C. Kim, K. L. Lerud, and D. Harrell, "GrFNN Toolbox 1.0: Matlab tools for simulating signal processing, plasticity and pattern formation in gradient frequency neural networks," 2014. [Online]. Available: https://github.com/GrFNN/Toolbox-1.0

[43] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber, "PyBrain," *Journal of Machine Learning Research*, vol. 11, pp. 743–746, 2010.

[44] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[45] C. Igel and M. Hüsken, "Improving the Rprop learning algorithm," in *Proceedings of the second international ICSC symposium on neural computation (NC 2000)*. Citeseer, 2000, pp. 115–121.

[46] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI*, vol. 14, 1995, pp. 1137–1145.

[47] S. Böck, F. Krebs, and G. Widmer, "A multi-model approach to beat tracking considering heterogeneous music styles," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014), Taipei, Taiwan*, 2014.

Grammatical Evolution with Zipf's Law Based Fitness for Melodic Composition

Róisín Loughran
NCRA, UCD CASL,
Belfield, Dublin 4
roisin.loughran@ucd.ie

James McDermott
NCRA, UCD CASL,
Belfield, Dublin 4
jmmcd@jmmcd.net

Michael O'Neill
NCRA, UCD CASL,
Belfield, Dublin 4
m.oneill@ucd.ie

ABSTRACT

We present a novel method of composing piano pieces with Grammatical Evolution. A grammar is designed to define a search space for melodies consisting of notes, chords, turns and arpeggios. This space is searched using a fitness function based on the calculation of the Zipf's distribution of a number of pitch and duration attributes within the given melodies. In this way, we can create melodies without specifying a key or time signature. We can then create simple accompanying bass parts to repeat under the melody. This bass part is evolved using a grammar created from the evolved treble line with a fitness based on Zipf's distribution of the harmonic relationship between the treble and bass parts. From an analysis of the system we conclude that the designed grammar and the construction of the compositions from the final population of melodies is more influential on the musicality of the resultant compositions than the use of the Zipf's metrics.

1. INTRODUCTION

Music composition is a complex, aesthetic process. In recent years many composers, musicologists and computer scientists have looked to machine learning, autonomous methods of creating music either in conjunction with, or instead of the traditional human composer. We present one such study in which we employ an Evolutionary Computation (EC) method, namely Grammatical Evolution (GE) in the composition of piano pieces.

GE [1, 2] offers a versatile way of accessing and searching through a problem while taking advantage of problem domain knowledge. GE has been shown to be effective at a wide range of creative tasks including pylon and truss design, navigation in computer games and graphical logo design [3–6]. EC methods in general are not deterministic; a solution is rarely determined outright but rather approached from a number of locations. This makes them particularly suitable to aesthetic problems such as musical composition — composition is not a linear, deterministic process, but a combination of decisions that, once started, would be unlikely to end up in the same position twice.

Copyright: ©2015 Róisín Loughran 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.

This paper discusses the representations, grammars and fitness functions that we use to employ GE as an autonomous composer of piano pieces.

The following section details some previous experiments in using EC methods to compose music. Section 3 introduces Grammatical Evolution and gives a background the Zipf's Law power distribution used throughout this study. Section 4 details the workings of the experiment: the grammar used, the fitness measured and the manner in which we create an accompanying bass part. Section 5 presents and discusses and number of the melodies created by the system. Some conclusions and future work are proposed in Section 6.

2. PREVIOUS WORK

A number of previous studies have employed EC techniques for melodic composition. One of the most successful and well-known applications is GenJam [7] which uses a Genetic Algorithm (GA) to evolve jazz solos. This system has been modified and developed into a real-time, MIDI-based, interactive improvisation system that is regularly used in live performances in mainstream venues [8]. A modified GA was used in GeNotator [9] to manipulate a musical composition using a hierarchical grammar. Göksu et al. evolved and evaluated both melody and rhythm separately using MLPs [10]. These evolved melodies were then mixed to produce verses and whole songs. Dahlstedt developed a system that implements recursively described binary trees as genetic representation for the evolution of musical scores. The recursive mechanism of this representation allowed the generation of expressive performances and gestures along with musical notation [11].

GE was first used for musical composition by de la Puente et al [12]. They tested the use of GE to generate melodies for a specific processor but did not present or discuss the melodies produced. More recently GE has been implemented for composing short melodies in [13]. From four experimental setups of varying fitness functions and grammars they determined that users preferred melodies created with a structured grammar. GE was again employed for musical composition using the Wii remote for a generative, virtual system entitled Jive [14]. This system interactively modifies a combination of sequences to create melodic pieces of musical interest.

Most of the above methods employ Interactive EC (IEC) methods, whereby a human observer is used within the fitness function. While a human observer is ideal for making subjective judgments on aesthetic processes such as art

and music, IEC is extremely costly. In the proposed experiments we avoid IEC and instead opt for an autonomous evaluation of the individuals based on Zipf's Laws.

Zipf's Law has been used in the investigation of pleasantness in music [15] and has been used previously as a fitness function in EC [16]. Zipf's Law relates to the frequency of occurrence of events and has been shown to turn up in many aspects of nature [17]. Formally, Zipf's Law states:

$$P(f) \sim 1/f^n \quad (1)$$

where P(f) is the probability of an event whose ranked frequency of occurrence is f and where n is close to 1. The number of occurrences are noted for each type of event. These occurrences are plotted against their statistical rank on a log-log scale. For an ideal Zipf's distribution we expect all points to fall on a straight line with a slope of -1. The R^2 value is a measure of how much the given points conform to this line, ranging from 0 to 1 with 1 denoting a straight (ideal) line. In order to calculate the Zipf's fitness for an attribute within a given individual (melody), we calculate the slope and R^2 of the rank-frequency distribution of this attribute and compare it to these ideal values.

The contribution of this study to the field of algorithmic composition lies in the exploitation of GE's capabilities to use grammars in representing and manipulating musical phrases. We use the population aspect of GE to combine multiple highly fit individuals together. We then use a two-run process where the first run evolved a treble melody, a new grammar is dynamically created in response to it, and then the second run uses this grammar to evolve an accompanying bass line. At the end of the study, we examine the resultant melodies in relation to each of the aspects used in creating them.

3. GRAMMATICAL EVOLUTION

GE is a grammar based algorithm based on Darwin's theory of evolution. As with other evolutionary algorithms, the benefit of GE as a search process results from it's operation on a population of solutions rather than a single solution. From an initial population of random genotypes, GE performs a series of operations such as selection, mutation and crossover over a number of generations to search for the optimal solution to a given problem. A grammar is used to map each genotype to a phenotype that can represent the problem under investigation. The success or 'fitness' of each individual can then be assessed as a measure of how well this phenotype solves the problem. Successful or highly fit individuals reproduce and survive to successive generations while weaker individuals can be weaned out. Such grammar-based generative methods can be particularly suitable to generating music as it is a genome that is being manipulated rather than the piece of music itself. This allows the method to generate an output with a level of complexity far greater than the original input. This added complexity generation is helpful in creating interesting and diverse pieces of music. In the experiments proposed in this paper, the grammar defines the search domain — the allowed notes and musical events in each composition. Successful melodies are then chosen by traversing this search space according to the defined fitness function.

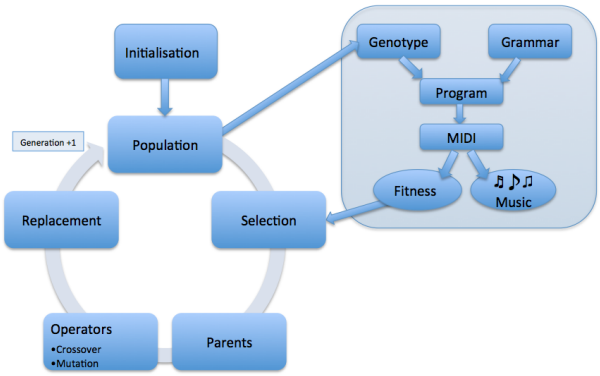


Figure 1: Overview of GE Process.

The creative capabilities of GE come from the choices offered within the mapping of the grammar. The grammars in GE are used to map the genotype to the phenotype and are often in Backus-Naur Form (BNF). Typically, the genome is represented by a combination of integers known as *codons*. These codons select the particular rule for a given expression according to the mod value from the number of choices for that rule.

$$\text{Rule} = (\text{Codon Integer Value}) mod (\#\text{ of choices}) \quad (2)$$

Using this we can introduce biases to our grammar by including multiple instances for preferred choices. For example, operand, depicted in Equation 3 offers three choices, two of which are choice1. Thus there is a 2:1 bias towards the selection of choice1 over choice2. We make use of such biases in our experiments to incorporate our knowledge of the musical domain into the designed grammar.

$$<\text{operand}>:: = <\text{choice1}> \mid <\text{choice1}> \mid <\text{choice2}> \quad (3)$$

We exploit the representational capabilities of GE resulting from the design of a grammar that defines the given search domain. GE maps the genotype to a phenotype — typically some form of program code. This phenotype can then be interpreted by the user in a predetermined manner. In these experiments, the programs created are written in a command language based on integer strings to represent sequences of MIDI notes. We design a grammar to create this command language which is in turn used to play music. An overview of the GE process including the mapping of the grammar to MIDI notes is shown in Figure 1.

4. METHOD

This section describes the methods used in composing the piano melodies that accompany this paper.

4.1 Grammar

The BNF grammar, shown below, maps the genotype (integer codon) to a series of musical events entitled notes, chords, runs, turns and arpeggios to create a musical representation. These events are re-written to numerical values that comprise a command language (series of integers) that is interpreted as individual MIDI notes.

```
<piece>::=<event>|<piece><event>
 |<piece><event><event>
 |<piece><event><event><event>
<event>::=111,<style>,<oct>,<pitch>,<dur>,

<style>::=100|100|100|100|100|100|100|100
 |50,<chord>|50,<chord>|50,<chord>
 |50,<chord>|70,<turn>,100 | 80,<arp>,100
<chord>::=<int>,0,0|<int>,<int>,0
 |12,0,0|<int>,0,0|<int>,0,0|<int>,0,0
 |<int>,<int>,<int>
<turn>::=<dir>,<len>,<dir>,<len>,<stepD>
<arp>::=<dir>,<int>,<dir>,<int>,<ArpDur>

<int>::=3|4|5|7|5|5|7|7
<len>::=<step>|<step>,<step>
 |<step>,<step>,<step>
 |<step>,<step>,<step>,<step>
 |<step>,<step>,<step>
<dir>::=45|55
<step>::=1|1|1|1|1|2|2|2|2|2|2|2|2|3
<stepD>::=1|2|2|2|2|2|2|4|4|4|4|4|4
<ArpDur>::=2|2|2|4|4|4|4|4|8|8
<oct>::=3|4|4|4|4|5|5|5|5|6|6
<pitch>::=0|1|2|3|4|5|6|7|8|9|10|11
<dur>::=1|1|1|2|2|2|4|4|4|8|8|16|16|32
```

The first line creates a melody <piece> from either a single note <event> or a concatenation of note events. The inclusion of extra <event> in this first line encourages expansion of the phenotype. Each <event> starts with the indicator 111 and has the descriptors <style>, <oct>, <pitch> and <dur>. Each descriptor is mapped by the grammar in relation to what it represents. Pitch is simply a value between 0 and 11 chosen with equal probability to indicate which of the 12 pitches in the chromatic scale. Octave refers to the octave number the current event starts in and is limited to 3-6 for these experiments with a bias towards 4 and 5. Each note is assigned a specific duration ranging from a demisemiquaver (value 1) to a semibreve (value 32). As with the octave descriptor, a bias is introduced to encourage shorter notes within the melodies with notes shorter than a quaver (value 4) given more instances and hence higher preference over longer minim and semibreve notes.

The type of event determined by <style> can be a plain note denoted by 100, a chord (50), a turn (70) or an arpeggio (80). This grammar has a strong bias towards including more notes and chords as they take less time to play but can be more pivotal to the overall piece than turns. A plain note requires no further information than the octave, pitch and duration already assigned to it and so requires no further grammar. A chord (50) is defined by the pitch and duration already specified and the inclusion of either one, two or three notes played in conjunction with it.

Both turn (70) and arp (80) result in a series of notes played in sequence. The direction up or down is chosen at the beginning and again halfway through the turn. As the second choice of direction is independent from the first, this grammar will produce a run (both directions the same) 50% of the time, resulting in no need for a separate grammar line for runs. The length of each section of the turn is one, two, three or four notes with a bias towards three. Each step size within the turn is either one or two semitones, with the occasional allowance of three. The duration of the step is limited to either semiquavers or quavers. An arpeggio is created in a similar manner.

Table 1: Attributes measured from a given individual

Name	Description
Pitch	pitch class (value 1-12)
Dur	duration
Pitch-Dur	pitch*duration
Pitch-Dist	distance between instances of a given pitch
Pitch Int	pitch interval from each note to the next
Pitch Bigram	pitch distance between successive intervals

4.2 Fitness Function

Once the grammar has mapped to the phenotype, the fitness function is called to evaluate the given individual according to a defined fitness measure. We give each individual an initial fitness based on the duration of the melody produced. We aim for a melody of duration of 300 but with a tolerance of 30. If the duration is within this tolerance the initial fitness is set to 1, else the initial fitness is calculated as the absolute value of the difference from the duration to 300 and the tolerance, plus 1. The addition of the constant 1 is to prevent a fitness of zero as this initial fitness is now adjusted by multiplication according to the Zipf's distribution of a number of attributes.

The final fitness of the individual is measured in relation to the distance in vector space of the Zipf's distribution of each of the measures shown in Table 1. These particular attributes, a subset of those used in previous experiments [15, 16], were chosen as they are most suited to the representation and methods used in this study. Measures related to the absolute pitch value were not incorporated as the grammar already controls a bias towards the use of certain octaves. Hence the term 'pitch' in these measures relates merely to the pitch class (value 1-12). Likewise we do not consider the fractal measures used in previous studies as the original duration of the pieces in these experiments is so short.

4.3 Melody Construction

The above grammar and fitness measurement create very short melodies. In order to create longer compositions we concatenate fit individuals from the final generation together. We can implement this by exploiting the fact that GE produces a population of fit individuals. In the final generation a number of the most fit individuals should be quite similar as they share common highly fit traits. Thus if we play the best individuals together we expect similar melody snippets or motifs to emerge. Previous studies in using EC for algorithmic composition have used the entire population and generations of populations in creating a single melody [18,19]. Due to the large diversity within our final population, discussed in the next section, we only con-

sider a small number of top individuals for inclusion in the final composition. A number of melodies accompany this paper displaying varying degrees of repetition and variation on a theme. Each of these longer melodies were created by concatenating the four top individuals from the final generation together.

4.4 Bass Accompaniment

Conventional piano music generally consists of two separate parts, treble and bass. Thus as an extra experiment we use a two-stage GE run that uses the best individual from the treble run to create a new grammar to compose an accompanying bass line.

Although no tonality has been enforced on the melody, the Zipf's metrics used will cause certain pitches to be played more frequently than others. Thus without predefining a key signature we can encourage a bass accompaniment to sound tonally similar to this melody by ensuring the bass exhibits the same pitch biases as the treble. We can control this effectively using our GE composition system by creating a new grammar for the bass which is derived from the evolved treble line. In this way we can ensure that only pitches already used within the piece will be considered when composing an accompaniment.

We create the grammar file for the bass part once the best individual for the treble has been found. This grammar file is created with initial predetermined lines to specify allowed note duration and octave. Only plain notes and chords are allowed in the bass grammar. The line of the grammar that defines the allowed pitches is created from an ordered list of pitches in the treble line. From this we create a list of pitches available to the bass that includes the top notes from the melody four times, the next two notes three times, the following two notes twice and includes the sixth, seventh and eight most frequent note once. This creates a bias within the bass towards the pitches most frequently used within the melody. For example if the most frequent pitches in the treble melody were A, C, F#, G, D, E, D# and C#, in decreasing frequency, the line:

```
<pitch>::=9|9|9|9|0|0|0|6|6|6|7|7|2|2|4|3|1
```

would be added to the predefined grammar, completing the grammar for the GE to evolve the bass accompaniment.

The bass grammar considers the tonality of the treble and bass parts but it does not take into account the progression or timing between the two. A simple method to create an accompanying line is to create one bass part that repeats underneath all four similar melodies. To achieve this we must be more strict in the duration of the treble melody evolved; if we want the accompanying bass to repeat twice under each melody individual we must ensure each bass individual is exactly half the duration of the treble. Thus we re-run the experiment again with a target duration of 128 for the treble, 64 for the bass and zero tolerance for both parts. As a duration of 1 represents a demisemiquaver, a duration of 32 could represent one bar in 4/4 time. Hence we can consider the melody to be of length four bars and the bass to be of length two, although the duration of individual bar lengths is not enforced.

To measure the fitness of the bass individual we again consider a Zipf's distribution, but this time on the harmonic relationship between the pitches in this bass and the melody it is accompanying. To consider the harmonic progression between the two parts we must examine the relationship between each pair of notes at every time-step. To examine this we expand out the pitch line for both the treble and bass so that we have a value at each instance (each moment of duration 1). For example, if there is a crotchet (duration 8) played at D (pitch 2) we represent this with a list of 8 values of 2. This results in two lists, one for treble and one for bass that indicate the pitch of each line at every moment of duration. In the case of a chord, only the root note of the chord is considered. We then subtract the bass from the treble list to create a list of intervals. As we are only considering pitch values within an octave, this results in a negative value should the pitch value of the bass be higher than that of the treble. We correct this by adding a value of 12 when this occurs.

We then categorise the resultant interval list into a list of rankings according to standard Western tonality. These rankings indicate how consonant or dissonant an interval is, with 0 being the most consonant (least dissonant) and 12 being the most dissonant (least consonant). From this list of rankings (which is already sorted), we can then enforce a Zipf's distribution and adjust the fitness in accordance to the deviation from this distribution as per the treble part.

The system described is implemented in python using PonyGE `https://code.google.com/p/ponyge/`. Details of the experiments run are given in the following section.

5. COMPOSITIONS

The experiments were run with a population of 200 for 50 generations. All other parameters were left to the default settings in PonyGE: the mutation coefficient was set to 0.01, crossover was set to 0.7 and there was an elite size of 1. Each experiment was run with a minimising fitness function whereby zero is the ideal fitness.

A selection of melodies created by this system are available at `http://ncra.ucd.ie/Site/loughranr/smc_2015.html`. In this section we discuss the creation of the melodies in relation to fitness evolution, the Zipf's distributions, variety in the final generation and the creation of an accompanying bass part.

5.1 Short Melodies

5.1.1 Fitness Evolution

The progress of any evolutionary run is best examined by observing the progress of the average and best solution in successive generations across multiple runs. Figure 2 displays the average versus best fitness across 30 evolutionary runs for the creation of the melody line. We note that the fitness is calculated directly but the natural log is shown for illustrative purposes. It is evident from this graph that on average a near optimal fitness can be found after about 30 generations. In contrast to this, the average fitness remains

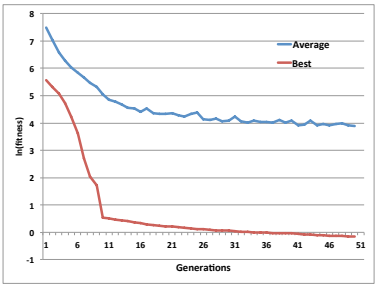


Figure 2: Average vs. Best Fitness over 50 generations average over 30 runs.

Table 2: Zipfs measurements from individual melody attributes. Numbers in parenthesis indicate the ideal values.

Attribute	Slope (-1)	R^2(1)	Fit (0)
Pitch	-1.01	0.94	0.07
Duration	-1.1	0.98	0.13
PitchDur	-1.0	0.95	0.06
Interval	-1.0	0.93	0.07
PitchDist	-1.0	0.91	0.08
Bigram	-1.02	0.95	0.08

quite high. This implies that after 50 generations the population is still very diverse. We consider the reason for this diversity later, but first we examine an individual melody in terms of the fitness attributes measured.

ShortMelody is the best evolved individual across all runs with a final fitness of 0.49. This melody contains all melodic events the grammar is capable of producing — single notes, chords, turns, runs and arpeggios. The evolution of each of the attributes is shown in Figure 3. These plots show the progression of the slope and R^2 for each of the six measured attributes for the best and median individual in each generation, measured by fitness. As expected the best value approaches the ideal for each value quickly whereas the median values show much more variation. It should be noted that the median value at generation 1 tends to be zero. This is because the first generation have many very weak individuals (more than half the population) that are very short resulting that the median's initial fitness is too weak to be adjusted using Zipf measurements. This variety with the median values across the generations show that while the best fitness is easily met, the population remains diverse in relation to each of the fitness attributes.

5.1.2 Zipf's Distribution

Figure 4 displays the distribution of each of the six attributes measured from ShortMelody in relation to the Zipf's ideal. These plots clearly illustrate that the points converge to a straight line with a negative slope. For a closer inspection Table 2 displays the the specific values from these plots for slope, R^2 and attribute fitness. Despite the small number of points on these graphs, they do indicate that the attributes display Zipf-like distribution with each of their slopes approaching the ideal of -1 and R^2 approximating the ideal of 1.

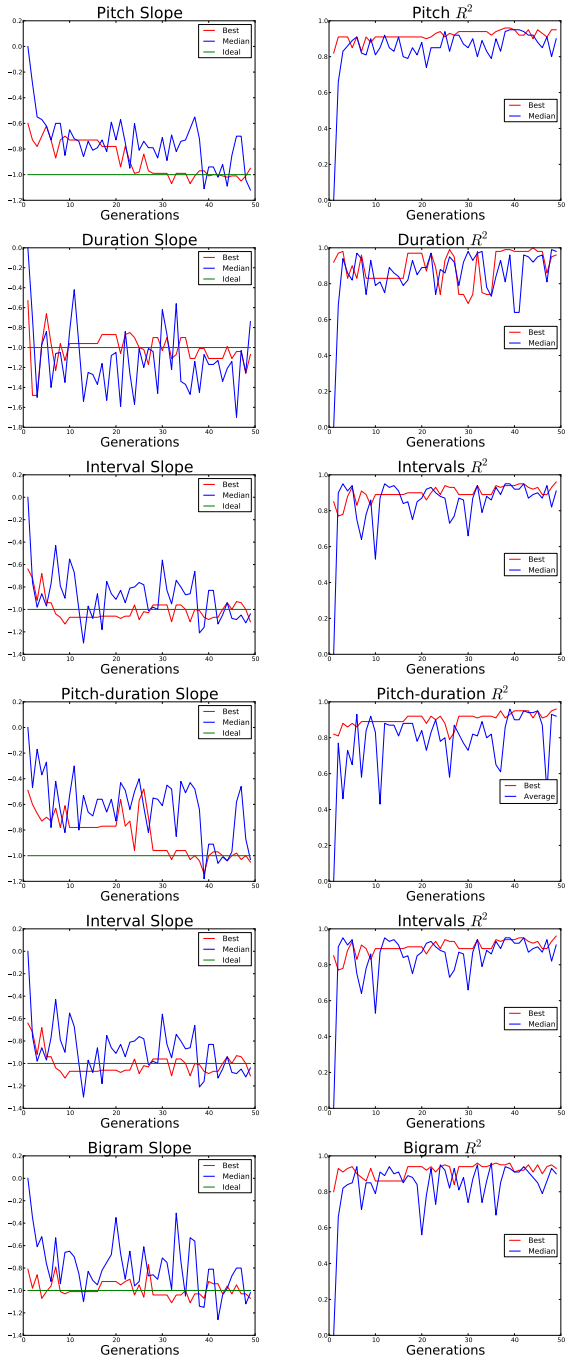


Figure 3: Evolution of attributes for most fit melody

5.1.3 Final Generation

Figure 2 shows that after 50 generations there is still a large difference between the average and best fitness in a population, indicating that the final population is still very diverse. To determine why this may be, we examine the individuals within the final population of an evolutionary run. Figure 5 shows the fitness values within the final population. These show that over 50% of the population do have very low (good) fitness. The distance between the average and the best is caused by a small number of very weak individuals that drag the average up. Examining the lengths in the final population indicate that a similar number of individuals have very short durations. As the initial fitness is

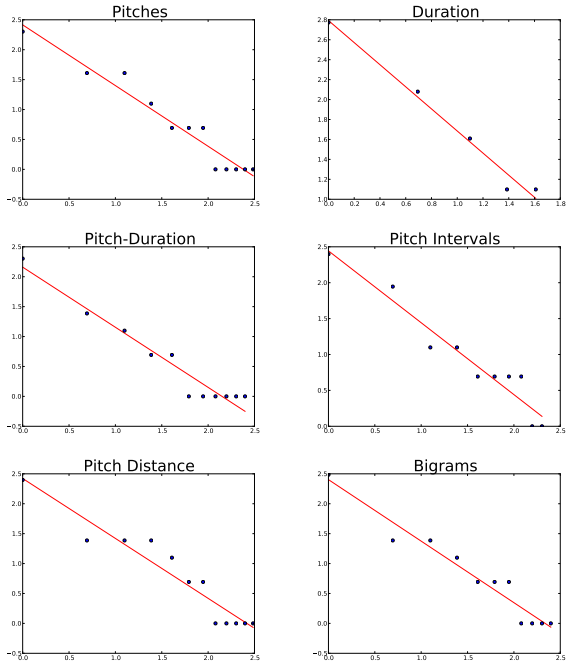


Figure 4: Attribute distributions for most fit melody. The red line indicates the Zipf's ideal distributions.

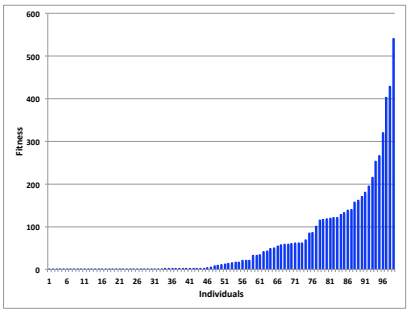


Figure 5: Overall Fitness Values in final generation.

based on length, a short duration will dramatically increase the fitness, thus a small number of short melodies will alter the average fitness of that generation. Similarly, we can examine individual attribute fitness measures within the final generation. Again we find that the attributes approach ideal values for those individuals with low fitness but deviate from the ideal in weaker individuals in the population. The attribute that shows most deviation is duration. This is unsurprising as the addition or removal of a single turn can significantly alter the instances of a given duration. Even so, Figure 5 clearly shows that there are a large number of melodies with a very good fitness, hence we can be confident in our choice of the single best or make use of a combination of the best as described in the following section.

5.2 Composite Melodies

Eight composite melodies accompany this paper displaying varying degrees of repetition and variation on a theme. Each of these longer melodies were created by concatenating the four top individuals from the final generation together. Melody4.mp3 offers an interesting motif emerging within the middle of each individual. This motif is notated in Figure 6. Similar themes can be heard to emerge in the other melodies. These motifs or themes ground the compositions giving them a sense of oneness and modularity. The emergent themes can vary in length; Melody6.mp3 can be heard to root itself in a long F# both at the middle and end of each individual giving a very repetitive flow to the melody. Each melody presented displays the events produced by the grammar in terms of runs, chords and arpeggios, they all display some level of modularity through repetition of motifs but they are all very distinct from one another. Although Melody1, Melody2 and Melody3 result in the best fitness, the authors found Melody4 and Melody5 to be more pleasing to the ear. This raises questions as to how much merit we should attach to fitness measures such as these — the fitness function can be used to traverse the search space but it did not necessarily lead to the 'best' melody.

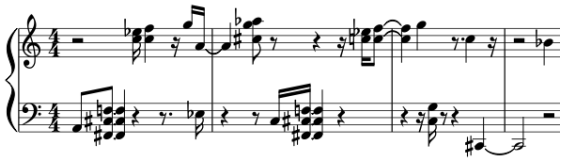


Figure 6: Theme emergent in Melody4.mp3

5.3 Bass Accompaniment

Although some of the melodies, such as that illustrated in Figure 6 are written on both staves, it is a single part melody that is evolved. We ran the experiment again to create both treble and bass parts producing three compositions as described in Section 4.4. In each of these compositions we can hear a repetitive bass part underlying the melody. In Accompany1 and Accompany2 these do not quite fall in time with the upper line, but the fact that they are of strict durations (bass 64, treble 128) keeps the two parts together in a cyclical manner. Accompany3 offers a much more syncopated accompaniment that compliments the treble melody more pleasantly.

A notable, if somewhat obvious, point to make is that it is much more difficult to compose two accompanying lines. One method around this would be to co-evolve the two parts together, although we find something unnatural about this. While there are exceptions where two melodies are composed together, in general when we think of an accompanying line, it is just that: a new part that is written to complement another already composed melody. We have avoided specifying key or time signatures through these experiments, instead opting for ranking and statistical measures to control the content. We feel that this may work well between lines in regards to pitch, as we can constantly measure the harmonic distance between two accompanying parts, but the temporal nature of music gives rise to difficulty when considering rhythm. The repeating measure reported here serves it's function but we acknowledge that it is very limiting. In future work we hope to inves-

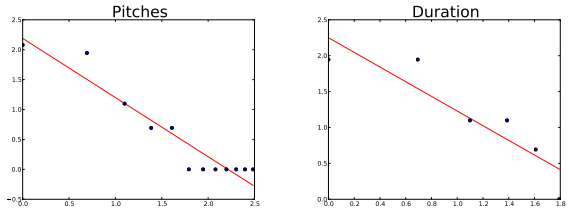


Figure 7: Distribution of the Pitch and Duration attributes in a melody created with a simple Grammar

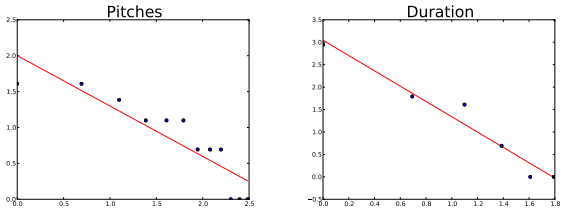


Figure 8: Distribution of the Pitch and Duration attributes in a melody created with a simple Fitness

tigate better methods of creating melodies that temporally and rhythmically complement one another.

5.4 Compositional Elements

The compositions created by this system are largely due to three distinct processes:

1. Representation created by the grammar
2. The Duration and Zipf's Law Fitness Function
3. Repetition of motifs from concatenation of individuals

To determine the significance of each of these aspects, we reran the experiment with varying combinations of each aspect and examined the output. BasicGram1 and BasicGram2 are created using a grammar that only allowed single notes. BasicFit1 and BasicFit2 are evolved with a fitness function that was targeted solely on the length of the melody, disregarding any Zipf-based measurements. ShortMelody is the single best individual evolved, but is not concatenated with any other individuals from the population.

5.4.1 BasicGram

From listening to the melodies created using the basic grammar, it is clear that these are less interesting, less engaging and less pleasant than those created with the more involved grammar. Nevertheless, these BasicGram melodies have equally good (or even better) fitness as our other composite melodies according to our defined fitness function. Figure 7 displays the Zipf's distributions for the pitch and duration attributes for BasicGram1. As expected, these display typical Zipf's distributions with slopes of -0.99 and -1.02. This demonstrates that we need more than a good statistical fitness measure to create a good melody.

5.4.2 BasicFit

BasicFit1 and BasicFit2 were evolved using the full grammar but with a minimal fitness function that only measured the duration of the piece. Thus the best fitness of 1 was reached very quickly, within five generations. Although they were not taken into account during evolution, we calculated the Zipf's distribution for each attribute used in the rest of the experiments. A plot of distribution for the pitch and duration are shown in Figure 8. Although these may initially appear to portray a Zipf-like distribution, a closer analysis shows that the slope for the Pitch and Duration attributes are -0.7 and -1.7 respectively. Similarly, the slopes of the distributions for the pitch-duration, intervals, pitch-distance and bigram attributes were -0.52, -0.58, -0.89 and -0.81. Nevertheless, from listening to these melodies, we find them to be more interesting than those evolved using just the basic grammar.

5.4.3 Short Melody

As discussed in the fitness results, ShortMelody is the best individual found throughout our evolutionary run. It displays all of the compositional elements from the grammar – notes, turns and chords — and it exhibits very accurate Zipf's distribution on each of the measured attributes. Melody1 is the concatenation of this melody with the next top three individuals from the final generation of that run. From listening to both it is evident that the longer concatenated melody is more pleasing and offers more structure than the original short melody on it's own. This aspect of emergent motifs due to repetition is even more evident in other compositions such as Melody4. The degree of repetition is related to the similarity between the selected individuals. In some final generations the top individuals are all very similar giving a high degree of repetition and musical motifs. In other experiments, one or more of the top individuals differ significantly yet have similar fitness. As the concatenation of individuals is one of the most effective methods for creating musicality with this system, we plan to explore this process further with a view to using a similarity measure between individuals as a means of concatenating them into one composition.

Overall, we found that the grammar and representation used in these experimented in combination with the concatenation of a number of highly fit individuals have had a more pleasing aesthetic result in the creation of musical compositions than the use of the Zipf's based fitness. We encourage the reader to evaluate these for themselves, but the authors concurred that in regards to musicality, the melodies created using the full system [1] are much more pleasant to the ear than those from a single individual or those that do not make use of the full grammar. Evolution is driven by the fitness function used, so it is our conclusion that future work should be focussed on finding a more beneficial and musical way of measuring this fitness.

6. CONCLUSIONS

We have composed a series of piano compositions with Grammatical Evolution driven by a Zipf's Distribution of

[1] in particular we enjoyed Melody4, Melody5 and Accompany3

a variety of compositional attributes. A notable issue with the compositions produced is that they lack overall form. We would like to continue this work to develop the progression of the pieces to include a distinctive beginning, middle and end and ideally follow some discernible trajectory as the piece develops. We plan to develope future versions of this system with a better fitness function. Although Zipf's distribution of the attributes measured have been shown in previous literature to correlate with musical pleasantness, we did not find them to be the most important aspect in creating an interesting composition. Instead we found that exploiting GE's use of grammar and concatenating similar but not identical individuals together was more important in the musicality of the result.

While Section 5 offers details and results showing the workings of the experiments run, the best measure of a compositional system is in judging it's musical output. Inherently, this is an aesthetic judgement and it is one that is not easily defined or quantified. Nevertheless, the authors find merit in the compositions produced. We acknowledge that there is a lack of form to the compositions, and that there is notable room for improvement in using the system to create two part melodies. However, as a new system incorporating GE with new grammars and representation it offers worth as a compositional aid; it can create original musical ideas that could be utilised and modified by a human musician in the creation of a larger composition.

7. ACKNOWLEDGMENTS

This work is part of the App'Ed (Applications of Evolutionary Design) project funded by Science Foundation Ireland under grant 13/IA/1850.

8. REFERENCES

[1] M. O'Neill and C. Ryan, *Grammatical evolution*. Springer, 2003.

[2] I. Dempsey, M. O'Neill, and A. Brabazon, *Foundations in grammatical evolution for dynamic environments*. Springer, 2009.

[3] M. O'Neill, J. McDermott, J. M. Swafford, J. Byrne, E. Hemberg, A. Brabazon, E. Shotton, C. McNally, and M. Hemberg, "Evolutionary design using grammatical evolution and shape grammars: Designing a shelter," *International Journal of Design Engineering*, vol. 3, no. 1, pp. 4–24, 2010.

[4] M. Fenton, C. McNally, J. Byrne, E. Hemberg, J. McDermott, and M. O'Neill, "Automatic innovative truss design using grammatical evolution," *Automation in Construction*, vol. 39, pp. 59–69, 2014.

[5] D. Perez, M. Nicolau, M. O'Neill, and A. Brabazon, "Reactiveness and navigation in computer games: Different needs, different approaches," in *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*. IEEE, 2011, pp. 273–280.

[6] M. O'Neill and A. Brabazon, "Evolving a logo design using lindenmayer systems," in *IEEE World Congress on Computational Intelligence*. IEEE, 2008, pp. 3788–3794.

[7] J. Biles, "GenJam: A genetic algorithm for generating jazz solos," in *Proceedings of the International Computer Music Conference*. International Computer Music Association, 1994, pp. 131–131.

[8] J. A. Biles, "Straight-ahead jazz with GenJam: A quick demonstration," in *MUME 2013 Workshop*, 2013.

[9] K. Thywissen, "GeNotator: an environment for exploring the application of evolutionary techniques in computer-assisted composition," *Organised Sound*, vol. 4, no. 02, pp. 127–133, 1999.

[10] H. Göksu, P. Pigg, and V. Dixit, "Music composition using genetic algorithms (GA) and multilayer perceptrons (MLP)," in *Advances in Natural Computation*. Springer, 2005, pp. 1242–1250.

[11] P. Dahlstedt, "Autonomous evolution of complete piano pieces and performances," in *Proceedings of Music AL Workshop*. Citeseer, 2007.

[12] A. O. de la Puente, R. S. Alfonso, and M. A. Moreno, "Automatic composition of music by means of grammatical evolution," in *ACM SIGAPL APL Quote Quad*, vol. 32, no. 4. ACM, 2002, pp. 148–155.

[13] J. Reddin, J. McDermott, and M. O'Neill, "Elevated Pitch: Automated grammatical evolution of short compositions," in *Applications of Evolutionary Computing*. Springer, 2009, pp. 579–584.

[14] J. Shao, J. McDermott, M. O'Neill, and A. Brabazon, "Jive: A generative, interactive, virtual, evolutionary music system," in *Applications of Evolutionary Computation*. Springer, 2010, pp. 341–350.

[15] B. Manaris, J. Romero, P. Machado, D. Krehbiel, T. Hirzel, W. Pharr, and R. B. Davis, "Zipf's law, music classification, and aesthetics," *Computer Music Journal*, vol. 29, no. 1, pp. 55–69, 2005.

[16] B. Manaris, D. Vaughan, C. Wagner, J. Romero, and R. B. Davis, "Evolutionary music and the Zipf-Mandelbrot law: Developing fitness functions for pleasant music," in *Applications of Evolutionary Computing*. Springer, 2003, pp. 522–534.

[17] G. K. Zipf, *Human behavior and the principle of least effort.* Addison-Wesley Press, 1949.

[18] R. Waschka II, "Composing with genetic algorithms: GenDash," in *Evolutionary Computer Music*. Springer, 2007, pp. 117–136.

[19] A. Eigenfeldt and P. Pasquier, "Populations of populations: composing with multiple evolutionary algorithms," in *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Springer, 2012, pp. 72–83.

MODELING OF PHONEME DURATIONS FOR ALIGNMENT BETWEEN POLYPHONIC AUDIO AND LYRICS

Georgi Dzhambazov, Xavier Serra
Music Technology Group
Universitat Pompeu Fabra,
Barcelona, Spain
{georgi.dzhambazov,xavier.serra}@upf.edu

ABSTRACT

In this work we propose how to modify a standard approach to text-to-speech alignment for solving the problem of alignment of lyrics and singing voice. To this end we model the duration of phonemes, specific to the case of singing. We rely on a duration-explicit hidden Markov model (DHMM) phonetic recognizer based on mel frequency cepstral coefficients (MFCCs), which are extracted in a way robust to background instrumental sounds. The proposed approach is tested on polyphonic audio from the classical Turkish music tradition in two settings: with and without modeling phoneme durations. Phoneme durations are inferred from sheet music. In order to assess the impact of the polyphonic setting, alignment is evaluated as well on an acapella dataset, compiled especially for this study. Results show that the explicit modeling of phoneme durations improves alignment accuracy by absolute 10 percent on the level of lyrics lines (phrases). Comparison to state-of-the-art aligners for other languages indicates the potential of the proposed method.

1. INTRODUCTION

Lyrics are one of the most important aspects of vocal music. When a performance is heard, most listeners will follow the lyrics of the main vocal melody. The goal of automatic lyrics-to-audio alignment is to generate a temporal relationship between textual lyrics and sung audio. In this particular work, the goal is to detect the start and end times of every phrase from lyrics.

The problem of lyrics-to-audio alignment has inherent relation to text-to-speech alignment. For spoken utterances phonemes have relatively similar duration across speakers. Unlike that, in singing durations of phoneme (especially vowels) have higher variation [1]. When being sung, vowels are prolonged according to musical note values, which in term have intrinsic relation to musical meter (e.g. duration could align with beats in a musical bar).

Another aspect that distinguishes speech from music is that unlike clean speech, singing voice is accompanied by background instruments. Instrumental accompaniment and non-vocal segments can deteriorate significantly the alignment accuracy.

The goal of this study is to test the hypothesis that extending a state-of-the-art system for automatic lyrics-to-audio alignment with modeling of phoneme durations, can improve its accuracy. More specifically, we aim to show that durations of vocals (inferred from musical score) can guide the recognition process in cases when it looses track in polyphonic audio. Such guidance can be compared to the way modeling prosodic rules helps in automatic speech understanding.

While being aided by sheet music, our modeling approach allows at the same time room for certain temporal flexibility to handle cases of expressive singing, in which vocals are sustained in a way not obeying the reference sheet music. The proposed approach was tested on polyphonic audio from the classical Turkish tradition which is characterized by a high degree of expressive singing, thus providing challenging material with versatile temporal deviations.

2. RELATED WORK

To date most of the studies of automatic lyrics-to-audio alignment exploit phonetic acoustic features and state-of-the-art work is based on a phoneme recognizer [2, 3].

An example of such a system [2] relies on hidden Markov model (HMM) and was tested on Japanese popular music. To reduce the spectral content of background instruments, the authors perform automatic segregation of the vocal line. Then Viterbi forced alignment [4] is run utilising mel frequency cepstral coefficients (MFCCs) extracted from the vocal-only signal. In both [2] and [3] the phoneme models are trained on speech and later adapted to singing voice. This is necessary because of the lack of a big enough training singing voice corpus. In [2] additionally an adaptation to the voice of a particular singer is carried out.

In other works the duration of the lyrics has been applied as a reinforcing cue: In [5] relative estimated durations are inferred directly from textual lyrics. The estimation process is done based on supervised training on singing voice.

A common-occurring drawback of HMMs is that their capability to model exact state durations is restricted. The wait time in a state is implicitly set to a geometric distribution (derived from the self-transition likelihood). Duration is usually modeled by duration-explicit hidden Markov models (DHMM) (a.k.a. hidden semi-Markov models). In

Copyright: ©2015 Georgi Dzhambazov 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.

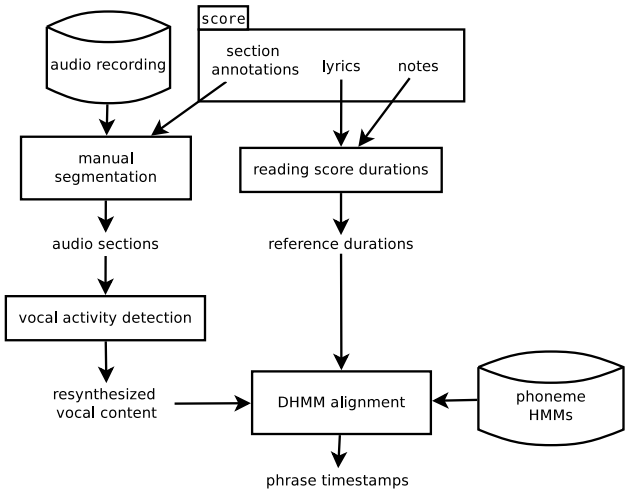


Figure 1. Overview of the modules of the proposed approach. Leftmost column represents audio preprocessing steps, while the middle column shows how durations are modeled.

DHMMs the underlying process is allowed to be a semi-Markov chain with variable duration of each state [6]. Each state in turn can be assigned any statistical distribution. DHMMs have been shown to be successful for modeling chord durations in automatic chord recognition [7].

3. PROPOSED SYSTEM

Similar to [2] in this work we develop a phoneme-recognizer-based forced alignment employing the Viterbi algorithm [4] to decode the most optimal state sequence.

We have adopted the idea of [7] not to explicitly add states for durations in the model, but instead to extend the Viterbi decoding to handle duration of states. For brevity in the rest of the paper our model will be referred to as DHMM.

Figure 1 presents an overview of the proposed system. An audio recording and its corresponding score are input. Relying on HMMs of phonemes the DHMM returns start and end timestamps of aligned lyrical phrases.

First an audio recording is manually divided into sections (e.g. verse, chorus) as indicated in the score, whereby instrumental-only sections are discarded. All further steps are performed on each audio section. If we had used automatic segmentation instead, potential erroneous lyrics and durations could have biased the comparison of a baseline system and DHMM. As we focus on evaluating the effect of DHMM, manual segmentation is preferred.

3.1 Vocal activity detection (VAD)

Next a predominant singing voice detection (a.k.a. vocal activity detection) method is applied on each section to attenuate the spectral content from accompanying instruments, because they have negative effect on the alignment. We utilize a method that performs detection of segments with predominant singing voice and in the same time melody transcription for the detected segments [8].

3.1.1 Vocal resynthesis

For the regions with predominant vocal, based on the extracted melodic contours and a set of peaks in the original spectrum, the vocal content is resynthesized as separate audio using a harmonic model [9]. A problem in the resynthesis are spectral peaks of the singing voice, for which there is overlap with peaks from the spectrum of a background instrument. These distorted peaks lead to deformation of the original voice timbre. To detect these peaks we apply the main-lobe matching technique [10]. The detected spectral peaks have been excluded from the harmonic series in the harmonic model. [1] More details and examples of the resynthesis step can be found in [11].

3.2 Reading score durations

For each lyrics syllable a reference duration is derived from the values of its corresponding musical notes. Then the reference duration is spread among its constituent phonemes, whereby consonants are assigned constant duration and the rest is assigned to the vowel.

Each phoneme is modeled by a 3-state HMM. The three states represent the initial, sustain and decay phase of the phoneme acoustics. A lookup table of reference durations R_i for each state i is constructed from the reference phoneme durations. [2] We assume that the duration d for a state i may vary according to a normal distribution $P_i(d)$ with mean at the reference duration $d = R_i$ and a global for all phonemes standard deviation σ. To align a given recording the score-inferred lengths are linearly rescaled to match its musical tempo. In this work the unit of R_i is number of acoustic frames.

3.3 Duration-explicit HMM alignment

For each phoneme a HMM is trained from a corpus of turkish speech utilizing MFCCs. For given lyrics, the words are expanded to phonemes based on grapheme-to-phoneme rules for Turkish [12, Table 1] and the trained HMMs are concatenated into a phoneme network. The network is then aligned to the MFCC features, extracted from the resynthesized audio signal, by means of the duration-explicit decoding. In what follows we describe a variation of Viterbi decoding method, in which maximization is carried over the most likely duration for each state. The decoding is adapted from the procedure described in [7]. Let us define:

R_{max} : $\max_i(R_i) + \sigma$

$b_i(O_t)$: observation probability for state i for feature vector O_t (complying with the notation of [4])

$\delta_t(i)$: probability for the path with highest probability ending in state *i* at time *t* (comply with the notation of [4, III. B]))

[1] In fact, resynthesis is not an obligatory step, but was performed in order to allow to track the intelligibility of different vocals after the application of the vocal detection and main-lobe matching.

[2] We used the simple strategy of assigning equal duration to each of the three states within a phoneme

3.3.1 Recursion

For $R_{max} < t \leq T$

$$\delta_t(i) = \max_d\{\delta_{t-d}(i-1). P_i(d)^\alpha [B_t(i,d)]^{1-\alpha}\} \quad (1)$$

where

$$B_t(i,d) = \Pi_{s=t-d+1}^{t} b_i(O_s) \quad (2)$$

is the observation probability of staying d frames in state i until frame t. The domain of d is $(\max\{R_i - \sigma, 1\}, R_i + \sigma))$ and complies to a normal distribution, but is reduced for states with reference duration $R_i < \sigma$.

A duration back-pointer is defined as

$$\chi_t(i) = \arg\max_d\{\delta_{t-d}(i-1). P_i(d)^\alpha [B_t(i,d)]^{1-\alpha}\} \quad (3)$$

Note that in forced alignment the source state could be only the previous state $i-1$.

To be able to control the influence of the duration we have introduced a weighting factor α. Note that setting α to zero is equivalent to using a uniform distribution for $p_i(d)$.

3.3.2 Initialization

For $t \leq R_{max}$

$$\delta_t(i) = \max\{\delta_t(i)^*, \kappa_t(i)\} \quad (4)$$

where a reduced-duration delta $\delta_t(i)^*$ is defined in the same way as in (1) but

$$d \in \begin{cases} \varnothing, & t \leq R_i - \sigma \\ (R_i - \sigma, \min\{t-1, R_i + \sigma\}), & else \end{cases} \quad (5)$$

reduces the duration to $t-1$ when $t < R_i + \sigma$. Lastly the probability of staying at initial state i at time t is defined as:

$$\kappa_t(i) = \pi_i P_i(t)^\alpha [\Pi_{s=1}^{t}(O_s)]^{1-\alpha} \quad (6)$$

for $t \in (1, R_i + \sigma)$.

3.3.3 Backtracking

Finally the decoded state sequence is derived by backtracking starting at the last state N and switching to a source state a number of $d = \chi_t(i)$ frames ahead according to the backpointer from (3).

4. EXPERIMENTAL SETUP

Alignment is performed on each manually divided audio section and results are reported per recording (one total for its sections). [3]

To assess the benefit of duration modeling for alignment a comparison to a baseline system with Viterbi decoding with no state durations (as proposed by [4]) is conducted.

[3] To assure reproducibility of this research we publish source code at `https://github.com/georgid/AlignmentDuration`

total #sections	#phrases per section	section duration
75	2 to 5	7-20 seconds

Table 1. Section and phrase statistics for test dataset.

We present results for the most optimal value of $\alpha = 0.97$. It was found by minimizing the alignment error (see section 4.2) on a separate development dataset of 20 minutes Turkish acapella recordings. To assure optimality we aligned on word-level ground truth.

To train the speech model the HMM Toolkit (HTK) [13] is employed. The acoustic properties (most importantly the formant frequencies) of spoken phonemes can be induced by the spectral envelope of speech. To this end, we utilise the first 12 MFCCs and their delta to the previous time instant.

A 3-state HMM model for each of 38 Turkish phonemes is trained, plus a silent pause model. For each state a 9-mixture Gaussian distribution is fitted on the feature vector.

4.1 Datasets

The test dataset consists of 12 single-vocal recordings of 9 compositions with accompaniment with total duration of 19:00 minutes [4]. The compositions are drawn from the CompMusic corpus of classical Turkish Makam repertoire [14]. Scores are provided in the machine-readable *symbTr* format [15].

Additionally a separate acapella dataset of the same 12 recordings sung by professional singers has been recorded especially for this study. It can be considered a vocal-track-only version of the original polyphonic dataset [5]. Evaluation on the acapella corpus was conducted in order to assess the impact of the vocal extraction step.

Each song section was manually annotated into musical phrases as proposed by [16]. A musical phrase usually corresponds to a lyrical line. If a phrase boundary splits a word we have modified it to include the complete word. Short instrumental motives have not been excluded from the phrases. Furthermore we split or merged some melodic phrases so that phrases within a recording have roughly the same number of musical bars (1 or 2). Table 1 presents statistics about phrases.

4.2 Evaluation metrics

Alignment is evaluated in terms of alignment accuracy (AA) as the percentage of duration of correctly aligned regions from total audio duration (see [2, Fig.9] for an example). A value of 100 means perfect matching of phrase boundaries. We report as well the mean of the alignment error (AE): it measures the absolute error (in seconds) at the start and end timestamp of a phrase.

We define a metric *musical score in-sync* (MSI) to measure the approximate degree to which a singer performs a recording in synchronization with note values indicated in the musical score. Thus low accuracy of MSI indicates a

[4] Dataset is available at `http://compmusic.upf.edu/turkish-sarki`

[5] Dataset is available at `http://compmusic.upf.edu/turkish-makam-acapella-sections-dataset`

System variant	accuracy	error
musical score in-sync	88.14	0.32
HMM polyphonic	67.46	1.04
DHMM polyphonic	77.74	0.63
DHMM acapella	90.04	0.26
HMM+adaptation [3]	-	1.4
HMM+singer adaptation [2]	85.2	-

Table 2. Alignment accuracy (in percent) for musical score in-sync; different system variants: baseline HMM and DHMM; state-of-the-art for other languages. Alignment accuracy is reported as total for all recordings. Additionally the total mean phrase alignment error (in seconds) is reported

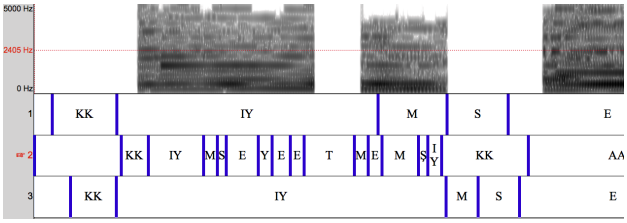


Figure 3. Example of decoded phonemes. *very top*: resynthesized spectrum; *upper level*: ground truth, *middle level*: HMM; *bottom level*: DHMM; (excerpt from the recording 'Kimseye etmem şikayet' by Bekir Unluater). Notice that no spectrum is resynthesized for regions with unvoiced consonants.

higher temporal deviation from musical score. We compute MSI per a recording as the AA of score-inferred reference durations R_i (defined in section 3.2) compared to ground-truth, as if they were results after alignment.

5. RESULTS

Table 2 presents comparison of the proposed DHMM system performance and a baseline HMM system. It can be observed that modeling of note values with DHMM increases HMM accuracy by 10 absolute percent. One reason for this are cases of long vocals, in which HMM switches to the next phoneme prematurely. One reason for this might be that the HMM is trained on speech and cannot stay long enough in a given state). In contrast, the duration-explicit decoding allows picking the optimal duration (which can be traced in an example in figure 3).

Figure 2 allows a glance at results per recording, ordered according to MSI. [6] . It can be observed that DHMM performs consistently better than the baseline (with some exceptions of where accuracy is close). Unlike the relatively stable accuracy for the acapella case, when background instruments are present, the accuracy variates more among recordings.

We compare our alignment results as well to the best hitherto alignment systems: one for English pop songs [3] and one for Japanese pop [2]. These are abbreviated in table 2 respectively as *HMM+adaptation* and *HMM+singer adaptation*. In these works alignment is evaluated also on the level of a lyrical line/phrase. Except for the duration-explicit decoding scheme, our approach differs from both works essentially in that they conduct speech-to-singing-voice adaptation. Unlike that we did not perform any adaptation of the original speech model. Adaptation data of clean singing voice for a particular singer might not always be available and thus does not allow the system to scale to data from unknown singers.

Apart from that, the VAD module of [2] showed to notably increase the average accuracy of 72.1 % for a baseline, to accuracy of 85.2 % for their final system. Similarly, we observe that evaluation on the acapella dataset yields an accuracy by about the same percent higher than the polyphonic one (see table 2). Investigating our results with low accuracy revealed that false positives of our VAD module is a considerable reason for misalignment. Since *HMM+adaptation* and *HMM+singer adaptation* are tested on material with different genre and language, no direct conclusions are possible. However, the comparable range of the results indicates a potential of our approach to perform on par with these systems, especially by further improving our VAD step.

[6] Per-recording results are published at `https://drive.google.com/file/d/0B4bIMgQlCAuqY3hKc25WTm9kTEk/view?usp=sharing`

6. CONCLUSION

In this work we evaluated the behavior of a HMM-based phonetic recognizer for lyrics-to-audio alignment in two settings: with and without utilising lyrics duration information. Using duration-explicit modeling for the former setting outperformed the latter for polyphonic Turkish classical recordings.

Importantly our approach reaches accuracy comparable to state of the art alignment systems by using an acoustic model trained on speech only. Furthermore, results outlined that the DHMM performs considerably better on an acapella version of the test dataset, which indicates that improving the vocal activity detection module can result in even better accuracy, which we plan to address in future work.

A limitation of the current alignment system is the prerequisite for manually-done structural segmentation, which we plan to automate in the future.

In general, the proposed approach is applicable not only when musical scores are available, but also for any format, from which duration information can be inferred: for example annotated melodic contour or singer-created indications along the lyrics.

Acknowledgments

This work is partly supported by the European Research Council under the European Unions Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583) and partly by the AGAUR research

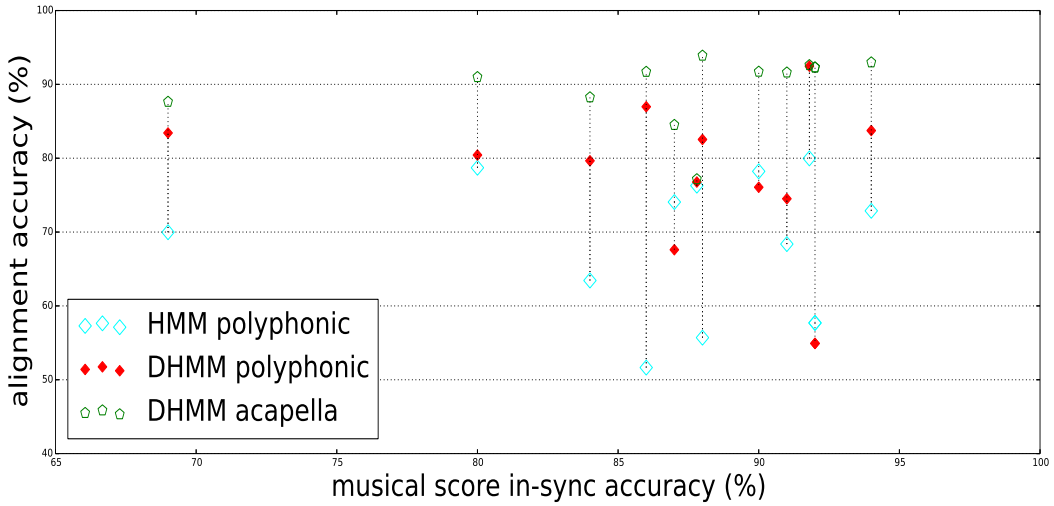


Figure 2. Comparison between results from DHMM (for both polyphonic and acapella) and baseline HMM. The metric used is alignment accuracy. A connected triple of shapes represents results for one recording. Results are ordered according to *musical score in-sync* (on horizontal axis)

grant.

7. REFERENCES

[1] A. M. Kruspe, "Keyword spotting in a-capella singing," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 2014, pp. 271–276.

[2] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno, "Lyricsynchronizer: Automatic synchronization system between musical audio signals and lyrics," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 6, pp. 1252–1261, 2011.

[3] A. Mesaros and T. Virtanen, "Automatic alignment of music audio and lyrics," in *in Proceedings of the 11th Int. Conference on Digital Audio Effects (DAFx-08*, 2008.

[4] 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.

[5] Y. Wang, M.-Y. Kan, T. L. Nwe, A. Shenoy, and J. Yin, "Lyrically: automatic synchronization of acoustic musical signals and textual lyrics," in *Proceedings of the 12th annual ACM international conference on Multimedia*. ACM, 2004, pp. 212–219.

[6] S.-Z. Yu, "Hidden semi-markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010.

[7] R. Chen, W. Shen, A. Srinivasamurthy, and P. Chordia, "Chord recognition using duration-explicit hidden markov models." in *ISMIR*. Citeseer, 2012, pp. 445–450.

[8] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 6, pp. 1759–1770, 2012.

[9] X. Serra, "A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition," Tech. Rep., 1989.

[10] 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, pp. 2145–2154, 2010.

[11] G. Dzhambazov, S. Sentürk, and X. Serra, "Automatic lyrics-to-audio alignment in classical Turkish music," in *The 4th International Workshop on Folk Music Analysis*, 2014, pp. 61–64.

[12] Ö. Salor, B. L. Pellom, T. Ciloglu, and M. Demirekler, "Turkish speech corpora and recognition tools developed by porting sonic: Towards multilingual speech recognition," *Computer Speech and Language*, vol. 21, no. 4, pp. 580 – 593, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230807000022

[13] S. J. Young, *The HTK hidden Markov model toolkit: Design and philosophy*. Citeseer, 1993.

[14] B. Uyar, H. S. Atlı, S. Şentürk, B. Bozkurt, and X. Serra, "A corpus for computational research of Turkish makam music," in *1st International Digital Libraries for Musicology Workshop*, London, United Kingdom, 2014, pp. 57–63. [Online]. Available: http://sertansenturk.com/uploads/publications/uyar2014corpus_dlfm.pdf

[15] M. K. Karaosmanoğlu, "A Turkish makam music symbolic database for music information retrieval: Symbtr," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012.

[16] M. K. Karaosmanoğlu, B. Bozkurt, A. Holzapfel, and N. Doğrusöz Dişiaçık, "A symbolic dataset of Turkish makam music phrases," in *Fourth International Workshop on Folk Music Analysis (FMA2014)*, 2014.

Generalizing Messiaen's Modes of Limited Transposition to a n-tone Equal Temperament

Adriano Baratè
Laboratorio di Informatica Musicale
Dipartimento di Informatica
Università degli Studi di Milano, Milan, Italy
Via Comelico, 39 20135 Milano, Italy
barate@di.unimi.it

Luca A. Ludovico
Laboratorio di Informatica Musicale
Dipartimento di Informatica
Università degli Studi di Milano, Milan, Italy
Via Comelico, 39 20135 Milano, Italy
ludovico@di.unimi.it

ABSTRACT

Modes of limited transposition are musical modes originally conceived by the French composer Olivier Messiaen for a tempered system of 12 pitches per octave. They are defined on the base of symmetry-related criteria used to split an octave into a number of recurrent interval groups. This paper describes an algorithm to automatically compute the modes of limited transposition in a generic n-tone equal temperament. After providing a pseudo-code description of the process, a Web implementation will be proposed.

1. INTRODUCTION

Olivier Messiaen is considered one of the most important composers of the 20th Century. His production includes not only music pieces, but also theoretical works about his musical language. Concerning the latter aspect, the interest in ancient Greek music and exotic modes was already clear in his early compositions. For instance, while a student he experimented with his theories about new music modes in his first published works, *Eight Preludes for piano*, and throughout his life Messiaen continued to develop and evolve new composition techniques, always integrating them into his musical style. In [1] Messiaen's music has been described as outside the Western musical tradition, although growing out of that tradition and being influenced by it.

With respect to music scales, the most relevant innovation introduced by Messiaen is probably the definition and adoption of *modes of limited transposition* (in French: *modes à transpositions limitées*). In order to create new music resources for harmony, he determined all the ways to split an octave into recurrent groups of notes, where each group was internally formed by the same intervals and groups overlapped as regards their boundaries, namely the highest pitch of a group was the lowest of the following one. The way to compute note groups for each mode will be described in detail in next sections, and some clarifying examples will be provided too. For a formal description of Messiaen's mode, please refer to [2].

In his theoretical works and music pieces, Messiaen was always referring to the equal division of an octave into 12 steps (12-EDO), commonly in use in Western music. Our goal is applying Messiaen's theories to a generalized n-tone equal temperament, where the original modes of limited transposition represent a special case. In order to achieve this goal, first we need to define some concepts, since Western music theory and score representation cannot be applied to this generalization in a straightforward way.

2. GENERALIZATION TO N-TONE EQUAL TEMPERAMENT

Even if Olivier Messiaen sought to overcome the limitations imposed by Western music system, his works had their roots in that musical culture and tradition. As regards melodic and harmonic aspects, Western music is largely based on 12-EDO. In this context, an octave is composed of 12 steps, where every pair of adjacent pitches has an identical frequency ratio, equal to $\sqrt[12]{2}$. In this way, 12-EDO divides the octave into 12 parts, known as *semitones* or *half tones*, which are the smallest musical interval commonly used in Western tonal music. Adopting equal temperament implies that all semitones are equal on a logarithmic scale. Since pitch is (roughly) perceived as the logarithm of frequency, the distance from every step to its nearest neighbor is the same for every step in the system.

In general terms, equal temperament is not the only possibility. Well-known even in ancient times [1] and far cultures, [2] this kind of temperament was extensively used in the European tradition only from the 16th Century, whereas Pythagorean tuning, Ptolemaic sequence and Zarlinian modality had been mainly adopted in earlier music [3]; 5-, 7- and 9-EDO are fairly common in ethnomusicology. For example, [4] discusses the issue of temperament in Thai music, whereas [5] analyses Javanese gamelan.

[1] One of the earliest descriptions of equal temperament is contained in the writing entitled *Elements of Harmony* by Aristoxenus, dating back to the 4th Century BC.

[2] For instance, in China an approximation for equal temperament was described by He Chengtian around 400 AD, whereas the *Complete Compendium of Music and Pitch* published by Zhu Zaiyu in 1584 contains a detailed discussion of this pitch theory with a precise numerical specification.

Copyright: ©2015 Adriano Baratè 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.

Another possible generalization is the application of equal temperament to non-octave intervals, thus passing from the concept of equal division of an octave into n subparts (n-EDO) to the n-tone equal temperament (n-TET). For instance, the equal-tempered version of the Bohlen-Pierce scale, described in [6] and [7], is based on the ratio 3:1. Such an interval, corresponding to a perfect fifth plus an octave in 12-EDO, is split into 13 equal parts. Consequently, every pair of adjacent pitches presents a frequency ratio equal to $\sqrt[13]{3}$.

A great number of equal divisions either of the octave or of other intervals have found use in microtonal music, ethnic cultures, theoretical experiences, etc. An in-depth discussion of tuning and temperament clearly goes beyond the goals of this work. For further details, please refer to [8] or [9].

Now we want to provide an extension of twelve-tone system, thus defining a generic n-TET where a given interval can be divided into n equally-spaced pitches. In order to avoid ambiguity with in-use terminology, we will define any available pitch of the equal temperament as a step. Each couple of adjacent steps presents a frequency ratio equal to $\sqrt[n]{r}$, where r is the frequency ratio of the interval to be subdivided and n is the number of equal steps. Since Messiaen defines his modes by splitting the octave, in the following we will focus on that interval, nonetheless our approach can be easily extended to any other interval.

Please note that reasoning in terms of steps instead of fixed frequencies allows an abstract description of the process. The modes defined in this way will be potentially instanced starting from any frequency, either available in the standard tuning system [10] or not.

3. INTRODUCTION TO MODES OF LIMITED TRANSPOSITION

According to many musicologists and experts, Messiaen's modes of limited transposition are the most relevant resource he used to create melody and harmony in music. The original idea was determining all the possible ways to split the tempered twelve-tone octave in a number of recurrent and non-overlapping note groups. Each group has to present the same internal pattern, made of a variable number of variable-size intervals. The smallest interval to build structures is the tempered semitone, but semitones can be aggregated to build bigger intervals.

In his theoretical works, Messiaen defines as *modes* the recurring note groupings which are limited in the amount of times they can be transposed, due to patterns within their structures. Based on a tempered system of 12 pitches, these modes are formed by several symmetrical groups, the last note of each group always being common with the first of the following group. At the end of a certain number of chromatic transpositions that varies with each mode, they are no longer transposable, giving exactly the same notes as the first [2].

In mathematics, this problem recalls the concept of *composition of an integer*. A composition of an integer n is a way of writing n as the sum of a sequence of (strictly) positive integers. Two sequences that differ in the order of their terms define different compositions of their sum, while they are considered to define the same partition of that number. Each positive integer n has 2^{n-1} distinct compositions.

In mathematical terms, the process adopted by Messiaen allows to find all the compositions of 12 where 12 is the number of semitones in an octave that match an additional criterion, namely those presenting a pattern made of k repetitions, with $k > 1$. For example, both

$$12 = 1 + 3 + 1 + 3 + 1 + 3$$

and

$$12 = 3 + 1 + 3 + 1 + 3 + 1$$

satisfy this condition, since it is possible to recognize one group that is repeated 3 times: in the former case, the recurrent group is [1, 3], in the latter [3, 1]. On the contrary,

$$12 = 1 + 3 + 3 + 1 + 1 + 3,$$

which in mathematical terms would be another composition of 12, does not match the condition, since we cannot determine $k > 1$ repetitions of the same pattern inside the composition.

A relevant property of this redefinition of the concept of composition is cycle invariance. In other terms, if we consider the interval sequence in a circular way, left- and right-shifting do not introduce new models. Invariance can be applied to the complete sequence as well as to groups.

4. REPRESENTATION ISSUES

A problem to face is the textual and score representation of steps in a generic n-TET. In fact, both pitch names and their corresponding staff position have been originally conceived for a diatonic scale, namely a musical scale composed of seven pitches. The granularity of semitones can be textually and graphically rendered through the use of accidentals, a practice that however introduces ambiguity in the spelling of enharmonic equivalents. Going deeply into microtonal music language, commonly-accepted representations are available only for specific subdivisions, such as quarter tones in 12-EDO.

Messiaen had to manipulate semitones and their possible aggregations in the context of a 12-step chromatic scale. In this context, a staff view could be provided - and actually was provided by the author - but it would require spelling notes and solving enharmonic ambiguities, which is not strictly necessary.

In the generalized case we are addressing, the problem is assigning a name and providing an effective graphical representation to each of the n subdivisions of an arbitrary interval. This issue will be discussed in the following subsections.

4.1 Pitch Naming

A practical solution to naming problems is the adoption of pitch classes as defined in [11]. Pitch classes are an abstraction of pitches divorced from register, notation and compositional realization [12], and they can be effectively

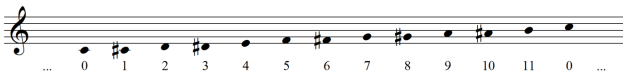


Figure 1. An example of assignment of integers to pitch classes. Any enharmonically-equivalent note spelling would be represented by the same integer value.

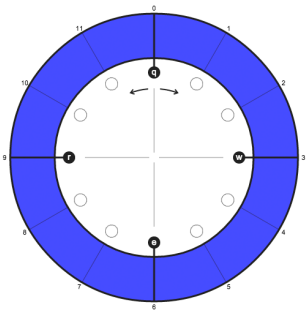


Figure 2. A ring diagram that provides a cyclic representation of pitch classes in 4-TET. The numbering of sectors is arbitrary as regards both the origin and the direction.

notated by assigning $pc = 0$ to a given step and consecutive integers to consecutive steps. Figure 1 shows one of the 12 possible correspondences among semitones in the 12-EDO and integer values, specifically the one where $pc = 0$ is assigned to C. In the definition of pitch classes, the octave is not relevant and the sequence of symbols can be read in a circular way, or in other words the system is modulo 12. This cyclic approach to pitch representation is coherent with Messiaen's one.

Please note that the adoption of pitch classes intrinsically solves the problem of score-spelling ambiguities: two spellings - like E♯ and F - that in an equal-tempered system produce the same sound, namely correspond to the same frequency, collapse into the same pitch class (e.g. $pc = 5$).

4.2 Ring Diagrams

As regards a graphical rendering suitable for our theoretical goals, we decided to represent tempered steps through a periodic tiling (or *tessellation*) of an annulus. Figure 2 provides an example where a generic interval has been divided into 4 equal steps. This diagram does not contain references to in-use note names: conventionally, each adjacent sector can be identified through consecutive numbers, and the origin can be set to any sector. Ring diagrams can be read (and their sectors can be numbered) both clockwise and counter clockwise.

A color code has been added to each sector, in order to visually mark both groups and intervals. Color combinations mainly have two purposes:

- Making visible the sequence of groups, all characterized by the same internal layout. Groups constitute the tessellation of the complete interval to be subdivided;
- Highlighting the internal composition of each group, in terms of intervals, i.e. step aggregations. The same sequence of intervals can be found in any group.

This chromatic approach implies that each group is made of one (consecutive) block per color, and when the color sequence restarts a new instance of the group occurs, as shown in Figure 3.

Please note that, inside a group, also intervals with the same size have different colors. For example, let us consider one of the possible subdivisions of the global interval, say a specific subdivision of an octave into 12 steps which originates three 4-step groups [1, 2, 1]. In our representation, the two single-step intervals inside the group have different colors, whereas the color layout composed in this case by 3 different colors is repeated group by group. The letter sequence of a typical QWERTY layout has been adopted in order to identify blocks and to play the corresponding pitch, as explained in Section 8.

One of the advantages of ring diagrams is providing a cyclic representation of groups, in accordance with the concept of pitch class and Messiaen's theory about modes. Besides, this approach offers the possibility to read diagrams either clockwise or counter clockwise, provided that the same criterion is used for all diagrams. Finally, such a graphical representation allows the reader to choose the boundary of any colored block as the starting point, which intrinsically solves the issue of equivalent group spellings. For instance, a group made of 4 steps can be spelled as [3, 1] or equivalently as [1, 3], since the latter case simply implies building the group from the second pitch, as shown by the ring diagrams in Figure 4.

For the sake of clarity, in our diagrams the minimum

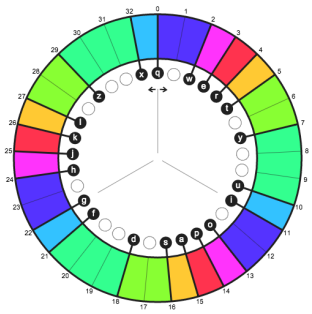


Figure 3. A ring diagram that highlights the [2,1,1,1,2,3,1] spelling of an 11-transposition mode in 33-TET.

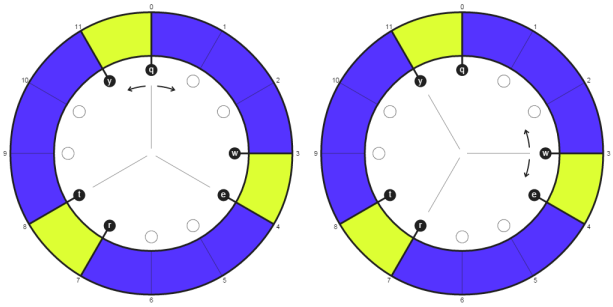


Figure 4. Ring diagrams for two equivalent spellings: [3,1] and [1,3].

interval (namely the step) is always surrounded by radial lines that delimit its extension. These lines are drown with strong strokes where a step aggregation starts or ends, whereas boundaries inside an aggregation are thin. However, colors and thin lines simply provide graphic hints to the user: the diagram's semantics resides in the specific tessellation of the annulus, which is different from a mode to another. As mentioned above, different spellings of the same mode can be obtained through suitable rotations of the diagram.

5. AN ALGORITHM TO CALCULATE GENERALIZED MODES OF LIMITED TRANSPOSITION

From a historical point of view, only some temperaments have been considered, due to their application to specific context (e.g. in ethnomusicology) or to theoretical reasons (for instance, the adherence of a given interval in a temperament to its theoretical value in terms of frequency ratio). Our goal is investigating the generalized equal temperaments by following an automatic approach.

In this section we will describe an algorithm to compute all the possible modes emerging from a given subdivision of an arbitrary interval. In order to perform calculations, the only required input is the number of steps we want to consider, i.e. the minimum granularity to build aggregations. If we need an audio rendering too, two more inputs are necessary, namely the frequencies of the pitches that delimit the global interval to be divided.

The algorithm can be decomposed into 3 steps:

1. *Calculating all the integer divisors of the global interval*. The key requirement by Messiaen is covering such an interval through a number $k > 1$ of occurrences of the same pattern. This implies that each group is made of an equal (integer) number of steps, say s. Consequently, the size n of the global interval is split into smaller groupings according to the following equation: $n = k \cdot s$. Since $k > 1$, n is not considered a divisor of itself, which adheres to Messiaen's theories. Provided that n has been set, the purpose of this step is finding all suitable values for k, and consequently for s. Please note that s is also the number of transpositions for a given mode, since after s 1-step transpositions pitches are repeated. For example, a chromatic scale in 12-EDO ($n = 12$, $k = 12$, $s = 1$) is a mode presenting only one transposition, whereas the whole-tone scale ($n = 12$, $k = 6$, $s = 2$) has two transpositions, misaligned by one semitone;

2. For each grouping size s, *finding all the compositions of* s, i.e. any way to write s as a sorted sum of positive integers. If the algorithm goes from smaller to bigger values, covering not only the mentioned divisors but any integer in the range $[1 \dots (s-1)]$, each iteration can benefit from already available compositions. For instance, the fifth iteration aims at finding the compositions of 5. One of them is $5 = 4 + 1$, but all the compositions of 4 have been already computed during the fourth iteration and can be reused here. A well-known programming technique to implement this behavior is recursion;

3. *Pruning*, i.e. removing unwanted values from data structures. The algorithm does not produce wrong results, nonetheless some values need to be purged. First, as some compositions directly come from aggregations of more atomic ones, in this context they are redundant. For instance, [2, 2] is a spelling of 4, but the mode built through the repetition of [2, 2] is indistinguishable from the mode made of single [2], already available inside the data structures. Besides, we have to manage the mentioned equivalent spellings, like [1, 2, 3], [2, 3, 1] and [3, 1, 2], corresponding to different ways to read the same interval pattern.

Key	Value
s = 1	[1]
s = 2	[2] ~~[1,1]~~
s = 3	[3] [2,1] ~~[1,2]~~ ~~[1,1,1]~~
s = 4	[4] [3,1] [2,1,1] ~~[1,3]~~ ~~[1,2,1]~~ ~~[1,1,2]~~ ~~[1,1,1,1]~~
s = 6	[6] [5,1] [4,2] [4,1,1] [3,2,1] [3,1,1,1] [2,3,1] [2,2,1,1] [2,1,1,1,1] ~~...~~

Table 1. The results for the computation of all available modes in 12-EDO. Slashed values are the ones removed by pruning. Intentionally the last row does not contain unwanted values since their number would be too high.

The data structure used to contain final results is a dictionary whose elements are dynamic arrays of dynamic arrays. The dictionary is made of couples $<K,V>$, i.e. key-value associations. In this case, the keys are the collection of divisors s identified during Step 1. Each key is associated to a number of corresponding compositions, cleaned from unwanted duplicates and redundant spellings at Step 3. The adoption of nested dynamic data structures for values comes from the fact that each s_i potentially presents a different number of compositions $c_{i,j}$, and compositions themselves are made of a variable number of addends.

Such an algorithm can be implemented in different programming languages. An implementation based on HTML5 and JavaScript will be described in Section 8 and has been made publicly available.

6. COMPUTATION AND REPRESENTATION OF MESSIAEN'S MODES

In this section, we will apply the proposed algorithm to the well-known domain discussed in Messiaen's works, namely modes of limited transposition in a 12-EDO. Table 1 illustrates the results obtained through the algorithm, which perfectly fit those described by Messiaen in his theoretical works. The corresponding ring diagrams are shown in Figure 5.

From the contents of the data structure it is possible to reconstruct the complete subdivision of the global interval. For any value of s, it is necessary to replicate the interval

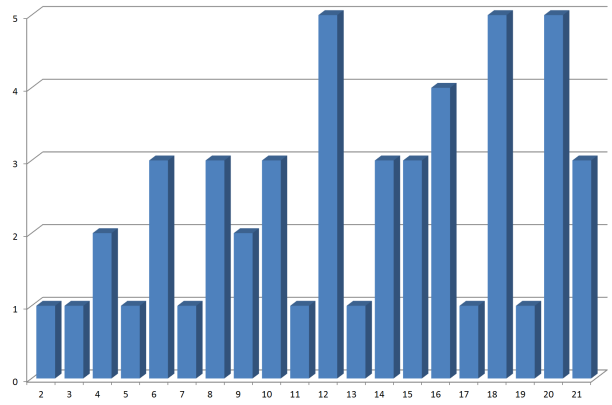


Figure 6. Divisors available in n-TET for $n \in [2..21]$.

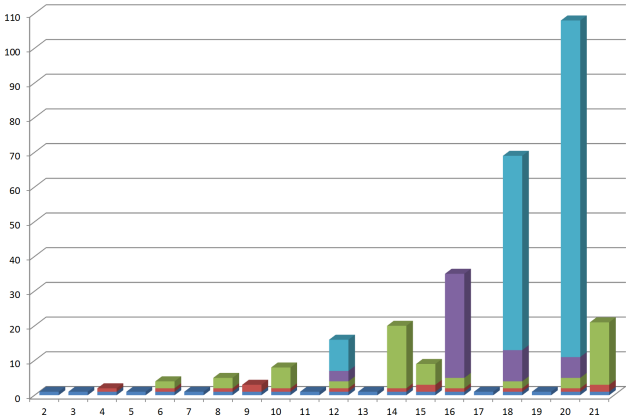


Figure 7. Spellings of the divisors in n-TET for $n \in [2..21]$.

pattern k times, where $k = n/s = 12/s$. In the following we will adopt the pitch class naming convention.

7. COMPUTATION AND REPRESENTATION OF GENERALIZED MODES

The approach used to validate the algorithm in 12-EDO can be easily extended to any other temperament and interval.

First, it is possible to represent (and listen to) specific temperaments that are relevant in musicology, ethnomusicology and microtonal composition, such as 5-, 7-, 29-, 31-, 41- and 53-EDO. In all these cases, we are choosing a *prime* [3] as the number of steps to divide the global interval. Consequently, in order to obtain a tessellation through repetitive patterns, only 1-step groupings are allowed. Other temperaments, e.g. 24-TET, support multiple modes and mode spellings, since they have many divisors of the original step number and divisors are great enough to allow many compositions.

The main advantage of an algorithmic approach is automatically obtaining modes of limited transpositions even in cases where their definition is difficult to obtain by hand. As shown in Figure 6, the number of divisors available in n-TET for $n \in [2..21]$ belongs to the range [1..5]; but Figure 7 demonstrates that the number of corresponding spellings rapidly grows.

8. WEB PROTOTYPE

The algorithm described above has been implemented in HTML5 and JavaScript, and a Web prototype has been released. The application is available at `http://www.lim.di.unimi.it/messiaen`.

The main goal of this prototype is showing generalized modes both from a graphical and from an audio point of view. Colored ring diagrams are produced on-the-fly depending by user's inputs, and both groupings and internal step aggregations are highlighted through the already mentioned graphical devices: colors selected from a chromatic space and different line strokes. In order to produce a ring diagram, it is sufficient to select one of the step number values (for computational reasons $s \in [2 \ldots 52]$), then one of the available divisors for that number, and finally one of the proposed groupings. Arrow-shaped controls are provided to show different spellings of each group, which virtually correspond to suitable rotations of the diagram.

The audio rendering of generalized modes requires some additional inputs. Specifically, two controls let the user set the start and end frequencies of the global interval to be split. [4] Default values are 220 Hz and 440 Hz, corresponding to a 2:1 ratio, namely to the octave interval. Changing this preset allows to subdivide any interval, thus implementing a first degree of generalization with respect to Messiaen's theories. The resulting frequencies for any pitch in the mode are shown to the side of the ring diagram. A sort of circular keyboard has been implemented: little circles can be mouse-clicked to produce the corresponding frequency, and they have been associated also to the keystrokes listed inside the black circles. Play, stop and BPM controls let the user listen to the selected mode as a perpetual scale.

A screenshot of the interface at the moment of writing is shown in Figure 8.

9. CONCLUSIONS

In this paper we presented a generalized approach to the theoretical work on modes by Olivier Messiaen. An algorithm has been designed and implemented in order to compute all possible groupings and interval patterns coming from a subdivision of a given interval into a given number of steps. Under this perspective, Olivier Messiaen's modes of limited transposition are one of the possible instances, as well as Nicholas Mercator researches on 53-EDO and other extensions of Pythagorean tuning. This work may present multiple implications, ranging from music performance to microtonal music theory, from tuning practice to composition.

As regards future work, the prototype will be improved

[3] A *prime* is a natural number greater than 1 that has no positive divisors other than 1 and itself.

[4] Please note that timbre is also relevant to the dissonance levels for intervals within different scales [13]. As a consequence, an additional control to choose among different timbres would be desirable.

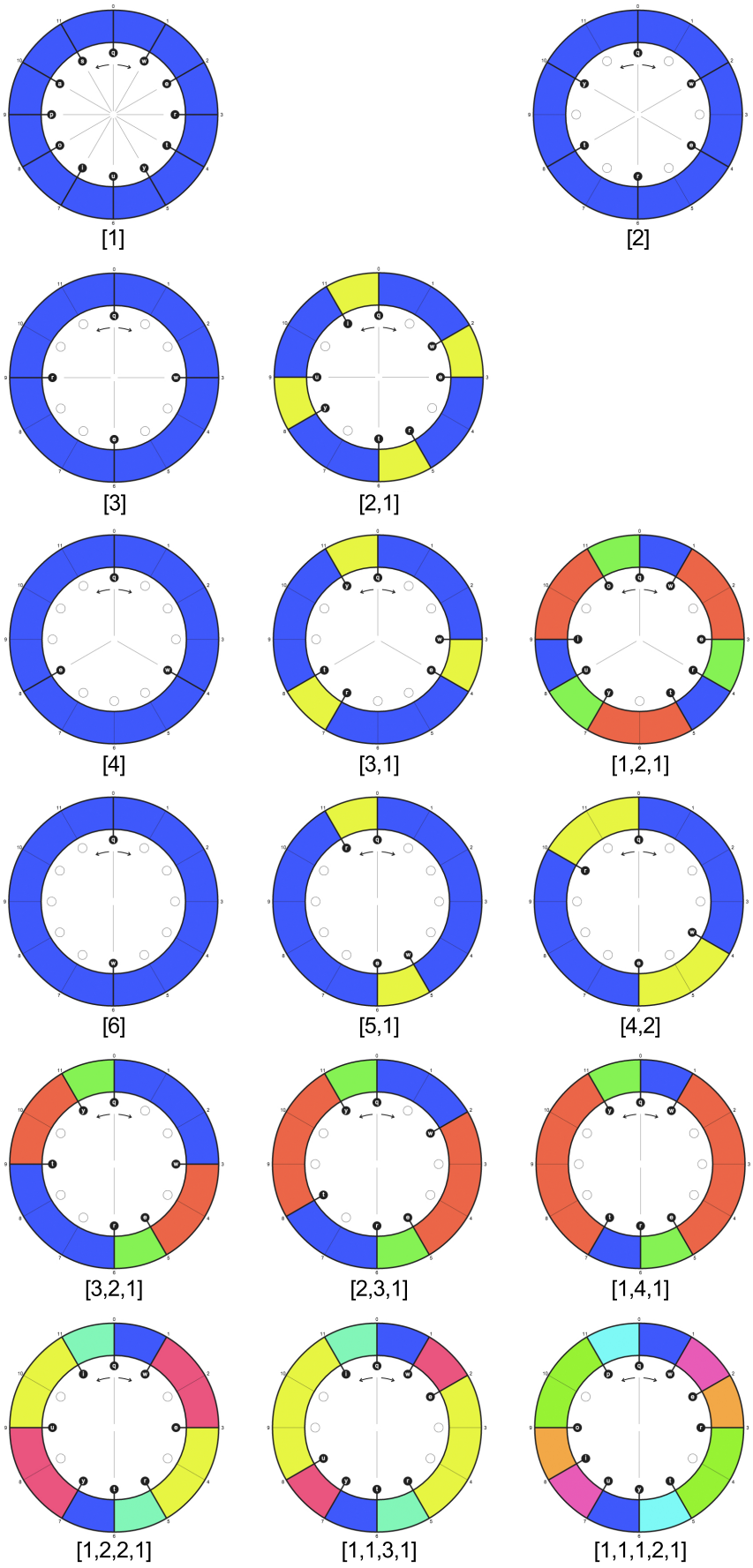


Figure 5. Ring diagrams for Messiaen's modes: *Row i. (left)* – Modes with 1 transposition; *Row i. (right)* – Modes with 2 transpositions; *Row ii.* – Modes with 3 transpositions; *Row iii.* – Modes with 4 transpositions; *Rows iv-vi.* – Modes with 6 transpositions.

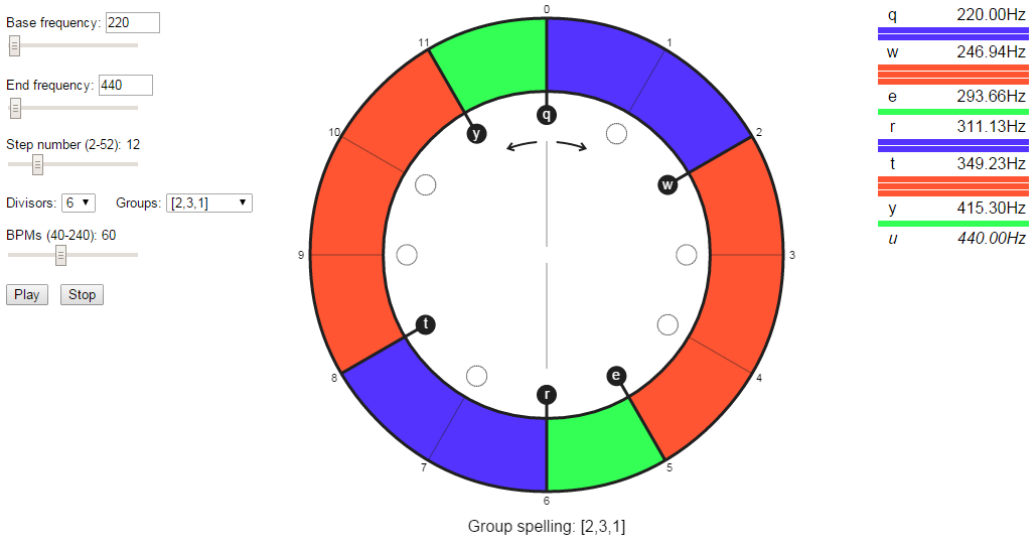


Figure 8. A Web interface to compute generalized Messiaen's modes.

by implementing controls of the timbre. Besides, the project will include a Max/MSP and a PureData porting, so that user-defined timbre generators will be able to interface with the scales defined by the algorithm.

10. REFERENCES

[1] P. Griffiths, *Olivier Messiaen and the music of time*. Faber & Faber, 2012.

[2] O. Messiaen, *Technique de mon langage musical*. A. Leduc, 1944, vol. 1.

[3] J. M. Barbour, "Irregular systems of temperament," *Journal of the American Musicological Society*, vol. 1, no. 3, pp. 20–26, 1948.

[4] D. Morton, *The traditional music of Thailand*. Univ of California Press, 1976.

[5] W. Surjodiningrat, P. Sudarjana, and A. Susanto, *Tone measurements of outstanding Javanese gamelan in Yogyakarta and Surakarta*. Gadjah Mada University Press, 1993.

[6] H. Bohlen, "13 tone steps in the twelfth," *Acta Acustica united with Acustica*, vol. 87, no. 5, pp. 617–624, 2001.

[7] J. Pierce, "Consonance and scales," in *Music, cognition, and computerized sound*. MIT Press, 1999, pp. 167–185.

[8] E. M. Burns and W. D. Ward, "Intervals, scales, and tuning," *The psychology of music*, vol. 2, pp. 215–264, 1999.

[9] J. M. Barbour, *Tuning and temperament: A historical survey*. East Lansing : Michigan State College Press, 1951.

[10] ISO 16:1975, *Acoustics – Standard tuning frequency (Standard musical pitch)*. ISO, Geneva, Switzerland, 1975.

[11] M. Babbitt, "Twelve-tone invariants as compositional determinants," *Musical Quarterly*, pp. 246–259, 1960.

[12] A. R. Brinkman, *PASCAL Programming for Music Research*. University of Chicago Press, 1990.

[13] W. A. Sethares, *Tuning, timbre, spectrum, scale*. Springer Science & Business Media, 2005.

SYNPY: A PYTHON TOOLKIT FOR SYNCOPATION MODELLING

Chunyang Song
Queen Mary, University of London
dr.chunyang.song@gmail.com

Marcus Pearce
Queen Mary, University of London
marcus.pearce@qmul.ac.uk

Christopher Harte
University of York
christopher.harte@york.ac.uk

ABSTRACT

In this paper we present SynPy, an open-source software toolkit for quantifying syncopation. It is flexible yet easy to use, providing the first comprehensive set of implementations for seven widely known syncopation models using a simple plugin architecture for extensibility. SynPy is able to process multiple bars of music containing arbitrary rhythm patterns and can accept time-signature and tempo changes within a piece. The toolkit can take input from various sources including text annotations and standard MIDI files. Results can also be output to XML and JSON file formats.

This toolkit will be valuable to the computational music analysis community, meeting the needs of a broad range of studies where a quantitative measure of syncopation is required. It facilitates a new degree of comparison for existing syncopation models and also provides a convenient platform for the development and testing of new models.

1. INTRODUCTION

Syncopation is a fundamental feature of rhythm in music and a crucial aspect of musical character in many styles and cultures. Having comprehensive models to capture syncopation perception allows us to better understand the broader aspects of music perception. Over the last thirty years, several modelling approaches for syncopation have been developed and widely used in studies in multiple disciplines [1–8]. To date, formal investigations on the links between syncopation and music perception subjects such as meter induction [9, 10], emotion [8], groove [11, 12] and neurophysiological responses [13, 14], have largely relied on quantitative measures of syncopation. However, until now there has not been a comprehensive reference implementation of the different algorithms available to facilitate quantifying syncopation.

In [15], Song provides a consolidated mathematical framework and in-depth review of seven widely used syncopation models: Longuet-Higgins and Lee [1], Pressing [2, 16], Toussaint's Metric Complexity [3], Sioros and Guedes [4, 17], Keith [5], Toussaint's off-beatness measure [6] and Gómez et al.'s Weighted Note-to-Beat Distance [7]. With the exception of Sioros and Guedes' model, code for which was open-sourced as part of the Kinetic project [18], reference code for the models has not previously been publically available. Based on this mathematical framework, the SynPy toolkit (available from the repository at [19]) provides implementations of these syncopation models in the Python programming language.

The toolkit not only provides the first open-source implementation of these models in a unified framework but also allows convenient data input from standard MIDI files and text-based rhythm annotations. Multiple bars of music can be processed, reporting syncopation values bar by bar as well as descriptive statistics across a whole piece. Strengths of the toolkit also include easy output to XML and JSON files plus the ability to accept arbitrary rhythm patterns as well as time-signature and tempo changes. In addition, the toolkit defines a common interface for syncopation models, providing a simple plugin architecture for future extensibility.

In Section 2 we introduce mathematical representations of a few key rhythmic concepts that form the basis of the toolkit then briefly review seven syncopation models that have been implemented. In Section 3 we outline the architecture of SynPy, describing input sources, options and usage.

Copyright: ©2015 Chunyang Song 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.

2. BACKGROUND

In this section, to introduce the theory behind the toolkit, we briefly present key aspects of its underlying mathematical framework (described in detail in [15]) and then give a short overview of each of the implemented syncopation models.

2.1 Time-span

The term *time-span* has been defined as the period between two points in time, including all time points in between [20]. To represent a given rhythm, we must specify the time-span within which it occurs by defining a reference time origin t_{org} and end time t_{end}, the total duration t_{span} of which is $t_{\text{span}} = t_{\text{end}} - t_{\text{org}}$ (Figure 1).

For the SynPy toolkit, we use *ticks* as as the basic time unit as opposed to seconds (in keeping with the representation used for standard MIDI files) where the rate is given in *Ticks Per Quarter-note* (TPQ). The TPQ rate that is chosen is arbitrary so long as the start time and duration of all notes in a rhythm pattern can be represented as integer tick values. As Figure 2 demonstrates, the *Son* clave rhythm pattern could be correctly represented both at 8 and 4 TPQ but not at 2 TPQ because the pattern contains a note that starts on the fourth 16th-note position of the bar.

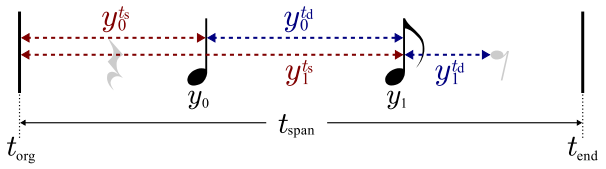


Figure 1. An example note sequence. Two note events y_0 and y_1 occur in the time-span between time origin t_{org} and end time t_{end}. The time-span duration t_{span} is three quarter-note periods. The rests at the start and end of the bar are not explicitly represented as objects in their own right here but as periods where no notes sound.

2.2 Note and velocity sequences

A single, *note* event y occurring in a time-span may be described by the tuple $(t_{\text{s}}, t_{\text{d}}, \nu)$ as shown in Figure 1, where t_{s} represents start or *onset* time relative to t_{org}, t_{d} represents note duration in the same units and ν represents the note *velocity* (i.e. the dynamic; how loud or accented the event is relative to others), where $\nu > 0$.

This allows us to represent an arbitrary rhythm as a *note sequence* Y, ordered in time

$$Y = \langle y_0, y_1, \cdots, y_{|Y|-1} \rangle \tag{1}$$

If TPQ is set to 4, an example note sequence representing the clave rhythm in Figure 2 could be:

$$Y = \langle (0, 3, \mathbf{2}), (3, 1, 1), (6, 2, \mathbf{2}), (10, 2, 1), (12, 4, 1) \rangle, \tag{2}$$

the higher velocity values of the first and third note tuples (in bold) showing that these are accented notes in this example.

An alternative representation of a rhythm is the *velocity sequence*. This is a sequence of values representing equally spaced points in a time-span; each value corresponding to the normalised velocity of a note onset if one is present or zero otherwise. The velocity sequence for the note sequence in Equation 2 can therefore be represented as

$$V = \langle 1, 0, 0, 0.5, 0, 0, 1, 0, 0, 0, 0.5, 0, 0.5, 0, 0, 0 \rangle. \tag{3}$$

It should be noted that the conversion between note sequence and velocity sequence is not commutative, because the note duration information is lost in the conversion. As a result, converting from velocity sequence to note sequence, an assumption must be made that note durations equal to the inter-onset-intervals. Converting the velocity sequence in Equation 3 back to a note sequence would therefore yield

$$Y' = \langle (0, 3, 2), (3, \mathbf{3}, 1), (6, 4, 2), (10, 2, 1), (12, \mathbf{4}, 1) \rangle, \tag{4}$$

which has different durations (in bold) for the second and fourth notes compared to the original sequence in Equation 2.

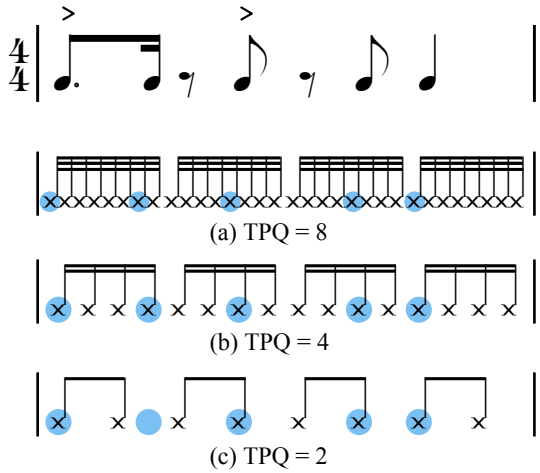


Figure 2. Representation of the *Son* clave rhythm at different Ticks Per Quarter-note (TPQ) resolutions. In (a) and (b) there is a tick for each note of the rhythm pattern thus all the sounded notes are captured (highlighted by the blue circles). However, in (c) where TPQ is 2, the second note of the pattern cannot be represented; the minimum resolution in this case is 4 TPQ.

2.3 Metrical structure and time-signature

Isochronous-meter is formed with a multi-level hierarchical metrical structure [20, 21]. The metrical hierarchy may be described with a *subdivision sequence* $\langle \lambda_0, \lambda_1, ..., \lambda_{L_{\text{max}}} \rangle$ such that in each metrical level L, the value λ_L specifies how nodes in the level above (i.e. $L - 1$) should be split to produce the current level (see Figure 3). Any time-signature can be described by specifying a subdivision sequence and the metrical level that represents the beat.

Events at different metrical positions vary in perceptual salience or *metrical weight* [22]. These weights may be represented as a *weight sequence* $W = \langle w_0, w_1, ...w_{L_{\text{max}}} \rangle$. The prevailing hypothesis for the assignment of weights in the metrical hierarchy is that a time point that exists in both the current metrical level and the level above is said to have a *strong* weight compared to time points that are not also present in the level above [20]. The hierarchy of weights and subdivisions forms a key component in the prediction value calculation for many syncopation models. The choice of values for the weights in W can vary between different models but the assignment of weights to nodes at a given level in the hierarchy, as described in [20], is common to all.

2.4 Syncopation models

In this section we briefly review each implemented syncopation model, discussing their general hypothesis and giving a flavour of their mechanism. It is not possible to go into the full details of each implementation here but a thorough review of the models is given in chapter 3 of [15]. To help compare the capabilities of different models, we also give an overview of the musical features each one captures in Table 1.

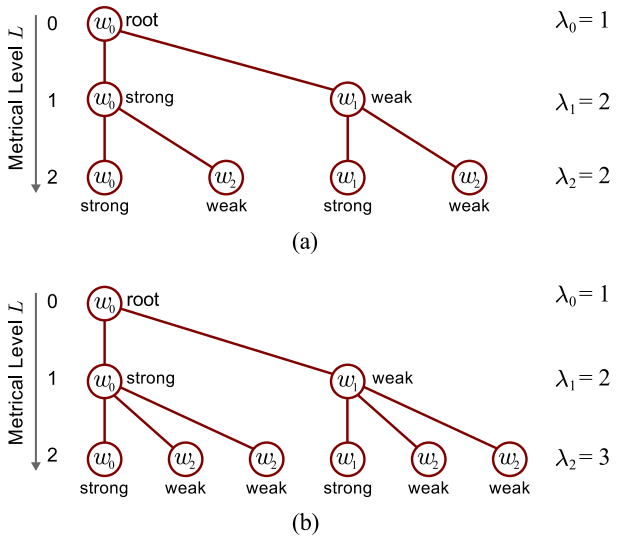


Figure 3. Metrical hierarchies for bars two time-signatures: (a) A simple-duple hierarchy dividing the bar into two groups of two (as with a 4/4 time-signature); (b) A compound-duple hierarchy dividing a bar into two beats, each of which is subdivided by three (e.g. 6/8 time-signature).

Property	LHL	PRS	TMC	SG	KTH	TOB	WNBD
Onset	✓	✓	✓	✓	✓	✓	✓
Duration					✓		✓
Dynamics				✓			
Mono	✓	✓	✓	✓	✓	✓	✓
Poly					✓	✓	✓
Duple	✓	✓	✓	✓	✓	✓	✓
Triple	✓	✓	✓	✓		✓	✓

Table 1. Musical properties captured by the different syncopation models. All models use note onsets, but only two use note duration rather than inter-onset intervals. Only SG takes dynamics (i.e. variation in note velocity) into account. All models handle monorhythms but the four models based on hierarchical decomposition of rhythm patterns are unable to handle polyrhythmic patterns. All models can process both duple and triple meters with the exception of KTH that can only process duple.

2.4.1 Longuet-Higgins and Lee 1984 (LHL)

Longuet-Higgins and Lee's model [1] decomposes rhythm patterns into a tree structure as described in Section 2.3 assigning metrical weights $w_L = -L$ i.e. $W = \langle 0, -1, -2, ... \rangle$. The hypothesis of this model is that a syncopation occurs when a rest (R) in one metrical position follows a note (N) in a weaker position. Where such a note-rest pair occurs, the difference in their metrical weights is taken as a local syncopation score. Summing the local scores produces the syncopation prediction for the whole rhythm sequence.

2.4.2 Pressing 1997 (PRS)

Pressing's cognitive complexity model [2, 16] specifies six prototype velocity sequences and ranks them in terms of *cognitive cost*. For example, the lowest cost is the *null* prototype for rhythms that contain either a single rest or note; a higher cost is given to the *filled* prototype that has a note in every position of the sequence e.g. $\langle 1, 1, 1, 1 \rangle$. The highest cost is given to the *syncopated* prototype that has a rest in the first (i.e. strongest) metrical position e.g. $\langle 0, 1, 1, 1 \rangle$. The model analyses the cost for the whole rhythm-pattern and for each of its sub-sequences at every metrical level determined by the subdivision factor. The final output is a sum of the costs per level weighted by the number of sub-sequences in each.

2.4.3 Toussaint 2002 'Metric Complexity' (TMC)

Toussaint's metric complexity measure [3] defines the metrical weights as $w_L = L_{\max} - L + 1$, thus stronger metrical positions are associated with higher weights and the weakest position will be $w_{L_{\max}} = 1$. The hypothesis of the model is that the level of syncopation is the difference between the metrical simplicity of the given rhythm (i.e. the sum of the metrical weights for each note) and the maximum possible metrical simplicity for a rhythm containing the same number of notes.

2.4.4 Sioros and Guedes 2011 (SG)

Sioros and Guedes [4, 17] also use metrical hierarchy to determine syncopation. The main hypotheses are that humans try to minimise the syncopation of a particular note relative to its neighbours in each level of the metrical hierarchy, and that syncopations at the beat level are more salient than those that occur in higher or lower metrical levels.

The metrical weights for this model are $w_L = L$ i.e. $W = \langle 0, 1, 2, ... \rangle$. The syncopation for a note is a function of its velocity, its position in the hierarchy and the weights of the previous and next notes in the rhythm sequence.

2.4.5 Keith 1991 (KTH)

Keith's model [5] defines two types of syncopated events: a *hesitation*, where a note ends off the beat (assigned a value of 1) and *anticipation*, where a note begins off the beat (assigned a value of 2). Where a note exhibits both a hesitation and an anticipation, a *syncopation* is said to occur and the respective values are summed to give a total of 3. The start and end time are considered off-beat if they are not divisible by the nearest power of two less than the note duration.

2.4.6 Toussaint 2005 'Off-Beatness' (TOB)

The off-beatness measure [6] is a geometric model that treats the time-span of a rhythm sequence as a T-unit cycle. The hypothesis, as applied to syncopation, is that syncopated events are those that occur in 'off-beat' positions in the cycle; the definition of *off-beatness* in this case being any position that does not fall on a regular subdivision of the cycle length T, thus the model is unable to measure cycles where T is 1 or prime.

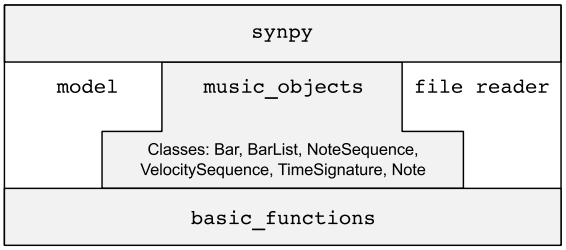


Figure 4. Module hierarchy in the SynPy toolkit: the top-level module provides a simple interface for the user to test different syncopation models. Musical constructs such as bars, velocity and note sequences, notes, and time-signatures are defined in the 'music objects' module; support for common procedures such as sequence concatenation and subdivision is provided in 'basic functions'. Models and file reading components can be chosen as required by the user.

2.4.7 Gómez 2005 'Weighted Note-to-Beat Distance' (WNBD)

The WNBD model of Gómez et al. [7] defines note events that start in between beats in the notated meter to be 'off-beat' thus leading to syncopation. The syncopation value for a note is inversely related to its distance from the nearest beat and is assigned more weight if the note crosses over the following beat.

3. FRAMEWORK

The architecture of the toolkit is shown in Figure 4. Syncopation values can be calculated for each bar in a given source of rhythm data along with selected statistics over all bars; the user specifies which model to use and supplies any special parameters that are required. Sources of rhythm data can be a bar object or a list of bars (detailed below in Section 3.1) or, alternatively, the name of a file containing music data. Where a model is unable to calculate a value for a given rhythm pattern, a 'None' value is recorded for that bar and the indices of unmeasured bars reported in the output. If no user parameters are supplied, the default parameters specified in the literature for each model are used. Output can optionally be saved directly to XML or JSON files. An example of usage in the Python interpreter is shown in Figure 5.

3.1 Music objects

The 'music objects' module provides classes to represent the musical constructs described in Section 2. A `Bar` object holds the rhythm information for a single bar of music along with its associated time-signature and optional tempo and TPQ values (see Section 2.1). `Bar` objects may be initialised with either a note sequence or velocity sequence and can be chained together in the form of a doubly-linked `BarList` allowing syncopation models to access next and previous bars where appropriate (several models [1, 2, 5, 7] require knowledge of the contents of previous and/or next bars in order to calculate the syncopation

```
>>>from synpy import *
>>>import synpy.PRS as model
>>>calculate_syncopation(model, "clave.rhy",
   outfile="clave.xml")
{'bars_with_valid_output': [0, 1],
 'mean_syncopation_per_bar': 8.625,
 'model_name': 'PRS',
 'number_of_bars': 2,
 'number_of_bars_not_measured': 0,
 'source': 'clave.rhy',
 'summed_syncopation': 17.25,
 'syncopation_by_bar': [8.625, 8.625]}
```

Figure 5. To use the toolkit, the top level `synpy` module is imported along with a model (in this example Pressing [2]). Calling `calculate_syncopation()` then gives the syncopation results as shown, also writing output to an XML file. Output file names and extra parameters for a model are added as optional arguments as required by the user.

```
T{4/4} # time-signature
TPQ{4} # ticks per quarternote
# Bar 1
Y{(0,3,2),(3,1,1),(6,2,2),(10,2,1),(12,4,1)}
# Bar 2
V{1,0,0,0.5,0,0,1,0,0,0,0.5,0,0.5,0,0,0}
```

Figure 6. Example rhythm annotation file `clave.rhy` containing two bars of the Son Clave rhythm as discussed Section 2. The first bar is expressed as a note sequence with resolution of four ticks per quarter-note; the second is the same rhythm expressed as a velocity sequence.

of the current bar). The note sequence and velocity sequence classes are direct implementations of the sequences described in Section 2.2. Common low-level procedures such as sequence concatenation and subdivision are provided in 'basic functions'.

3.2 File Input

Two file reader modules are currently provided: one for for reading plain text rhythm annotation (`.rhy`) files and one for reading standard MIDI files (`.mid`). These modules open their respective file types and return a `BarList` object ready for processing.

Our `.rhy` annotation format is a light text syntax for describing rhythm patterns directly in terms of note and velocity sequences (see Figure 6). The full syntax specification is given in Backus Naur Form on the toolkit repository [19].

The MIDI file reader can open type 0 and type 1 standard MIDI files and select a given track to read rhythm from. Notes with zero delta time between them (i.e. chords) are treated as the same event for the purposes of creating note sequences from the MIDI stream. Time-signature and tempo events encoded in the MIDI stream are assumed to correctly describe those parameters of the recorded music so it is recommended that the user avoids incorrectly anno-

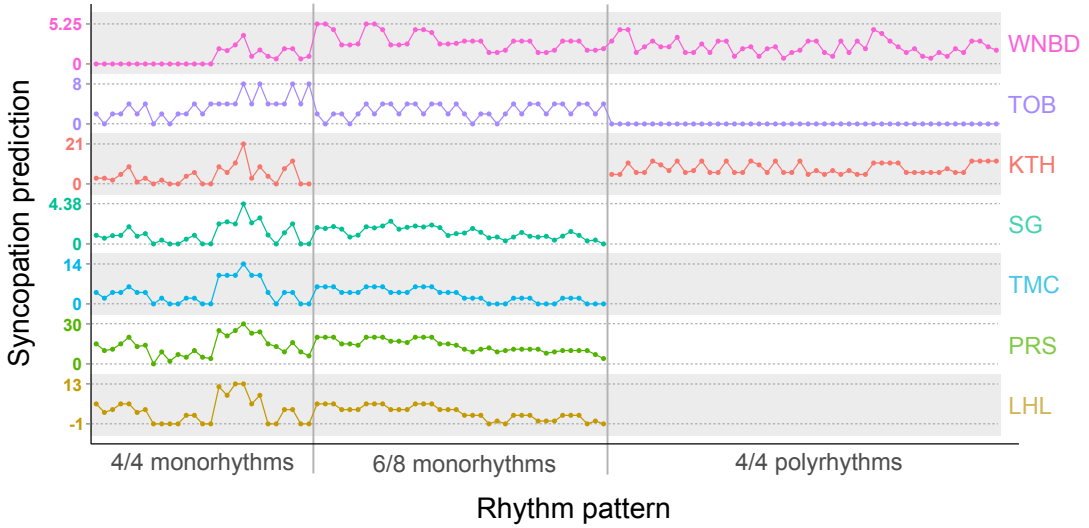


Figure 7. Syncopation predictions of the seven models in the toolkit for the syncopation dataset from [15]. For each model, the absolute range of prediction values is shown across all rhythm patterns in the dataset; ranges differing between models due to their different mechanisms. Within each rhythm category, the rhythm patterns are arranged by tatum-rate (i.e. quarter-note rate then eighth-note rate) then in alphabetical order (the data set naming convention uses letters a-l to represent short rhythm components that make up longer patterns). Gaps in model output occur where a particular model is unable to process the specific rhythm category i.e. LHL, PRS, TMC, SG cannot process polyrhythms and KTH can only measure rhythms in duple meters.

tated or unquantised MIDI files.

3.3 Plugin architecture

The system architecture has been designed to allow new models to be added easily. Models have a common interface, exposing a single function that will return the syncopation value for a bar of music. Optional parameters may be supplied as a Python dictionary if the user wishes to specify settings different from the those given in the literature for a specific model.

4. SYNCOPATION DATASET

The major outcome of the SynPy toolkit is to provide prediction of the level of syncopation of any rhythm pattern that can be measured by a given model. As a demonstration, we apply all seven syncopation models on the rhythms patterns used as stimuli for the syncopation perceptual dataset from [15, 23]. This dataset includes 27 monorhythms in 4/4 meter, 36 monorhythms in 6/8 and 48 polyrhythms in 4/4; altogether forming a set of 111 rhythm patterns.

Figure 7 plots the syncopation predictions of individual models for each rhythm. It presents the different ranges of prediction values for each model and shows their capabilities in terms of rhythm categories (refer to Table 1).

5. CONCLUSION

In this paper we have described SynPy, an open-source Python toolkit for calculating syncopation prediction values. We have introduced the theoretical concepts underpinning the toolkit and briefly reviewed the hypothesis and mechanism of the seven implemented models. The architecture of the toolkit has been introduced in Section 3 and an example of command line usage shown demonstrating ease of use. We have presented the syncopation predictions calculated by SynPy for the dataset from [15], providing an overall visualisation of the prediction ranges and capabilities of each individual model.

The SynPy toolkit possesses a number of merits, including the ability to process arbitrary rhythm patterns, convenient input from different sources of music data including standard MIDI files and text annotations, and output to XML and JSON files for further data analysis. It will be a valuable tool for many researchers in the computational music analysis community. It will be particularly useful to those who study syncopation models because it enables a level of comparison and testing for new models that was hitherto unavailable. The plugin architecture of the toolkit allows new models to be added easily in the future and open-source hosting in a repository on the sound-software.ac.uk servers ensures long term sustainability of the project.

Acknowledgments

This work was funded by the UK Engineering and Physical Sciences Research Council as part of the Soundsoftware Project based in the Centre for Digital Music at Queen Mary, University of London.

6. REFERENCES

[1] H. C. Longuet-Higgins and C. S. Lee, "The rhythmic interpretation of monophonic music," *Music Perception*, vol. 1, no. 4, pp. 424–441, 1984.

[2] J. Pressing, "Cognitive complexity and the structure of musical patterns," in *Proceedings of the 4th Conference of the Australian Cognitive Science Society*, 1997.

[3] G. T. Toussaint, "A mathematical analysis of african, brazilian, and cuban clave rhythms," in *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, 2002, pp. 157–168.

[4] G. Sioros and C. Guedes, "Complexity driven recombination of midi loops," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, 2011, pp. 381–386.

[5] M. Keith, *From Polychords to Pólya: Adventures in Music Combinatiorics*. Vinculum Press, 1991.

[6] G. T. Toussaint, "Mathematical features for recognizing preference in sub-saharan african traditional rhythm timelines," in *3rd International Conference on Advances in Pattern Recognition*, 2005, pp. 18–27.

[7] F. Gómez, A. Melvin, D. Rappaport, and G. T. Toussaint, "Mathematical measures of syncopation," in *BRIDGES: Mathematical Connections in Art, Music and Science*, 2005, pp. 73–84.

[8] P. E. Keller and E. Schubert, "Cognitive and affective judgements of syncopated musical themes," *Advances in Cognitive Psychology*, vol. 7, pp. 142–156, 2011.

[9] D.-J. Povel and P. Essens, "Perception of temporal patterns," *Music Perception*, vol. 2, no. 4, pp. 411–440, 1985.

[10] W. T. Fitch and A. J. Rosenfeld, "Perception and production of syncopated rhythms," *Music Perception*, vol. 25, no. 1, pp. 43–58, 2007.

[11] G. Madison, G. Sioros, M. Davis, M. Miron, D. Cocharro, and F. Gouyon, "Adding syncopation to simple melodies increases the perception of groove," in *Proceedings of: Conference of Society for Music Perception and Cognition*, 2013.

[12] M. A. G. Witek, E. F. Clarke, M. Wallentin, M. L. Kringelbach, and P. Vuust, "Syncopation, body-movement and pleasure in groove music," *PloS ONE*, vol. 9, no. 4, p. e94446, 2014.

[13] I. Winkler, G. P. Háden, O. Ladinig, I. Sziller, and H. Honing, "Newborn infants detect the beat in music," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 7, pp. 2468–2471, 2009.

[14] P. Vuust, M. Wallentin, L. Ostergaard, and A. Roepstorff, "Tapping polyrhythms in music activates language areas," *Neuroscience Letters*, vol. 494, pp. 211–216, 2011.

[15] C. Song, "Syncopation: Unifying music theory and preception," Ph.D. dissertation, School of Electronic Engineering and Computer Science, Queen Mary, University of London, 2015.

[16] J. Pressing and P. Lawrence, "Transcribe: a comprehensive autotranscription program." in *Proceedings of the 1993 International Computer Music Conference*, 1993, pp. 343–345.

[17] G. Sioros, A. Holzapfel, and C. Guedes, "On measuring syncopation to drive an interactive music system," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012, pp. 283–288.

[18] G. Sioros, "Kinetic. gestural controller-driven, adaptive, and dynamic music composition systems," http://smc.inescporto.pt/kinetic/?page_id=9, 2011.

[19] C. Song, C. Harte, and M. Pearce, "Synpy toolkit and syncopation perceptual dataset," https://code.soundsoftware.ac.uk/projects/syncopation-dataset, 2014.

[20] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. Cambridge, Mass: MIT Press, 1983.

[21] J. London, *Hearing in Time: Psychological Aspects of Musical Meter*. Oxford University Press, 2004.

[22] C. Palmer and C. L. Krumhansl, "Mental representations for musical meter," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 16, no. 4, pp. 728–741, 1990.

[23] C. Song, A. J. Simpson, C. A. Harte, M. T. Pearce, and M. B. Sandler, "Syncopation and the score," *PloS ONE*, vol. 8, no. 9, p. e74692. doi:10.1371/journal.pone.0074692, 2013.

MEPHISTO: A Source to Source Transpiler from Pure Data to Faust

Abdullah Onur Demir and Hüseyin Hacıhabiboğlu
Graduate School of Informatics
Middle East Technical University (METU)
Çankaya, Ankara, Turkey, TR-06800
aaonurdemir@gmail.com, hhuseyin@metu.edu.tr

ABSTRACT

This paper introduces Mephisto, a transpiler for converting sound patches designed using the graphical computer music environment Pure Data to the functional DSP programming language Faust. Faust itself compiles into highly-optimized C++ code. The aim of the proposed transpiler is to enable creating highly optimized C++ code embeddable in games or other interactive media for sound designers, musicians and sound engineers using PureData in their work flows and to reduce the prototype-to-product delay. Mephisto's internal structure, conventions, limitations and performance are presented and discussed.

1. INTRODUCTION

Sound synthesis is crucially important not only for electro-acoustic music but also for games and virtual reality applications. High quality audio has a decisive role while creating realistic environments and evoking interest in user experience in video games [1]. There are two common techniques used to create high quality sounds in games. As a first technique, prerecorded clips, also known as Foley sounds, are used. They are the same as sampling and can be modified heavily by processing their samples. Although prerecorded clips provide perfect realism and low computational cost, memory footprint becomes the main bottleneck. Prerecorded clips should reside in memory since the I/O latency of the disk is unacceptable. Besides, loading every sound in physical memory is not a very good solution because of the fact that typically a limited amount of hardware resources is allocated for sounds in games. Moreover, it is particularly hard if not impossible to record sounds such as jet engines, gunshots, rain or wind due to physical and practical reasons.

The second alternative is based on the concept of parametric sound synthesis with which impressive results can be obtained by algorithmic means. Such algorithms allow the model-based generation of hard-to-record sounds by simple signal generators and signal processing methods yet provide convincing results. This approach extends the sound designer's palette by providing her with the means to construct and control virtual sound objects. Synthesized sounds are computer programs that can be executed and parametrically adjusted in real time [2].

Copyright: ©2015 Abdullah Onur Demir 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.

1.1 Pure Data

Pure Data (Pd) [3] is a popular *data flow* programming language with which composers, performers and developers can synthesize sounds without writing code but by using graphical objects and connections between them. Pd does real time computations using a Max-like message interpreter and scheduler and operates on vector samples in order to minimize the interpretation overhead and to satisfy the needs of the real time audio applications. However, sample level computations have to be carried out by using external plug-ins or primitives. Hence, Pd programs depend on a run-time environment. As a result, although not impossible, embedding interpreted Pd programs in games or other systems is generally difficult and inefficient in comparison with code written directly in compiled languages like C or C++ [4].

1.2 Faust

Faust (Functional AUdio STreams) is a functional, *block diagram* programming language designed and used specifically for processing digital signals in real-time [5]. It can be used to create high-performance audio applications and plug-ins. While Faust is well-suited for signal processing, it lacks sufficiently elaborate control mechanisms offering only basic user interface elements like buttons, sliders and number boxes. Faust is designed 1) to be highly expressive, 2) to have clean mathematical semantics, and 3) to be highly efficient [6, 7].

1.3 Motivation

Data flow languages such as Pure Data or Max are popular since programming audio graphically is much easier than writing code. Besides, such languages allow changing parameters and observing the results on-the-fly. However, a particular problem with Pd is that the developed algorithms are difficult to integrate in the final product due to the reduced performance [1], necessity to bundle the run-time environment with the final product (e.g. a game or a mobile app) and incompatibility with existing development frameworks.

[1] PureData is reported to be roughly three orders of magnitude slower than its C equivalent for multiplying floating point numbers [8].

In contrast, programs written in Faust can be directly translated to optimized C++ code, can be embedded into a final product in a more straightforward way and also provide a higher performance. However, the main drawback of the Faust language is that the programmer has to learn functional programming concepts, syntax and semantics of Faust and mathematical descriptions of signals to be able to code in Faust.

This paper presents Mephisto, a source-to-source transpiler from Pure Data to Faust which aims to bridge the gap between the two programming languages and to facilitate easier design and integration of audio algorithms in relevant software development processes.

The paper is organized as follows. Section 2 presents the works which bring solutions to the problems in a similar way with Mephisto. Section 3 talks about how Mephisto is designed and how it can be used to generate Faust code from Pure Data patches. Section 4 talks about the conventions that Mephisto uses along with its limitations. The performance of Mephisto/Faust generated code in comparison with Pd itself and libpd is discussed in Section 5. Section 6 talks about possible directions for Mephisto and Section 7 concludes the paper.

2. RELATED WORK

Two other similar tools to what is presented in this paper exist. These are PUre DAta Compiler (*PuDaC*) and *libpd*. A short overview of these tools are given in this section.

2.1 PuDaC

PuDaC (PUre DAta Compiler) is a compiler created in order to address the performance issues pertaining to Pd [8]. PuDaC considers data as if it consists of two parts: High-bandwidth (audio) and low-bandwidth (control) signals. The equivalent of each Pd object in the patch is transformed into C language. The connections are translated as function calls. As a result, the Pd patch is transformed into a C program which can run efficiently on embedded systems with limited computational resources. However, the resultant C program is not optimized specifically for audio processing in contrast to C++ code that can be generated by Faust and is thus suboptimal in terms of performance.

2.2 libpd

libpd [9] is a free and open source software library which enables the usage of *PureData* almost everywhere from embedded devices to phones and computers.

libpd is not a substitute of Pd but it is the embedded version of Pd itself. Since it is embeddable, it can run on any hardware that can run native code. Hence, Pd patches can be incorporated within games, Android or iOS applications, or programs written in C running on embedded Linux systems including microcontroller boards like Intel Galileo. *libpd* helps Pd run in the musical application or the game as a compiled program.

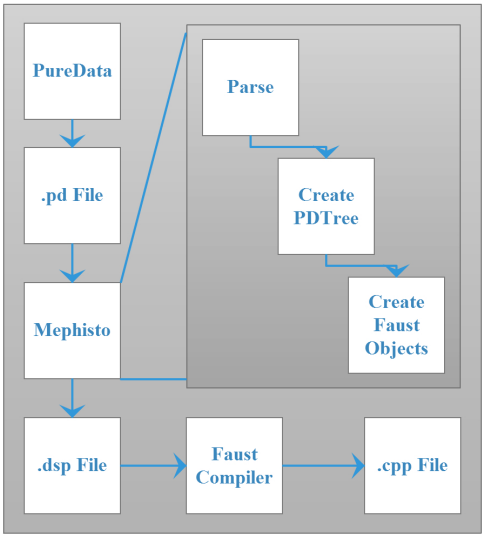


Figure 1: Transpiling pipeline of Mephisto

3. MEPHISTO: A SOURCE TO SOURCE TRANSPILER

Mephisto is developed in order to enable Pure Data users to incorporate the audio algorithms they design into games and other applications by utilizing the highly optimized C++ code created by Faust. With this motivation, Mephisto transpiles PureData sources into Faust sources and the creation of optimized C++ code is then left to the Faust compiler.

Transpilation process consists of four steps: 1) parsing the Pd source, 2) creating Pd object tree and traversing it, 3) creating Faust source on-the-fly while traversing the tree, and 4) compiling the transpiler-generated Faust code to obtain optimized C++ code. A visual representation is shown in Fig. 1.

3.1 Parsing

ANTLR v4 is a powerful parser generator used to read, process, execute, or translate structured text such as program source code, data, and configuration files [10]. ANTLR v4 also has the capability to carry out lexical analysis, token generation, and parse tree creation. Given these advantages that simplify and integrate the work flow, we implemented the Pure Data parser with ANTLR v4.

Each object positioned on the canvas of a Pd patch is represented as a single row in the source file. Each row contains the definition of the corresponding object and its parameters. By using the format specification of Pd and ANTLR v4, a formal language description, i.e. the *grammar* of Pure Data, was created. This grammar is used by ANTLR for generating a standalone Java program that processes Pd source files and builds a data structure representing the input (i.e. *parse tree*). The row nodes of an example parse tree are shown in Fig. 2. Additionally, ANTLR v4 automatically generates *tree walkers* (listeners) that can be used to visit the nodes of the parse tree to create a Pd object tree. A parse tree listener interface consists of simple event listeners triggered by the built-

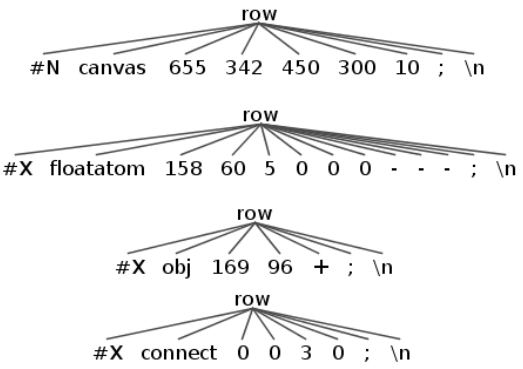


Figure 2: Each row represents a Pd object. Each element in a row represents arguments of that object. All tokens are generated from Pd source in Mephisto's parsing stage

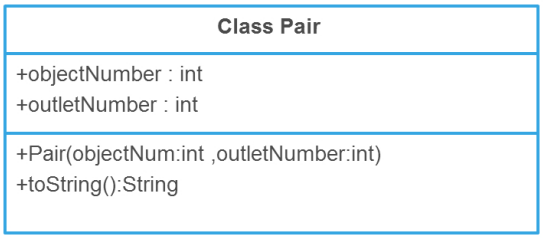


Figure 3: Pair Class representing connections in a Pd patch

in tree walker [10]. ANTLR v4 generates enter and exit methods for each node in the parse tree. Enter event is triggered when tree walker enters a node. Exit event is triggered when it completes traversing the node's children nodes and leaves it. By using these listeners, current node is determined and can be processed based on its context. Details of the file format specification of Pure Data source used by Mephisto can be found in Pd's distribution site [11].

3.2 Creation and the Traversal of Pd Object Tree

In order to define the semantics of Pd objects and the connections between them, we created a data structure to form a tree which will henceforth be called a *PDTree*. The data structure is shown in Figs. 3 and 4

`Pair` class represents an outgoing connection of a Pd object. `objectNumber` represents unique ID of it and `outletNumber` represents which outlet of it belongs to the referred connection.

`PDObject` class represents a Pd object itself. `defaultVal` keeps the default value of the Pd object as statically set in the patch (e.g. the initial frequency of an oscillator). For multiple initialisation arguments, `args` array is used instead of the `defaultVal` attribute. The most important part of this class is the `objectInlets` attribute which is a hash map whose keys are representing its inlets. Each key holds a `Pair` instance. For example, index `0` holding a `Pair` object with the values `objectNumber=2` and

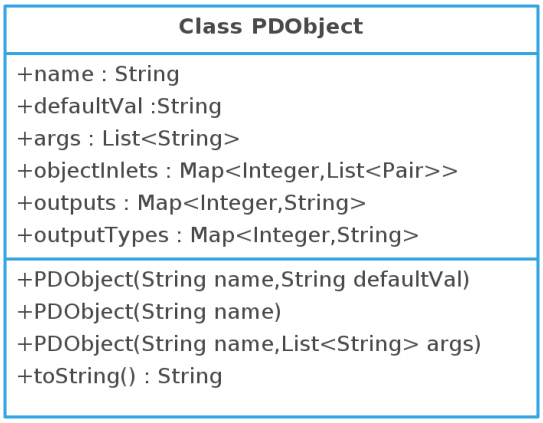


Figure 4: PDObject Class representing Pd objects in a Pd patch

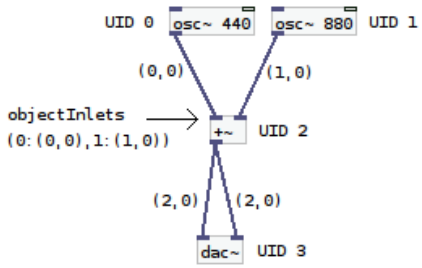


Figure 5: Visualization of usage of Pair Class and PDObject Class instances

`outletNumber=0` means that the first outlet of another Pd object whose `UID` is `2` is connected to the first inlet of the present Pd object. The `outputs` attribute represents the outlets of the Pd object. Each key refers to each outlet of the Pd object mapping the key `i` to outlet `j` and so on. Fig. 5 illustrates the data structure.

While a tree walker walks on the ANTLR v4 generated parse tree, it dispatches listener events. We use listeners automatically generated by ANTLR which are `enterRow` and `exitFile`. `PDObject` instances are created in `enterRow` callbacks which are called with a context argument keeping the children nodes of the row node being processed in the parse tree. Since all arguments and parameters which are children of that node are found in the context, a `PDObject` with a unique ID can easily be created without explicitly traversing its children. After the creation of a `PDObject` instance, it is pushed to a global list in order to be accessed later. Additionally, if `enterRow` callback is called for a connection token, a `Pair` instance is created representing a connection between Pd objects and is inserted within the corresponding `PDObject`'s hash-map.

At the end of the traversal of the parse tree, a `PDTree` is created which has `dac~` object in the Pd patch as its

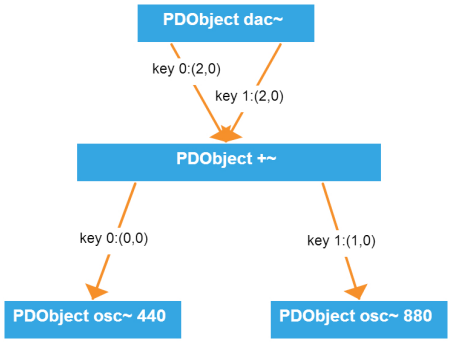


Figure 6: PDTree. The numbers represent traversal order

root. At the end of the tree traversal process, `exitFile` callback function is called and a second traversal is started beginning from the root of the `PDTree` following depth-first search (see Fig. 6).

3.3 Traversal of `PDTree` and creation of Faust Code

The second traversal is started on the `dac~` object by the call of the `exitFile` callback function. At first, the hashmap `objectInlets` of root object, `dac~`, is scanned. The first key is always `0` representing the first inlet of the object. Additionally, the value of the key represents a `Pair` object indicating a connection incident from an outlet of another `PDObject` instance. The traversal is carried out by recursion by a function having the definition, `createObject_setOutput(int objectNumber,int outletNumber)`.
After giving the value(Pair) of the key `0` to this function as `<Pair instance>.objectNumber` and `<Pair instance>.outletNumber` it sets the outputs and `outputTypes` of that object and finally returns the Faust equivalent outlet value of the processed `PDObject` instance. This procedure is applied to each key stored in the `objectInlets` attribute for each `PDObject` instance.

Since `dac~` object in Pure Data represents the sound hardware that will output signals incident to its inlets, it is transpiled to a Faust `process` which forms the entry point of the generated Faust program. The inlets of the `dac~` object are mapped to left and right channels in the generated Faust program. Transpilation process is completed when the tree traversal ends.

3.3.1 createObject_setOutput(int objectNumber,int outletNumber) Function

This function is at the core of Mephisto and is used to create the Faust equivalents of Pd objects. The first argument (`objectNumber`) defines the `UID` of the `PDObject` instance stored in the aforementioned global list. The second argument (`outletNumber`) defines the outlet value to be returned. The function first reads the instance from the list by its `objectNumber`. After this is obtained and the name of the object is checked, the function calculates values incident from other connected objects into its inlets by using the same function recursively. After obtaining all incoming values for each inlet, a Faust function definition is created. Since Faust is a functional programming language, the defined function's name, complete with its arguments, is returned as an output.

3.4 Compiling transpiler-generated Faust code

On a system where Faust is installed, the resulting `.dsp` file can be compiled and a `.cpp` file can be generated. As a result, a well optimized C++ code can be created without writing any lines of code using Mephisto. The generated C++ code can be embedded in a rather straightforward way in any desired software project written using the C++ language.

4. CONVENTIONS AND LIMITATIONS

There are significant differences between Pure Data and Faust. Mephisto uses certain conventions to reconcile these differences. At its present version, Mephisto also has certain limitations. These conventions and limitations are discussed below:

1. Mephisto ignores the "cold inlet" mechanism of Pure Data. Since Faust is a language tailored primarily for processing audio streams, everything is considered as signals and there is no messaging or control mechanism except for elementary user interfaces. Hence, the transpiling process ignores the cold inlet semantic of Pd while translating it into Faust code.

2. Mephisto will not transpile Pd patches unless the patch includes a `dac~` object. Therefore, Pd patch to be transpiled should be formed as a connected tree [12] in which there is a `dac~` object. Since Mephisto transpiles Pd patches by constructing a tree, objects which are not included in this connected tree will be ignored by Mephisto.

3. Mephisto's main aim is to transpile Pd patches focused on signal blocks and control blocks are not covered. However, simple control mechanisms like number box, message, trigger, pack and unpack are provided. In addition, mathematical and logical operators in both the control and the signal blocks in Pd are also implemented in Mephisto. This is done by converting these to signal blocks since both messages and signals in Pd have to be represented as signals in Faust. Hence, all control logic, except for elementary controls, should be implemented separately in C++ (or any other programming language that Faust would support in the future) to incorporate within the code in which the generated C++ code will be embedded. Consider the patch in Fig. 7, for example. The patch synthesizes the standard CCITT dialing tone [13] which is the sound heard just after picking up the phone. It consists of nearly all signal objects except for two, which are messages that control starting and stopping the

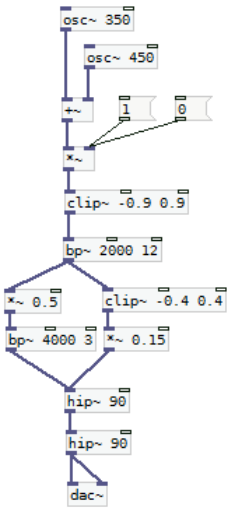


Figure 7: Dialing tone patch synthesizing the standard CCITT dialing tone

generated output. These two message objects can be transpiled into check-boxes in Faust code in order to preserve content integrity. However, these control statements should be properly implemented and connected to a (possibly event-driven) control code in which they will be embedded. Faust code automatically generated by Mephisto for the CCITT patch (Fig. 7) by Mephisto as well as the block diagram representation of the same Faust code are given in the Appendix.

4. Mephisto does not yet support sub-patches and external objects as it is designed to parse only one Pd source. Allowable objects are predefined in the transpiler and others are not recognized. Therefore, Pd projects that include sub-patches or external objects should be flattened to a single Pd patch.

5. Fig. 8a shows the regular convention in Pd to generate a counter. This counter incorporates a `float` object which is not recognized by Mephisto. In addition, the structure of the counter violates the tree structure of `PDTree` since it has a cycle. Since trees are defined as graphs which do not include cycles and that cycles will cause Mephisto to enter infinite recursion, counters cannot be created as a direct translation. In order to alleviate this problem, Mephisto requires that a special Pd patch, `fcounter`, be used which is an abstraction of the patch shown in Fig. 8b.

5. PERFORMANCE

A performance comparison of three different ways to execute algorithms designed in Pure Data is given in this section. Specifically, three different conditions were tested: 1) Pure Data natively running a patch, 2) the same patch executed via the embeddable Pure Data library *libpd* [9] and 3) C++ code generated by Mephisto and



(a) Conventional counter used in most of the Pd patches

(b) Mephisto Fcounter abstraction sub-patch. Used instead of conventional counter

Figure 8: PDObject and Pair classes

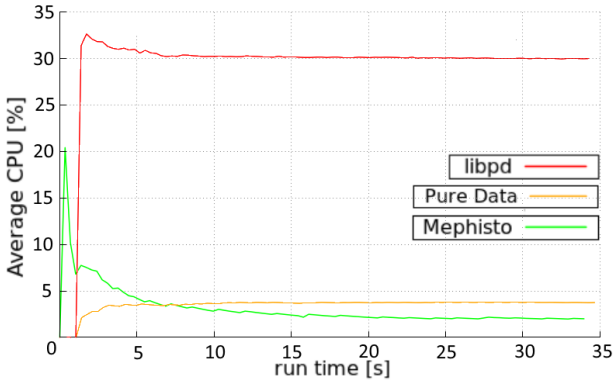


Figure 9: The average CPU utilization plots of the three tested conditions.

Faust, cross-compiled with JACK [14]. In each case the relevant process was isolated and the average CPU utilisation was obtained by the profiler, Audria [15]. The patch that was used in benchmarking these three cases consists of the product of the outputs of two oscillators for a duration of 30 s. The process was tracked for 35 s in each case. At the sampling rate of 44.1 kHz, this amounts to the generation of 2,646,000 floating point numbers and 1,323,000 floating point multiplications in total. Fig. 9 shows the average CPU utilization for each case. It may be observed that the highly optimized code generated by Mephisto and Faust outperforms both the native Pure Data and libpd-based implementations. In the case of Pure Data only, the average CPU utilization starts at zero and plateaus at slightly less than 4%. In the case of libpd, the same process results in around 30% utilization. For C++ code generated via Mephisto by Faust, the average utilization starts at 20% and quickly falls to just above 2%. The peak in the CPU utilisation curve for *Mephisto* between t=0 and t=1 is thought to be caused by external libraries (JACK and Qt) used in compiling the C++ code to create a working program. These results indicate that generating C++ code by Faust via Mephisto is promising in terms of the potential performance gains that it can provide.

6. FUTURE WORK

Mephisto is at a stage that basic sound synthesis algorithms designed in Pure Data can be transpiled into Faust code in order to obtain highly optimized C++ code for use in games and other applications. However, the limitations outlined in the previous section remain to be solved.

A possible solution to cold inlet and messaging problem

can be developed by designating an impulse signal as a bang message. In Faust, the lack of messages means that numbers are treated as signals and there is no gating mechanism as in Pd. The deficiency caused by Faust's lack of such control mechanisms could be filled by creating specialized Faust functions and revising the traversal mechanism.

It is critical that the Pd patch does not include any cyclic connections. However, Mephisto can be further developed so as to allow transpiling Pd patches containing disjoint or cyclic trees [16] with no constraints on the root objects.

In addition, Pd abstractions for Faust code could be written for Mephisto so that it could deduce what the semantic of the abstraction is and does not care about its internal structure. Faust implementation can then be created and added to the source code of Mephisto. The example for `fcounter` can act as a starting point.

The work presented in this paper acts as a proof-of-concept showing that transpilation from Pure Data to Faust is possible and thus the present version of Mephisto does not cover more than a few Pd objects. Any additional object that is added to Mephisto would increase the palette of objects available to the Pd programmers. For this reason, Mephisto will be released with the GPL v3.0 license and will be made open source in a GitHub repository [17] to allow other developers to work on it. Additionally, further examples including fire, jet engine and wind sound adapted from [2] reside in the repository.

7. CONCLUSION

Pure Data is both a Turing complete programming language and a very useful tool for designing audio algorithms from scratch. However, since Pure Data is an interpreted (as opposed to compiled) language, it presents important performance issues. Therefore, once the prototype algorithms are designed, the designer or the developer still has to carry the burden of writing code to integrate the algorithm in a final standalone app such as a game or an audio app typically using another high-level language such as C++. While this approach is feasible, the effort required from the C++ programmer is considerable and a solution that can automatically generate C++ code from Pure Data code has practical benefits. A transpiler from Pure Data to Faust, named Mephisto, is described in this paper that can achieve this. While Mephisto can generate Faust code from Pure Data patches, Faust is used as an intermediate language which already has a compiler that can generate C++ code.

8. ACKNOWLEDGMENTS

This work was supported by the Middle East Technical University Faculty Research Grant, BAP-08-11-2013-057.

9. REFERENCES

[1] D. B. Lloyd, N. Raghuvanshi, and N. K. Govindaraju, "Sound synthesis for impact sounds in video games," in *Proceedings of the Symposium on Interactive 3D Graphics and Games 2011*. ACM, 2011.

[2] A. Farnell, *Designing Sound*. Cambridge, MA, USA: MIT Press, 2010.

[3] M. Puckette, "Pure Data: Another integrated computer music environment," in *Proc. Second Intercollege Comp. Mus. Concerts*, Tachikawa, Japan, 1996, pp. 37–41.

[4] Y. Orlarey, D. Fober, and S. Letz, "FAUST: an efficient functional approach to DSP programming," in *New Computational Paradigms for Computer Music*. Editions Delatour, Paris, France, 2009.

[5] ——, "An algebra for block diagram languages," in *International Computer Music Conference*, Göteborg, Sweden, September 2002.

[6] J. O. Smith III. (2010) Audio signal processing in Faust. [Online]. Available: https://ccrma.stanford.edu/jos/aspf

[7] Y. Orlarey, D. Fober, and S. Letz, "Syntactical and semantical aspects of faust," *Soft Computing*, vol. 8, no. 9, pp. 623–632, 2004.

[8] R. N. Jacobs, "A wireless sensor-based mobile music environment compiled from a graphical language," Ph.D. dissertation, MIT Media Lab, Cambridge, MA, USA, September 2007.

[9] libpd ≫ about. [Online]. Available: http://libpd.cc/about/

[10] T. Parr, *The definitive ANTLR 4 reference*. The Pragmatic Bookshelf, 2012.

[11] (2004, October) Unofficial PD v0.37 fileformat specification. [Online]. Available: http://puredata.info/docs/developer/PdFileFormat

[12] J. A. Bondy and U. S. R. Murty, *Graph theory with applications*. Macmillan London, 1976.

[13] International Telecommunications Union (ITU), *Application of tones and recorded announcements in telephone services*, CCITT Recommendation E.182, 1998.

[14] JACK Audio Connection Kit. [Online]. Available: http://jackaudio.org/

[15] audria - A Utility for Detailed Resource Inspection of Applications. [Online]. Available: https://github.com/scaidermern/audria

[16] M. Nilsson and H. Tanaka, "Cyclic tree traversal," in *Third International Conference on Logic Programming*. Springer, 1986, pp. 593–599.

[17] A. O. Demir. (2015) Mephisto Source Code. [Online]. Available: https://github.com/aonurdemir/Mephisto

APPENDIX

Mephisto-generated Faust Code for the CCITT dialling tone patch

CCITT dialling tone patch was given as an example earlier in the paper in Fig. 7. Faust code automatically generated by Mephisto for this patch is:

```
CCITT.dsp
import("music.lib");
import("math.lib");
import("filter.lib");

osc1=osc(350);
osc2=osc(450);
checkbox3=checkbox("1");
msg3 = checkbox3 * 1;
checkbox4=checkbox("0");
msg4 = checkbox4 * 0;
clip7(s) = if (s < (-0.9), (-0.9) , if ( s>0.9 ,0.9 ,s) );
resonbp8=clip7(((((osc1+osc2))*((msg3)+(msg4))))):resonbp(2000,12,1);
clip10(s) = if (s < (-0.4), (-0.4) , if ( s>0.4 ,0.4 ,s) );
resonbp11=((resonbp8)*0.5):resonbp(4000,3,1);
resonhp13=(resonbp11+((clip10((resonbp8)))*0.15)):highpass(1,90);
resonhp14=(resonhp13):highpass(1,90);
process=resonhp14,resonhp14;
```

Figures below show the different block diagrams generated for the same Pure Data patch. The figures were generated using FaustWorks IDE. The figures follow the tree traversal order as explained in the text, starting with the `dac~` object and following through to the `osc~` objects and the message boxes (emulated using checkboxes in Mephisto-generated Faust code.).

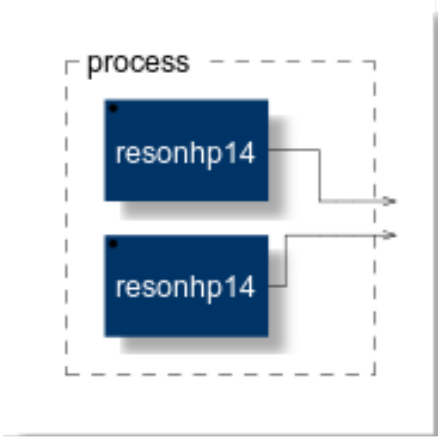


Figure 10: Equivalent of ”dac∼” object

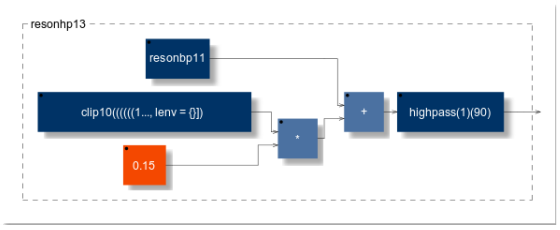


Figure 12: Equivalent of ”hip∼ 90” object

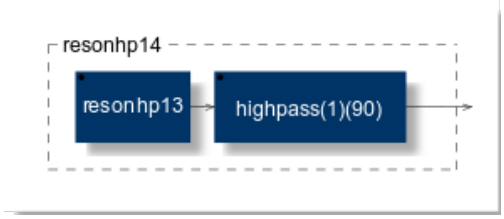


Figure 11: Equivalent of ”hip∼ 90” object

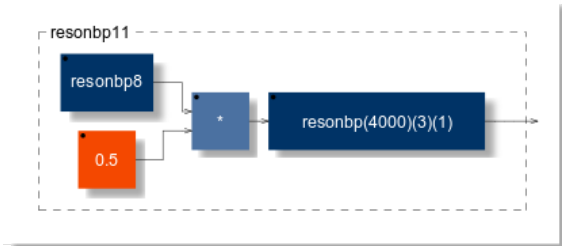


Figure 13: Equivalent of ”bp∼ 400 3” object

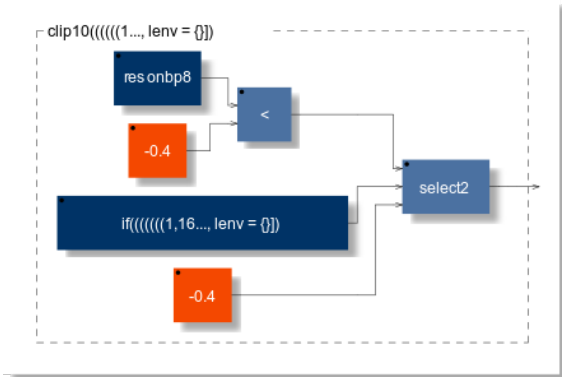


Figure 14: Equivalent of ”clip∼ -0.4 0.4” object

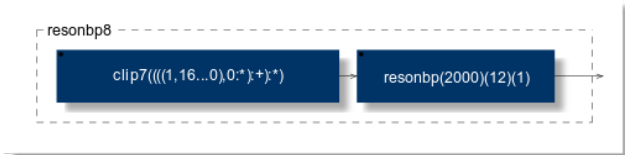


Figure 15: Equivalent of ”bp∼ 2000 12” object

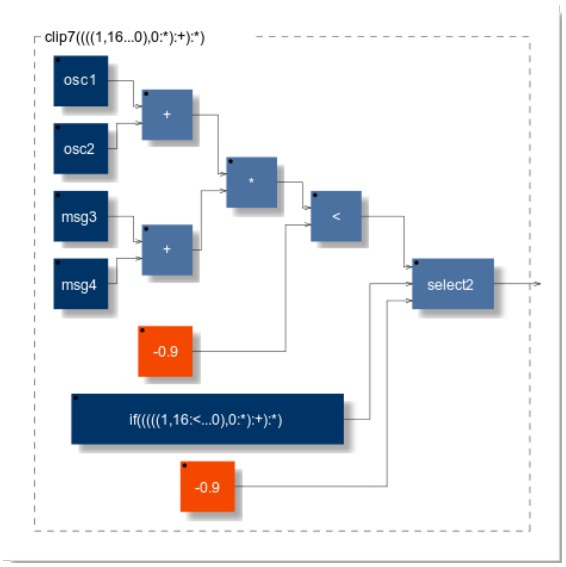


Figure 16: Equivalent of ”clip∼ -0.9 0.9” object

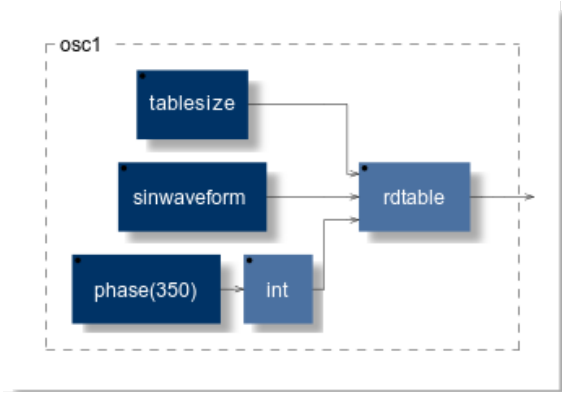


Figure 17: Equivalent of ”osc∼ 350” object

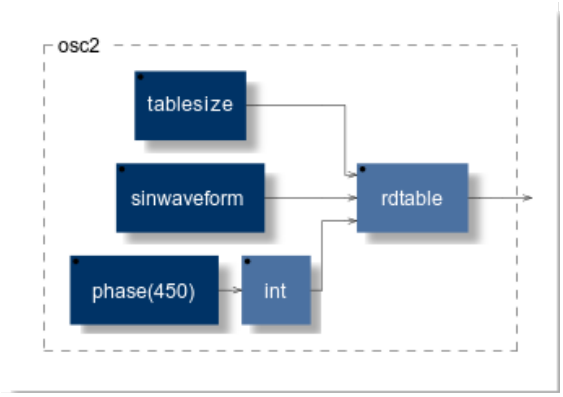


Figure 18: Equivalent of ”osc∼ 450” object

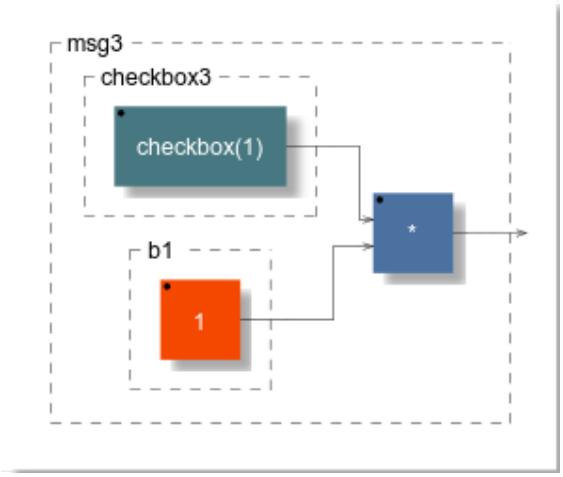


Figure 19: Equivalent of message object ”1”

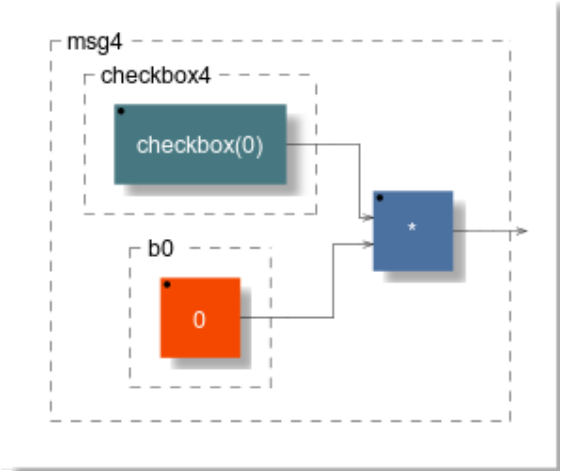


Figure 20: Equivalent of message object ”0”

To "Sketch-a-Scratch"

A. Del Piccolo[1], S. Delle Monache[2], D. Rocchesso[2], S. Papetti[3], and D.A. Mauro[2]
[1]Ca' Foscari University of Venice, Department of Environmental Sciences, Informatics, and Statistics
956199@stud.unive.it
[2]Iuav University of Venice, Department of Architecture and Arts
sdellemonache, roc, dmauro@iuav.it
[3]Zürcher Hochschule der Künste, Institute for Computer Music and Sound Technology
stefano.papetti@zhdk.ch

ABSTRACT

A surface can be harsh and raspy, or smooth and silky, and everything in between. We are used to sense these features with our fingertips as well as with our eyes and ears: the exploration of a surface is a multisensory experience. Tools, too, are often employed in the interaction with surfaces, since they augment our manipulation capabilities. "Sketch-a-Scratch" is a tool for the multisensory exploration and sketching of surface textures. The user's actions drive a physical sound model of real materials' response to interactions such as scraping, rubbing or rolling. Moreover, different input signals can be converted into 2D visual surface profiles, thus enabling to experience them visually, aurally and haptically.

1. INTRODUCTION

In everyday interaction with the environment, we experience surface textures mostly through touch and vision, although audition can contribute to forming multisensory percepts too [1]. Textures can be rubbed with a fingertip, scraped with a nail, or rolled-over with a ball. All of these actions can be described by microscopic contact events occurring between the probe (be it the finger itself, or an object such as a pen) and the explored surface. These events can be simulated by the physical modeling of impact and friction phenomena in order to reproduce in a virtual setting the experience of interacting with a surface texture.

The importance of haptics for conveying a convincing textural experience in virtual and augmented environments has been widely advocated [2], although force-feedback devices are impractical or expensive in many contexts. This explains the emergence of pseudo-haptics [3,4], i.e. the exploitation of multisensory illusions to render forces through alternative sensory channels.

When performed with a tool, the exploration of a surface often produces an audible signal that carries information about surface roughness, hardness, and friction through sound [1]. On the other hand, a sound signal can be interpreted as a surface profile that may be appreciated with other senses. Hence, the qualities of a surface can be imposed by means of synthesized or acquired sound signals.

In addition, co-location of action and feedback is a crucial factor, as it recalls the typical situation of many manual activities that afford the development of expressiveness and virtuosism, such as painting or drawing.

Investigating the multisensory exploration of a virtual surface can provide clues on how the sensory channels integrate in the forming of similar experiences in the real world. Through these channels, different aspects of complex physical phenomena are rendered. While some of these aspects may be impossible or impractical to render accurately in a digital environment, their modeling helps investigating how they might possibly be replaced or imitated by means of other sorts of stimuli. For instance, some haptic sensations, i.e. the lateral forces [5], which are usually not conveyable through easily accessible devices, require alternative solutions in order for them to be sensed.

We present "Sketch-a-Scratch", an experimental tool for multisensory sketching and augmented exploration of surface textures. This apparatus is based on a vibroacoustically augmented graphic tablet and stylus, and on real-time physics-based simulation of contact mechanics, and can render surface textures by means of visual, auditory, and vibratory feedback. Multimodal exploration and modeling of virtual surfaces coexist in that the user can acquire a surface texture from various sources, from still images to drawing, from audio recordings to vibration sensing, and hence experience them through probe-mediated touch, vision, and audition. The tool allows to transform these actions into performative acts, and finally experience and exploit surface qualities across different modalities.

The paper is organized as follows. First we describe the research background of Sketch-a-Scratch, and the motivations behind it, that is the multisensory exploration of virtual surface profiles. Then, we describe the concept and the general architecture of the tool. Further on, the basic framework that we devised for experimentation is illustrated, as well as its hardware and software requirements. We then outline two contexts of use for the tool, namely an artistic performance and a self-contained interactive installation. The different contexts of use trace the path of our investigation on tool-mediated exploration and design of virtual surfaces. We especially elaborate the ex-

Copyright: ©2015 A. Del Piccolo[1], S. Delle Monache[2], D. Rocchesso[2], S. Papetti[3], and D.A. Mauro[2] 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.

perience gathered by the public installation showcase and outline some possible directions for future development of the tool.

2. BACKGROUND

Surfaces can be experienced visually, when they are acquired as pictures, or haptically through a process of scanning.

The haptic sensation of a surface can be provided by actuating either the surface [6] or the probe [7]. An alternative to the mechanical actuation is the use of electricity-induced vibrations in different forms: electrocutaneous [8], electrostatic [9] or electrovibration [10].

Direct touch differs from tool-mediated exploration in that the former gives a spatial, intensive measure of roughness, while the latter carries information about roughness, hardness and friction in the form of a multidimensional signal in the time variable. In the more traditional category of mechanically-induced haptic feedback, the actuation of the interactive surface by means of piezoelectric bending motors, voice coils or solenoids [11] allows for direct touch, yet presents several drawbacks. Due to the flexibility of the surface, the feedback is usually not uniform throughout the active area, especially with large surfaces. Besides, motors are often located on the sides of the surface, which makes it hard to convey the co-location of stimulus (the user's touch) and response (the vibration).

The tool-mediated exploration of a surface is a common practice in many creative processes such as writing and figurative art. When using this paradigm for digital simulation it also presents practical advantages. The actuator for haptic feedback can be incorporated in the probe, making the intensity of the feedback independent from size and geometry of the screen. Moreover, the round, plastic tip of a stylus exerts less friction than the finger tip on the glass surface of a touch screen. Starting from a situation of reduced inherent friction, the implementation can achieve a wider range of levels of friction.

In sound synthesis, contact phenomena taking place at the interface between an object and a surface can be simulated by means of several models. One type of such phenomena is friction, which is based on stick-slip commutation [12]. Other types, such as rolling, are rendered by patterns of impacts [13]. In those models, surfaces are often specified as one-dimensional height profiles, either sampled or algorithmically generated.

Visually, the exploration of a surface has its salient points in the regions of maximal change, e.g. in brightness or colour, which may represent tangible discontinuities such as ridges. In the same way, the evenly-coloured parts of the image represent flat parts of the surface. It is therefore possible to detect visual cues and use their parameters in order to drive other forms of feedback and their intensity (e.g. loudness of a scraping sound, intensity of a vibratory impulse). As a consequence, the representation of any kind of input as a visual surface enables its later multisensory exploration.

The most direct way to accurately replicate the experience of a tangible texture is by modeling the mechanical events which are originated during the exploring action. This modeling must take into account variables such as the local geometry of the surface around the point of contact between probe and surface, or local energy dissipation, which depends on the material.

A dynamic impact model for synthesizing scratching, rubbing and rolling sound-actions has been developed by Conan et al. [14, 15]. Impacts are distributed in time and controlled in amplitude according to stochastic models of these actions. Another synthesis engine is the Sound Design Toolkit [16], which offers a set of physics-based sound algorithms organized according to an ecological taxonomy of everyday sounds. Merrill et al. [17] proposed the gestural exploration of physical surfaces as means to drive the sound synthesis. By brushing, scraping, striking, etc. on physical textures it is possible to control the continuous playback and modification of prerecorded audio samples.

Sketch-a-Scratch aims at reconstructing the surface exploration experience by exploiting the Sound Design Toolkit as a basis for modeling the reaction of different materials to probe contact. The visual texture can be generated by means of image acquisition, real surface scanning, or by the interpretation of audio recordings as surface profile. One immediate use of the audio recordings consists in translating the temporal envelope of a sound into spatially linear information.

3. THE "SKETCH-A-SCRATCH" CONCEPT

Sketch-a-Scratch is an abstract experimental workbench conceived to explore several research-through-design topics: Sonic sketching of surface qualities; creative texture modeling and multimodal exploration; exploration of auditory contents rendered by means of auditory, visual and tactile feedback.

In this paper we focus on the contributions of visual, auditory, and haptic feedback in the probe-mediated exploration of surface textures (see Section 5.2). Aspects such as the interaction style and the qualities of real materials in terms of force dissipation are taken into account, as well as the peculiarities of a single surface. Visual, auditory, and haptic feedback channels are composed to simulate and reconstruct the experience of the contact with a real surface.

The interaction takes place over an interactive surface, namely the touch screen of a tablet, on which a digitized texture is displayed. The user runs the tip of a vibrotactile-augmented stylus on the screen. A physical sound model of a real material (e.g. wood, glass, dry soil) is driven by the exploration of the different features of the surface such as even areas, bumps, creases and ridges. A sound output is consequently generated in real-time, as well as vibrations for the stylus. A local visual deformation helps keeping track of the contact position between probe and screen, and completes the multisensory experience. An overview of the whole system is depicted in Figure 1.

The qualities of a surface can be sketched through several alternative representations of a texture:

- a digital image, whose regions of maximal change of grey-level are interpreted as depth shifts such bumps

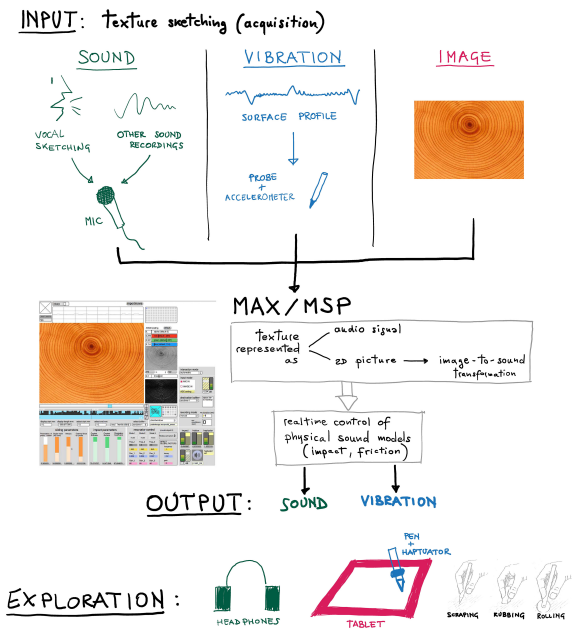


Figure 1. Sketch-a-Scratch concept.

and ridges;

- an audio signal, e.g. a vocal recording, whose features can be preliminarily converted into a visual mesh and finally interpreted as a map;
- a vibration, which can be generated by scanning a real surface with a probe to acquire its linear profile.

The system affords various types of contact (styles of interaction): scraping, rubbing and rolling, obtained by specific combinations of an impact and a friction model. The sensory cues are meant to be synchronous and coherent: A vibrating motor attached close to the stylus tip co-locates the haptic feedback at the point of contact with the surface, whereas intensity and frequency of sonic and vibratory impulse are generated proportionally with the gradient of gray-levels in the area of the image that is being crossed by the stylus.

4. THE "SKETCH-A-SCRATCH" FRAMEWORK

Sketch a Scratch uses an impact model that describes two colliding bodies [12]: a point-mass (exciter) and a resonating object. The contact force f_i is a function of the object compression x and compression velocity $\dot{x}$:

$$f_i(x,\dot{x}) = \begin{cases} -kx^{\alpha} - \lambda x^{\alpha}\dot{x}\,, & x > 0 \\ 0\,, & x \leq 0 \end{cases} \qquad (1)$$

where k accounts for the object stiffness, λ represents the force dissipation, and α describes the local geometry around the contact surface. When $x \leq 0$ the two bodies are not in contact.

In addition, a friction model is used, which describes the relationship between the relative tangential velocity v of two bodies in contact, and the produced friction force f_f. In what follows the exciter is called "rubbing" object while the resonator is called "rubbed" object. The model assumes that friction results from a number of microscopic elastic bristles, accounting for stick-slip phenomena:

$$f_f(z,\dot{z},v,w) = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 v + \sigma_3 w \qquad (2)$$

where z is the average bristle deflection, $\dot{z}$ the average bristle deflection velocity, the coefficient σ_0 is the bristle stiffness, σ_1 is the bristle damping, and the term $\sigma_2 v$ accounts for linear viscous friction. The noise component $\sigma_3 w$ represents surface irregularities. In particular the variable z describes the three regimes accounted for by the model:

elastic: the rubbed object is fixed and does not vibrate, while the rubbing object moves tangentially;

elasto-plastic: the rubbed object vibrates, while the rubbing object moves tangentially;

plastic: the rubbed object does not vibrate and is dragged by the rubbing one.

The two models are used in conjunction to simulate complex vibratory phenomena. A simulated surface profile is used in the impact model to modulate the relative displacement offset between the exciter and the resonating object (i.e., the stylus and the surface). The normal force applied to the stylus is also used to feed the impact model. In addition, when driven by the stylus' tangential motion and the normal reaction force f_i produced by the simulated micro-impacts, the friction model generates stick-slip phenomena.

The impact and friction models produce vibratory signals, which can be output as sound, to render the aural manifestation of texture exploration, as well as used to drive a vibration transducer. The model dynamics also produce forces which can be rendered through a haptic device. Similarly to what done in [18], the stylus is actuated by means of a vibrotactile transducer driven by the low-frequency components of the synthesized audio output. The main components of the system are a Max/MSP patch based on the Sound Design Toolkit [16] running on a Apple laptop, a 13.3" Wacom Cintiq graphic tablet (1920×1080 pixels) and stylus, and a TactileLabs Haptuator Mark II vibrotactile transducer attached to the stylus. For a localized emission of sound and vibration, a dynamic speaker is attached to the back of the tablet and wired to one of the two channels of a Sonic Impact T-Amp amplifier, the other channel being connected to the vibrotactile transducer. Audio signal acquisition is performed via an external microphone, that can be replaced by a portable digital audio recorder for its versatility.

Figure 2 shows the graphical user interface. It allows to load images, record audio tracks, and turn them into surface profiles. Different kinds of virtual materials (e.g. glass-like, metallic, wooden) and interactions (e.g. bouncy, sticky) can be synthesized and saved as presets. Up to six different roughness profiles can be recorded as audio signals and recalled, to drive the synthesis engine. The signals' buffer length is 1000 ms, ideally corresponding

to a 1000-mm-long surface. The 'impact parameters" describe the quality of the single collision (stiffness, sharpness, and energy dissipation affecting the occurrence of bouncing phenomena). The "sliding parameter" layer is used to interpret the stored surface profile and drive the impact model accordingly. The vertical penetration of the probe sets the threshold level of the roughness profile above which the signal is detected, while the probe width parameter sets the size of the sliding window on the roughness profile (in mm, large = rubber, small = sharp object). The probe is advanced every Δt ms by a distance $\Delta x = v\Delta t$, where v is the sliding velocity in m/s. Additional parameters (not displayed in Fig. 2) are Δt in ms and the diameter of a single contact area in cm.

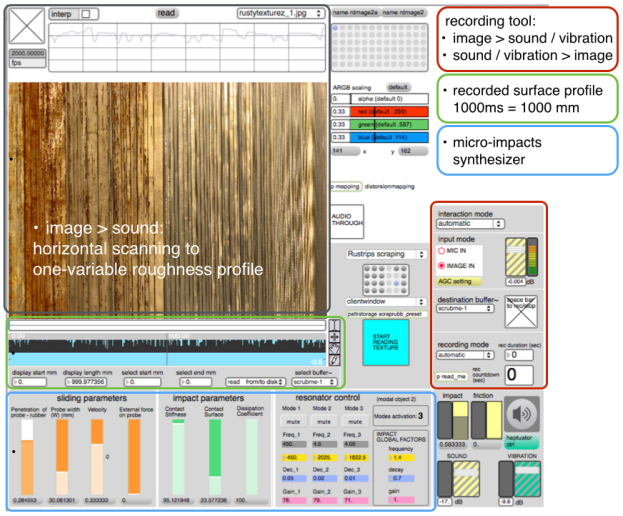


Figure 2. Sketch-a-Scratch GUI.

Thus, the profiles can be explored with virtual probes of different characteristics, to simulate scraping, and rubbing. In our realization, exploration can be either automatic by acting on the GUI (passive), or manually driven through the stylus (active).

In particular, the tilt of the stylus is exploited in active exploration to virtually change the configuration of the probe (i.e., the width), thus shifting the interaction style from scratching (stylus perpendicular to the screen) to rubbing (maximum tilt of the stylus). This feature represents a convenient way to foster the expressiveness of the tool during performative acts. Furthermore, the vertical force relative to the stylus' tip on the screen is used as a control of the vertical penetration of the probe on the virtual surface profile.

Audio or vibrotactile signals (of one variable) can be used to produce an image in different ways. One trivial yet effective transformation used in our tool is the stacking of luminance-translated audio signals to produce rows of pixels. As an example, the four rows, shown in Figure 3, represent the visual textures resulting by the transformation of four different sounds, originated by the vocal imitation of different impact noises (knocks, rolling, sawing, splatters, from top to bottom), each of a duration of 5 seconds.

This sound-to-image transformation affords different kinds of subsequent image-based exploration of the sound material (temporal expansion, inversion, interlacing, etc.). In addition, a local image deformation is applied at the point of interaction to mimic superficial vertical and lateral forces exerted by the stylus.

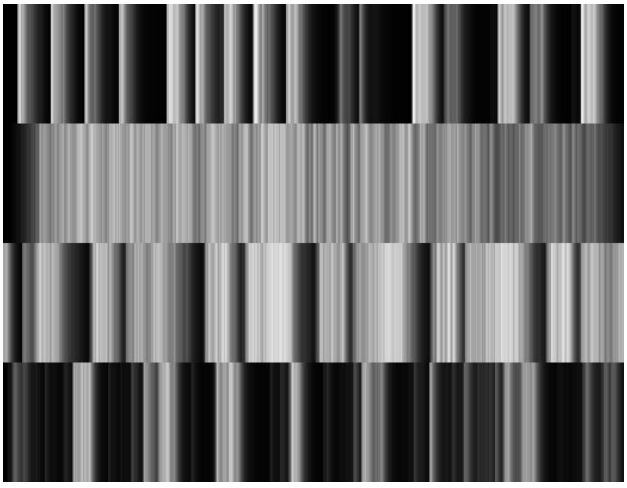


Figure 3. Example of sound-to-image transformations, from top to down: first row, knocks; second row, rolling; third row, sawing; fourth row, splatters.

5. "SKETCH-A-SCRATCH" IN ACTION

The basic configuration served as workbench to investigate the potential of Sketch-a-Scratch in different contexts of use, and for a variety of purposes: demonstrations, experimental research, live performances and installations.

In [5] we exploited the experimental workbench to find quantitative behavioral evidences of the effectiveness of image, sound and vibration as sensory substitutes of lateral forces in texture exploration tasks.

In this paper we focus on the exploitation of Sketch-a-Scratch as a performative tool and as a public installation. The performance setting aimed at sharing with an audience the intimate qualities of contact actions through listening, while the public installation enabled us to test the effectiveness of the multisensory rendition of the surface exploration.

5.1 Performance

Any tool that affords expressive manipulation will become, sooner or later, a device for artistic performance. Many examples are found in the history of musical instruments, from hunting bows converted to string exciters, to turntables converted to expressive scratching instruments [19].

The potential of Sketch-a-Scratch as a device for artistic performance was tested in a public performance in the occasion of the 2014 World Voice Day [1]. In that public event, two exemplars of Sketch-a-Scratch [2] were played by a quartet, as depicted in Figure 4. One vocalist provided vocal textures that were cyclically explored while

[1] http://en.wikipedia.org/wiki/World_Voice_Day.

[2] A video footage is also available at https://vimeo.com/93417532

the impact and friction parameters were dynamically manipulated by another performer. One drawer acted with the stylus on the tablet to explore four different kinds of material textures, each corresponding to one movement of the piece, under the direction of a fourth laptop performer.

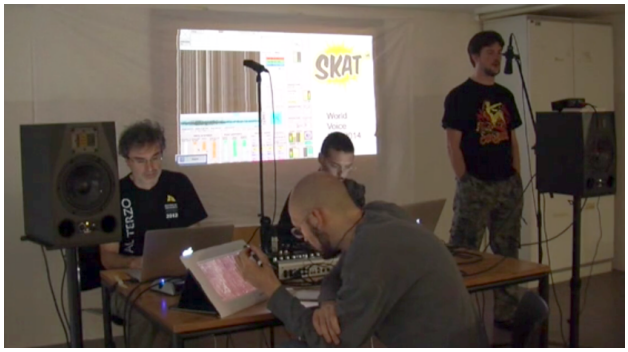


Figure 4. Performance rehearsal for The 2014 World Voice Day. One drawer (front side of the table) is exploring a material texture, while a vocalist (right, standing) is providing vocal textures. Two performers (back side of the table) are manipulating the impact and friction parameters in real time.

Similarly to what happens with musical instruments that are designed around tangible user interfaces [20], the engagement of a performer that is acting on the interactive surface and receiving localized feedback is transferred to the audience by means of body movements, visual projection of the interface, and aural result.

In Sketch-a-Scratch, the auditory feedback is consistent with the performer's actions, and communicates expressive sonic gestures about touch, an experience which is normally personal and non-sharable.

The performing quartet can be seen as a double duo, each duo being formed by a (voice or pen) source performer and a manipulator. As opposed to the usual practices of live electronics, however, the manipulator does not modify the audio material directly. Instead, the manipulator can either select and adjust the textures for pen-based exploration, or use the vocal material as textures to be explored by virtual probing.

5.2 Installation

Sketch-a-Scratch was also showcased as a self-contained interactive installation [3], aimed at demonstrating the contributions of visual, auditory and haptic feedback in the experience of tool-mediated exploration of surface textures. The occasion for this showcase was an academic celebration day including demonstrations, lectures, awards, and gourmet buffet. Like other demonstrations, our installation served as inspiration for the chefs invited to show their food designs.

Given the number of expected participants, and thus the natural presence of a loud background noise, and since the time of stay per visitor was expected to be quite short, both auditory and haptic feedback were exaggerated, in order to provide the Sketch-a-Scratch experience at a glance. Visitors were free to use the stylus to virtually scratch and scrape on four different surface textures displayed on the screen of the vibro-acoustically-augmented tablet. The audience was prompted to explore and savor bumps, ridges and creases, enriched by a vibro-acoustic feedback coherent with the material characteristics of the 2D image displayed on the screen (e.g. plastic, wood, glass). In addition, visitors could also record short audio excerpts, their voice for instance, and interact with the resulting virtual profile. The latter feature was aimed at stressing the richness of one's own vocal capabilities, by providing an immediate engagement in the design of surface textures.

The exhibition let us record valuable observations for further development of the tool. Video recordings, direct observations, talking-aloud impressions and post hoc comments by the visitors, especially regarding their own expectations, were collected in order to revise the system, improve the effectiveness of the interaction, and devise new creative and functional scenarios.

5.2.1 Design

Figure 5 shows the box that we designed to host our system. Sketch-a-Scratch shows up in the empty room as a monolith representing four different textures.

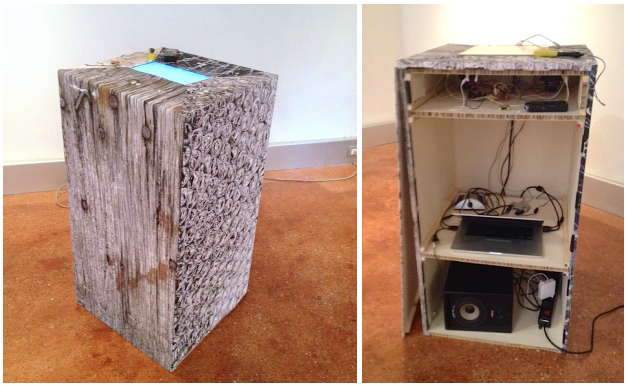


Figure 5. Sketch-a-Scratch installation. On the right, details of the hardware embedded in the box.

As shown in Figure 6, each side of the parallelepiped is covered with a print of a macro-image of a texture surface, namely bubble wrap, broken glass, wooden board, and cracked ground. The four textures were chosen in order to elicit diverse interactive experiences, and possibly prompt different responses and interaction styles or, in other words, gestures. In addition, a well-refined tweaking of the impact and friction parameters was aimed at strengthening the expectations and interaction with the materials displayed. The shell was designed in order to hide all the hardware, and to afford an interaction as natural and ecological as possible.

Users only had to handle the stylus, and start sketching their scratches on one of the four textures at a time or on the voice generated surface profile. The stylus was modified and the tip camouflaged, in order to reduce the effect of the typical affordances of the pen. In addition, users were prompted to hold the stylus between their index and middle

[3] https://vimeo.com/111889017



Figure 6. Macros of the four textures available for exploration: bubble wrap (top-left), broken glass (top-right), wooden board (down-left), cracked ground (down-right).

finger, to avoid the metaphor of writing and facilitate the full experience of touch.

Visitors could browse the available textures on the display by positioning a token on one of the four switches located on the table top, each associated to one side of the shell. The switches were implemented as simple open circuits painted on paper with conductive ink [4]. In Figure 7 it is possible to observe the two rounded electric terminals, placed at the centre of the wooden surface, and the token positioned on the bubble wrap texture.

Finally, users could record their voice by approaching a clearly-visible digital audio recorder, and engage in direct explorations of their sketches.

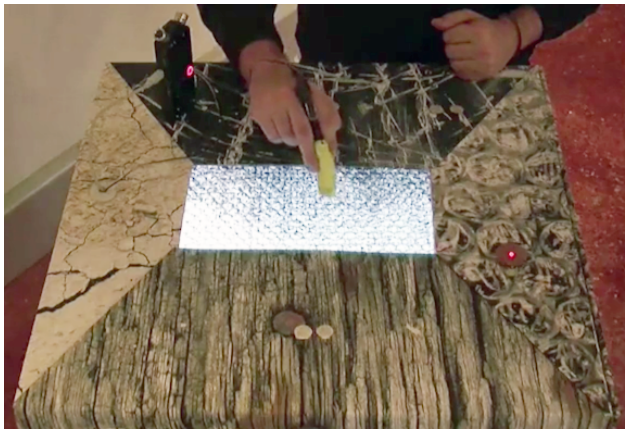


Figure 7. Top of the installation with tablet, stylus and audio recorder. The rounded token with the red led allows to browse and switch between the four textures.

In order to achieve the co-location of visual, auditory and haptic feedback, two small loudspeakers were placed on a shelf just below the tabletop. However, given the presence of a loud background noise, we reinforced the auditory feedback by adding an active speaker, which was placed on the bottom of the box. As a result, the friction sounded darker than what one would naturally expect from real-world situation. The haptic feedback was also reinforced accordingly.

[4] The conductive ink and the magnetic led component are part of the Circuit Scribe system: `http://www.123dapp.com/circuitscribe`.

5.2.2 *Observations*

We filmed the interaction with the installation by the most engaged visitors (12, 7 male and 5 female, average age 30), i. e. those who lingered enough time to acquire a basic understanding of the system and of its features. Environmental noise made comments almost inaudible, nonetheless several interesting comments were extracted. For instance, a professor of modern art history advocated the application of the Sketch-a-Scratch framework to the enhancement of navigation experience in art galleries for the visually impaired. Regarding the movements the visitors employed, different styles of interaction were displayed (see Figure 8), from the regular, neat stroke of a painter to the irregular touch intensity shown by non-trained individuals.

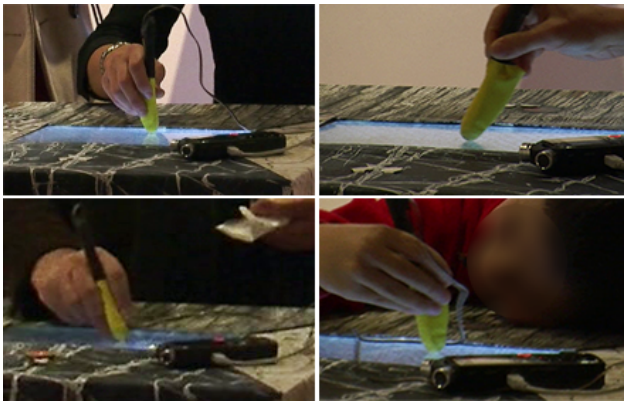


Figure 8. Different styles of interaction employed by visitors. From top left, clockwise: vertical popping, painter-like slanted stroking, quick scribbling, slow crossing of the texture's features.

In general, the installation was positively received. Direct observations of the visitors performing on Sketch-a-Scratch revealed a variety of personal and creative explorations. For instance, many users challenged the expressiveness of the local deformation of the image at the tip of the stylus and started "popping" the virtual surface. This behavior was especially evident in the case of the bubble wrap texture where users try to mimic the usual behavior. Many visitors commented the "popping" on virtual bubble wrap as an accurate and fun experience. Other users focused on the responsivity and fidelity of the feedback, e.g. by crossing slowly the cracks on the glass texture. Some minor latencies were reported. However, this can be attributed not only to the system, but also to the larger size of the "eraser" tip (compared to the pen tip), which reduces the friction on the display, though at the cost of a less accurate detection of the impacts. In addition, among the three sensory feedbacks, the haptic feedback took by surprise most of the users, at the same time being assessed as the most effective.

The auditory feedback was well received too, although

most of the comments were spurred by the vibrotactile feedback. Residual inaccuracies in the auditory response were not deemed as important, thus suggesting that users were more focused on visuals and haptics than on sounds. However the presence of a coherent sound played a role in augmenting the immersiveness of the experience. Specifically, the images were more appreciated as a navigation guidance than as a feedback source. The local distortion at the contact point of the tip on the surface went barely noticed, although they were crucial in letting the "popping" affordance emerge.

The audio sketching mode was received with milder interest due to its lower degree of immediacy, especially when the visitors were prompted to expose their body and voice in public. Most users were more prone to attend demonstrations of vocal sketching than to try it themselves in presence of others. Nevertheless, the audience was intrigued by the potential of the sketching tool and of the possible development and applications. Moreover, comments stressed a generally clear understanding of the causal link between the vocal gesture and its visual rendition: after a brief explanation, the visitors could recognize the visual impression of simple vocalizations, such as sustained sounds, rhythmic patterns, trills, etc.

6. CONCLUSIONS AND FUTURE WORK

We introduced Sketch-a-Scratch, a multisensory tool for probe-mediated sketching and exploration of augmented surfaces. The tool is currently being exploited as a framework to investigate the perceptual and cognitive aspects involved in the probe-mediated experience of (virtual) surfaces, and to expose the affordances and inherent expressiveness of this kind of interaction for design purposes and performative uses.

The rendering of this experience in virtual environments, such as ordinary interactive flat visual displays, requires effective strategies of augmentation, in terms of both actuating technology and design choices in feedback manipulation. In Sketch-a-Scratch, a strategy based on a physically-informed approach to sound synthesis and pseudo-haptics resulted effective in conveying the salient aspects of contact phenomena such as scraping and rubbing.

At the same time, the experiences collected with the current configuration of our tool, and in its diverse contexts of use, also highlighted the limits of virtualization. Coherent multisensory stimuli certainly increase naturalness in the interactions with virtual surfaces, resulting in a higher expressiveness during creative efforts. However, the actual lateral forces that are experienced when scraping a real surface with a tool remain hard to reproduce with sensory illusions; in addition, the visual feedback plays a predominant role over auditory and haptic feedback in trajectory-based tasks [5]. On this standpoint, we will investigate the effectiveness of our vibroacoustic augmentation approach of flat displays in conjunction with 3D textures. In particular, by superimposing a thin 3D texture on the display, the two-dimensional information (i.e., speed and location of the stylus) extracted by the Wacom can be integrated with the stylus information (i.e., tilt and force) deriving from the actual interaction with the real asperities of the overlay. We are currently making some explorations with 3D textures of few millimeters of thickness. For example, Figure 9 shows a three-dimensional realization of the four vocal imitations depicted in Figure 3. A 3D print of this tile was already used in public demonstrations to sensitize the participants to "real" probe-mediated texture exploration and to give a concrete example of what "scraping a vocal sound" means in practice [21].

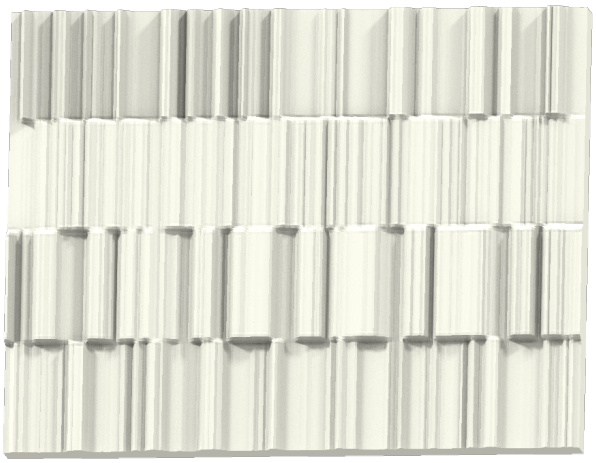


Figure 9. Rendering of the 3D printed texture representing the profiles derived from the sound-to-image transformations depicted in Figure 3.

Sound and vibration can be exploited to enhance the experience of creative acts such as painting and drawing, when these activities are performed on interactive surfaces. In addition, the stylus could be used not only as a probe, but also as an active tool for texture manipulation. A designer might wish to flatten or curl a region of the virtual surface, or to displace it. Finally, the integration of vocalizations in the sketching process might lead to a scenario where voice and hands are in a continuous conversation, thus collaborating seamlessly in the molding of the creative result. In this respect, Sketch-a-Scratch is a modulator of problem space, and serves as an open workbench for our design research in virtual texture modelling.

Acknowledgments

The work described in this paper is part of the project SkAT-VG, which received 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: 618067. The authors wish to thank Stefano Baldan for his contributions in performing with Sketch-a-Scratch and in realizing the installation, and Francesco Lenci for helping in the realization of the shell for the installation.

7. REFERENCES

[1] R. L. Klatzky and S. J. Lederman, "Multisensory texture perception," in *Multisensory object perception in the primate brain*, M. J. Naumer and J. Kaiser, Eds. Springer Verlag, 2010, pp. 211–230.

[2] G. Robles-De-La-Torre, "The importance of the sense of touch in virtual and real environments," *IEEE Multimedia*, vol. 13, no. 3, pp. 24–30, 2006.

[3] A. Lécuyer, "Simulating haptic feedback using vision: A survey of research and applications of pseudo-haptic feedback," *Presence: Teleoperators and Virtual Environments*, vol. 18, no. 1, pp. 39–53, 2009.

[4] K. V. Mensvoort, P. Vos, D. J. Hermes, and R. V. Liere, "Perception of mechanically and optically simulated bumps and holes," *ACM Trans. Appl. Percept.*, vol. 7, no. 2, pp. 10:1–10:24, Feb. 2010. [Online]. Available: http://doi.acm.org/10.1145/1670671.1670674

[5] D. Rocchesso, S. Delle Monache, and S. Papetti, "Multisensory texture exploration at the tip of the pen," *Special Issue on Data sonification and sound design in interactive systems*, 2015.

[6] M. Fukumoto and T. Sugimura, "Active click: Tactile feedback for touch panels," in *Proceedings of CHI 2001*. ACM, 2001, pp. 121–122.

[7] J. Lee, P. Dietz, D. Leigh, W. Yerazunis, and S. Hudson, "Haptic pen: A tactile feedback stylus for touch screens," in *UIST'2004*. ACM, 2004, pp. 291–294.

[8] K. Kaczmarek, J. Webster, P. Pach-y Rita, and W. Tompkins, "Electrotactile and vibrotactile displays for sensory substitution systems," *IEEE Transactions on Biomedical Engineering*, vol. 38(1), pp. 1–16, 1991.

[9] A. Yamamoto, S. Nagasawa, H. Yamamoto, and T. Higuchi, "Electrostatic tactile display with thin film slider and its application to tactile telepresentation systems," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12(2), pp. 168–177, 2006.

[10] O. Bau, I. Poupyrev, A. Israr, and C. Harrison, "Teslatouch: Electrovibration for touch surfaces," in *UIST'2010*. ACM, 2010.

[11] F. Giraud, M. Amberg, B. Lemaire-Semail, and G. Casiez, "Design of a transparent tactile stimulator," in *IEEE Haptics Symposium 2012*. IEEE, 2012, pp. 121–122.

[12] F. Avanzini, S. Serafin, and D. Rocchesso, "Interactive simulation of rigid body interaction with friction-induced sound generation," *IEEE Trans. Speech and Audio Proc.*, vol. 13, no. 5, pp. 1073–1081, Sept 2005.

[13] M. Rath and D. Rocchesso, "Continuous sonic feedback from a rolling ball," *IEEE Multimedia*, vol. 12, no. 2, pp. 60–69, 2005.

[14] S. Conan, E. Thoret, M. Aramaki, O. Derrien, C. Gondre, R. Kronland-Martinet, and S. Ystad, "Navigating in a space of synthesized interaction-sounds: rubbing, scratching and rolling sounds," in *Proceedings of the 16th Int. Conference on Digital Audio Effects*, Maynooth, Ireland, Sep. 2013. [Online]. Available: http://dafx13.nuim.ie/papers/32.dafx2013_submission_43.pdf

[15] S. Conan, E. Thoret, M. Aramaki, O. Derrien, C. Gondre, S. Ystad, and R. Kronland-Martinet, "An intuitive synthesizer of continuous-interaction sounds: Rubbing, scratching, and rolling," *Computer Music Journal*, vol. 38, no. 4, pp. 24–37, 2014.

[16] 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. [Online]. Available: http://doi.acm.org/10.1145/1859799.1859800

[17] D. Merrill, H. Raffle, and R. Aimi, "The sound of touch: physical manipulation of digital sound," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 739–742. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357171

[18] C. G. McDonald and K. J. Kuchenbecker, "Dynamic simulation of tool-mediated texture interaction," in *World Haptics Conference (WHC), 2013*. IEEE, Apr. 2013, pp. 307–312. [Online]. Available: http://dx.doi.org/10.1109/whc.2013.6548426

[19] J. Sterne, "Media or instruments? yes," *Offscreen*, vol. 11, no. 8-9, 2007. [Online]. Available: http://offscreen.com/view/sterne_instruments

[20] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner, "The reactable: Exploring the synergy between live music performance and tabletop tangible interfaces," in *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, ser. TEI '07. New York, NY, USA: ACM, 2007, pp. 139–146. [Online]. Available: http://doi.acm.org/10.1145/1226969.1226998

[21] S. Delle Monache, D. Rocchesso, and S. Papetti, "Sketch a scratch," in *Eight International Conference on Tangible, Embedded and Embodied Interaction, Work in progress*, ser. TEI '14, 2014.

INTER-CHANNEL SYNCHRONISATION FOR TRANSMITTING LIVE AUDIO STREAMS OVER DIGITAL RADIO LINKS

Dr. Stephen Brown
Computer Science, Maynooth University
stephen.brown@nuim.ie

Jorge Oliver
Computer Science, Maynooth University
grusite@gmail.com

ABSTRACT

There are two key challenges to the use of digital, wireless communication links for the short-range transmission of multiple, live music streams from independent sources: delay and synchronisation. Delay is a result of the necessary buffering in digital music streams, and digital signal processing. Lack of synchronisation between time-stamped streams is a result of independent analogue-to-digital conversion clocks. Both of these effects are barriers to the wireless, digital recording studio.

In this paper we explore the issue of synchronization, presenting a model, some network performance figures, and the results of experiments to explore the perceived effects of losing synchronization between channels. We also explore how this can be resolved in software when the data is streamed over a Wi-Fi link for real-time audio monitoring using consumer-grade equipment. We show how both fixed and varying offsets between channels can be resolved in software, to below the level of perception, using an offset-merge algorithm. As future work, we identify some of the key solutions for automated calibration.

The contribution of this paper is the presentation of perception experiments for mixing unsynchronized music channels, the development of a model representing how these streams can be synchronized after-the-fact, and the presentation of current work in progress in terms of realizing the model.

INTRODUCTION

One of the key challenges for the digital music studio is communicating the digitized sound data from the sources (instruments and microphones) to the mixing-desk for recording. The benefits are the removal of physical wiring, and increased flexibility; but the challenges due to increased latency and inter-channel synchronization are not insignificant.

Published results show that a delay between the visual and sound data of over 1.4-42ms is apparent [1] (for example, to a sound engineer on the recording desk). Even though Interaural Time Difference (ITD) is a well known phenomenon, there is less published work on the impact on the listener on inter-channel synchronization errors, and how to resolve them. In this paper we review related work, present our experimental results on the audibility of inter-channel synchronization errors, and present a solution showing that these errors can be reduced below a perceptible margin.

Copyright: © 2015 Stephen Brown. 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.

RELATED WORK

How a time delay between receiving a sound in the ears (ITD) is processed is explored in the original Jeffress Model [2] and more recent refinements (e.g. [3,4,5]). The ability to detect ITD and its effect on the localization of the sound source has been explored in many papers (e.g. [6], [7], and [8]) and a good overview of the research is presented in [9].

Sound spatialization for listeners is caused by both interaural intensity (IIT) and time differences (ITD) [8]. In trying to understand the impact of de-synchronisation between digital audio streams, we are only concerned with ITD. The apparent offset is frequency dependent (e.g. as discussed in [11]), but approximations from [5] are shown for frequencies below 500Hz (Equation 1) and above 2kHz (Equation 2), for an incident angle θ as shown in Figure 1. ITD is the inter-aural time difference, a is the head radius, c is the speed of the sound, and θ (in radians) is the incident angle shown in Figure 1. The angle θ is the perceived difference in the angle from which the sound is sourced (with respect to 0° indicating no synchronization error).

$$ITD_{500Hz} = (a/c)2 \sin \theta \quad (1)$$

$$ITD_{2kHz} = (a/c)(\theta + \sin \theta) \quad (2)$$

The consequences of this are that if two channels are not correctly synchronized, then an artificial apparent movement (θ) of the sound's source will result[1].

There are a number of different figures published for the Just Noticeable Difference (JND): for example in [12] a range of 10-20 μs is presented, and 15μs in [13].

[1] For low frequencies, $\theta = \sin^{-1}[ITD/(2a/c)]$
$\approx ITD/(2a/c)$ for small angles

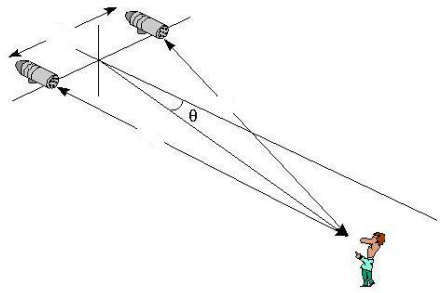


Figure 1. Sound Localisation Geometry.

The area of Networking Musical Performance (NMP) where physically remote musicians can play together is receiving significant research interest [14,15,16]. High latency introduces significant problems, specifically in terms of maintaining tempo [17]. The issues here are to do with synchronizing one's own performance with the other participants – and typical acceptable figures for latency an order of magnitude larger than the ITD figures (30-90 ms [17], 50-65ms [18]) – with larger latencies tolerable for slower tempi.

A significant amount of work has been done on using correlation to identify inter-signal time shifts (for example in [19]). In this paper we focus on the re-synchronisation problem.

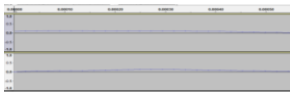
PERCEPTION OF ITD

As discussed in the previous section, there has been considerable research into exploring the limits of perception of ITD effects. However, much of this has used artificial or generated signals, for example [6,7,8,20,21], designed to explore the limits of human perception. Relating these results to the problem of un-synchronised music channels is not straightforward. Some examples using real-world signals are an extensive investigation into the comb filtering effect of inter-signal delays [19], the effect of latency on monitoring [1], and the effects of latency on remote music performance [14].

In this section we present the results of an experiment specifically designed to explore the limits of perception for two unsynchronized music channels. The purpose is to establish a baseline for the time-accuracy required in re-synchronising channels. The results of the experiment are presented for a number of listeners (using headphones). There are two key apparent effects: a shift in the apparent localization of the source, and an apparent change in the volume of the earlier stream.

The experimental setup consisted of preparing four different samples of three different stereo music recordings. Each of the four samples had a different delay introduced for the right channel (varying from 0 to 6 samples at 44,100 samples/second: 0 to 136 μs), using the Audacity *Time Shift Tool*, as shown in Figure 2.

In a limited experiment, three different volunteers (the columns labeled 1,2,3 in Table 1) were used for the experiment, with varying degrees of experience in music. The volunteers were asked whether they could perceive a shift in the apparent source of the sound (or any other difference in the sound experience, such as the relative volume, or other qualitative effects). This was reported on a scale of 0 (no perceivable difference with respect to the 0-sample shift recording) to 10 (a very noticeable shift that required no concentration to perceive) – these results are the figures shown in Table 1. The 0 figures shown for the 0-shift recording just represent the fact that this is the baseline against which the other recordings are compared.

(a) 0-sample offset 

(b) 2-sample offset 

(c) 4-sample offset 

(d) 6-sample offset 

Figure 2. Channel Offsets.

The results, shown in Table 1, indicate that a shift of 45.35μs was not perceivable, a shift of 90.1μs was just perceivable, and a shift of 136 μs was clearly perceivable for this sample group.

		Perceived		
Title	**Shift**[2]	**1**	**2**	**3**
Andrea Bocelli and Elisa	0	0	0	0
La Voce Del Silenzio	2	1	5	0
from Vivere	4	8	9	5
	6	10	10	10

		Perceived		
Title	**Shift**	**1**	**2**	**3**
Queen	0	0	0	0
We are the Champions	2	0	4	0
from Absolute Greatest	4	7	5	5
	6	10	10	10

		Perceived		
Title	**Shift**	**1**	**2**	**3**
Vivaldi	0	0	0	0
Larghetto from Concerto	2	2	6	0
Grosso OP. 3/8 in A-Minor	4	8	10	5
from Famous Concertos	6	10	10	10

Table 1. Experimental Perception Results.

[2] Shift is measured in samples at 44,100 samples-per-second: 1 sample=22.676μs

The conclusions drawn from these results are that synchronisation between the channels needs to be maintained within a window of at least 50µs in order to not introduce perceivable effects into a stereo music stream. Note that this is less than 3 samples at 44kHz, or less than 10 samples at 192kHz. Obviously, a larger sample group is required to determine a more robust figure. We will take this figure as our initial target for synchronisation; part of our future work is to do similar experiments for a larger selection of listeners.

SYNCHRONISATION MODEL

The key to synchronizing multiple, digital music streams is to know (a) the exact start time, and (b) the sample rate of each. This can be achieved in two ways: either by using a common clock (which can be used to solve both issues at once), or by independent clocks and timestamps (which allows the issues to be solved independently). Traditional mixing desks use the first approach: in the analog case, merely by having closely matched components and signal paths, and in the digital case by clocking the Analog-Digital Converters (ADC) synchronously. In the distributed sources case, as would be seen in a fully digital recording studio, either approach can be taken.

Achieving a microsecond-level master clock is not straightforward, but once it is achieved, then the synchronization problem is solved. In this work we address the more difficult case, where each ADC has its own clock, and both the start time of the stream, and the offsets between streams need to be maintained to allow accurate synchronization. We first show a general model of such a system, and then address one of the key algorithms – the real-time mixing of unsynchronized streams.

The basic model for the data flow with a single stream is shown in Figure 3.

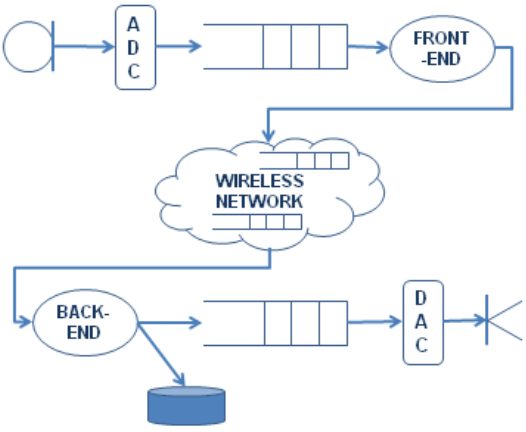


Figure 3. Single Stream Model.

The analog signal (from a microphone or instrument) is sampled and converted by the ADC. The time of the first sample is determined by when the software starts the ADC; the rate of the sampling is determined by both rate selected by the software, and by the accuracy of the ADC's internal clock. The data is then buffered in a queue to be delivered to the front-end software (responsible for relaying the audio stream). This buffer must be large enough to allow for the latency of the front-end software (to prevent buffer over-runs). The front-end then delivers the data to the network.

In this case we are considering a low-latency, local network (and not the Internet in general) – specifically the case where there are no routers in the network, so that congestion and packet prioritization are not issues. The data is then delivered to the back-end software (the mixing desk) which (a) stores the data on disk for recording and (b) delivers the data to a DAC to be converted for the monitoring function. Again, buffering to the DAC must be used to prevent buffer under-runs. A multi-stream model is shown in Figure 4.

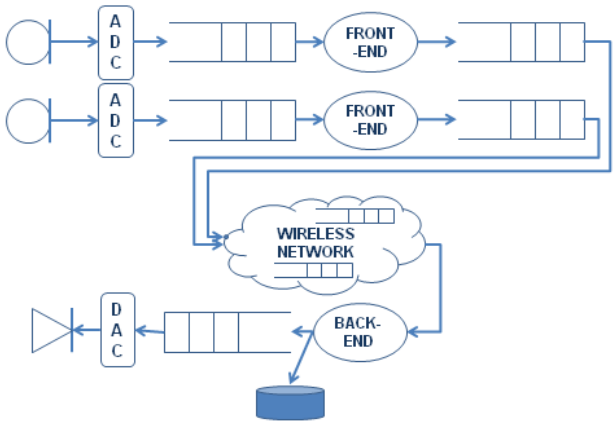


Figure 4. Multi-Stream Model.

Once an ADC has been started, then data will be delivered at a constant rate (dependent on the accuracy of the clock) over the data stream to the DAC. The size of this buffer is dependent on the ADC conversion rate, and the worst-case latency of the front-end software. The impact of the network is to introduce a delay in each stream which is not dependent on the data rate (it has two components: in the loss-free case, it is determined by the characteristics of the intervening network; in the lossy case, it will also be determined by the network retransmission strategy). Note that, as long as the ADC buffer never over-runs, the recording function is reliable; but the real-time monitoring function is also dependent on the network latency. It is assumed that the clocks of the source systems are synchronized (e.g. using PTP within 1µs, as discussed in [22] and [23]). The ADC clocks are, however, assumed to be independent.

To synchronise streams, a timestamp is required at the start of each stream. This allows the relative offset between the streams to be identified. To maintain synchronization (in the face of clock drift between multiple ADCs) a periodic timestamp is required in each stream. Even though the data may be delivered in smaller chunks,

as determined by the ADC buffer size, and the transport protocol behavior, the stream is therefore packetized into large packets (for example, 1 second long, with a timestamp and other status information at the head of each).

In this paper, we do not address the problems in resynchronising the independent streams after the recording function (obviously this will be required for final mixing). For the real-time monitoring function, the two issues of latency and synchronization must be addressed in real-time: latency, so that the sound engineer does not see a perceivable lag between the musicians' actions and the monitored sound; and synchronization, so that volume and position artifacts are not introduced. In this paper we show some initial results, showing that the first problem is surmountable, and address in more detail the second problem of mixing multiple digital audio streams with a time delay (using an offset merge, presented in the next section).

OFFSET MERGE AND RESAMPLING

The mix function requires two (or more) audio streams to be merged with a time offset (in order to counteract the offsets in the start time of each stream). The offset may not be fixed, due to the independent nature of the ADC clocks, and so we also present a resample-and-merge algorithm for this case (note that the periodic timestamps allow this case to be identified).

We introduce a simple Offset-Merge Algorithm (OMA) for achieving this, and present some initial results. To date we have implemented 2-stream OMA – multi-stream OMA is work in progress.

Offset Merge

A simple multi-channel merge algorithm with a per-channel offset allows inter-channel synchronization to be achieved. An example of the operation of this algorithm is shown for the 2-channel case in Figure 5.

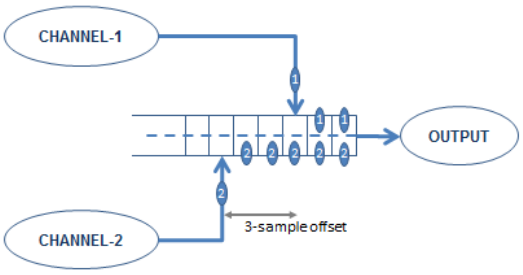


Figure 5. 2-Channel offset merge, with a 3-sample offset.

The two separate input streams are merged into a single two-channel stream as an output (which can be directly queued to the DAC). This requires that the merge buffer be at least twice the size of the output (DAC) buffer. If the offset is larger than this, then the intervening samples from the earlier channel can either be played on their own, or discarded.

Pre-Merge Resampling

Cumulative differences between input channel sampling rates can be detected by the periodic timestamps. It is assumed that the drift is less than one sample per DAC buffer size. Assuming that the sampling clocks have reasonably high accuracy (say 1 ppm) then this error will be reflected in an occasional 1-sample difference. The channel (or tracks in the case of multi-track communication) with fewer samples are re-sampled in real-time up to the larger number of samples (Figure 6 only shows 3 samples – in practice an entire buffer of say 1024 samples would be used) before being output to the monitoring DAC. Note also that, for simplicity, Diagram 5 shows an offset of 0 samples for the offset merge that follows the resampling.

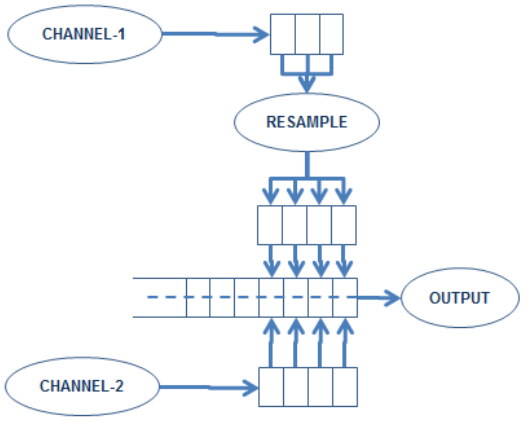


Figure 6. Resampling prior to an offset merge.

The real master clock in this case is actually the DAC output clock – it is future work to adopt this algorithm to use this as the reference clock for re-sampling (i.e. all inputs streams will be candidates for re-sampling to keep in synchronization with the output DAC).

EXPERIMENTAL RESULTS

The results are shown here for experiments evaluating the effectiveness of the Offset-Merge Algorithm for inter-stream synchronisation.

Experimental Setup

A Wi-Fi network was setup between two Linux nodes, with a server representing the mixing desk, and a client representing a digital, wireless microphone, as shown in Table 2.

The parameters were selected to reflect a data source of 44,100 2-byte samples per second as closely as possible. This allows a representative measurement of latency to be measured. The TCP Nagle algorithm was disabled using the TCP_PUSH protocol option to minimize network protocol latency.

Parameter	Value
Ethernet (Server-to-AP)	100 Mbps
Wi-Fi (Client-to-AP)	802.11g 36Mbps
Buffer Size	1024 bytes 512 samples
Inter-buffer delay	11 ms
Throughput	46,545 samples/s 93,090 bytes/s
Duration	400 buffers 4.4 seconds

Table 2. Configuration parameters.

Re-synchronisation

Measurements showed that channels can be successfully merged with an offset to resolve the inter-channel loss of synchronization down to single sample resolution. Fixed de-synchronisation delays were used in generating the desynchronized channels, and these were successfully removed at the merge stage. Some informal listening experiments with headphones also showed successful results.

There is significant contribution of the audio buffering to the total latency: for the 1024-byte buffer we found to be successful in preventing under-runs, this contributed 11.6ms at each end, giving a base latency of 23.2ms prior to any additional network or processing latency.

CONCLUSIONS & FUTURE WORK

Conclusions

The results of the project to date have shown the feasibility of model for a wireless studio with independently clocked digitization (ADCs) and real-time sound monitoring using off-the-shelf parts. These results can be reasonably extrapolated from a non-RT to an RT environment (such as RTLinux).

We have measured the required inter-stream synchronisation, and presented a simple algorithm for achieving this in real-time. In the context of the network and processing latencies of a standard Linux system, we have shown that with time-stamped data streams, this algorithm can execute successfully.

The project has identified a number of key additional areas to be investigated in order to determine the feasibility of a fully-digital, wireless studio (where all communication is digital). These are discussed in the next section.

Future Work

Four key issues have been identified for future work: real-time operation, a real-time wireless network, manual synchronisation controls, and automated synchronization.

Synchronisation

Inter-node synchronization can be achieved down to microsecond accuracy [22,23]. If the sound capture hardware/software supports it, then they can just be synchronized simply using the inter-node synchronised clock. If not, then the latency between the ADC converting a sample and its delivery to the software layers needs to be determined. On a non-RT O/S this can be estimated from timestamping the initial 'start' request to the CPU; by using provided latency measures (e.g. PulseAudio has a `get_latency()` function); or by external means (e.g. a calibration signal). We plan to evaluate these three measures, and, if the third measure is required, investigate auto-calibration options.

Real-Time Operating System

The key to low latency is to minimize buffer size (which preventing underflows). In a non-RT system (such as Linux) there is significant and variable system overhead, which requires relatively large buffer sizes (e.g. up to 1024 bytes=512 samples=11.6ms). With this buffer size at the recording ADC and the playback DAC, the latency is already in the region of 30ms – well above the 10ms target figure. This could be significantly mitigated by use of a real-time platform. We plan to investigate the use of real-time processes under Linux to see how well they perform; we also plan to experiment with an embedded system (powered by an Atmel CPU and a suitable RTOS), providing a guaranteed latency to hardware events in the order of 10's of microseconds. We plan to investigate this using a real-time operating system to investigate the minimum buffer size that can operate with no underflows.

Real-Time Wireless Networking

TCP and Wi-Fi are not intended for real-time operation – and again, buffering has to be used to prevent data underruns. This, in general, requires another 1024-byte buffer, this time at application level at the playback end, adding another 11.6ms to the latency. We intend to experiment with two alternatives here: one is the use of the RTP transport protocol (though for direct node-to-node communication, with a very low level of packet loss/retransmission and no congestion, we don't expect this to provide significant improvements).

We intend to experiment with UDP and a customized re-transmission policy that ignores congestion, and focuses on short-latency retransmissions.

Additional experiments are planned using a dedicated, point-to-point wireless protocol in association with an RTOS, and a simplified transport protocol that handles packet loss aggressively but not congestion (as in the configurations envisaged, all communication is node-to-node with no intervening routers and possibly a single high-performance Ethernet switch).

Resampling

The resampling operation has not yet been verified – but as we see this as purely a monitoring (rather than a recording) process, we don't expect it to cause any significant problems (and certainly no loss of quality of the rec-

orded signal). We plan to experiment with up-sampling and down-sampling to determine whether this causes a perceivable effect to the monitored signal.

Automated Synchronisation

A number of papers have reported ways of automatically determining the time-shift between two channels by correlating features in the sound streams, for example [19]. We plan to experiment with incorporating these into the offset merge operation to determine their effectiveness at the level of resolution required (i.e. approximately 2 samples).

REFERENCES

[1] M. Lester and J. Boley, "The Effects of Latency on Live Sound Monitoring", in Audio Engineering Society Convention 123, 2007.

[2] L.A. Jeffress, "A place theory of sound localization", in J Comp Physiol Psychol 41, 1948, pp. 35-39.

[3] D.C. Fitzpatrick, S. Kuwada, and R. Batra, "Neural Sensitivity to Interaural Time Differences: Beyond the Jeffress Model", Journal of Neuroscience, 20(4), 2000, pp. 1605-1615.

[4] J. Blauert and J. Braasch, "Acoustic Communications: The Precedence Effect", in Proc. of the Forum Acusticum, Budapest, OPAKFI, 2005.

[5] R.M. Tern *et al*, "Binaural Sound Localization", Chapter in Computational Auditory Scene Analysis, G. Brown and DeL. Wang, Eds., IEEE Press, 2006.

[6] W. Nager *et al*, "Tracking of multiple sound sources defined by interaural time differences: brain potential evidence in humans", Neuroscience Letters 344 (2003), Elsevier, 2003, pp. 191-184.

[7] D.P. Phillips, M.E. Carmichael, and S.E. Hall, "Interaction in the perceptual processing of interaural time and level differences", Hearing Research 211, Elsevier, 2006, pp. 96-102.

[8] N.H. Salminen *et al*, "Human cortical sensitivity to interaural time difference in high-frequency sounds", Hearing Research 323, 2015, pp. 99-106.

[9] K. Vonderschen and H. Wagner, "Detecting interaural time differences and remodeling their representation", Trends in Neurosciences, 37(5), 2014, pp. 289-300.

[10] S.G. Goodridge and M.G. Kay, "Multimedia Sensor Fusion for Intelligent Camera Control", in Proc. IEE/SICE/RSJ Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems, IEEE 1996, pp. 655-662.

[11] S. Carlile, "The Physical and Psychophysical Basis of Sound Localization", Chapter in Virtual Auditory Space: Generation and Applications, Springer, 1996, pp. 27-78.

[12] R.C.G. Smith and S.R. Price, "Modelling of Human Low Frequency Sound Localization Acuity Demonstrates Dominance of Spatial Variation of Interaural Time Difference and Suggests Uniform Just-Noticeable Differences in Interaural Time Difference", PLOS ONE, Feb. 18, 2014.

[13] H.S. Colburn, B. Shinn-Cunningham, G. Kidd, Jr, and N. Durlach, "The perceptual consequences of binaural hearing", International Journal of Audiology, 45(supplement 1), 2006, pp. S34-S44.

[14] X. Gu *et al*, "Network-centric Music Performance: Practice and Experiments", IEEE Communications Magazine, June, 2005, pp. 86-93.

[15] R.E. Saputra and A.S. Prihatmanto, "Design and Implementation of BeatME as A Networked Music Performance (NMP) System", in Proc. International Conference on System Engineering and Technology (ICSET), Bandung, IEEE, 2012, pp. 1-6.

[16] C. Alexandraki and D. Akoumianakis, "Exploring New Perspectives in Network Music Performance: The DIAMOUSES Framework", Computer Music Journal, 34(2), 2010, pp. 66-83.

[17] P.F. Driessen, T.E. Darcie, and B. Pillay, "The Effects of Network Delay on Tempo in Musical Performance", Computer Music Journal, 35(1), 2011, pp. 76-89.

[18] N. Bouillot, "nJam User Experiments: Enabling Remote Musical Interaction from Milliseconds to Seconds", in Proc. New Interfaces for Musical Expression (NIME'07), NY, 2007, 6 pages.

[19] A. Clifford and J.D. Reiss, "Using Delay Estimation to Reduce Comb Filtering of Arbitrary Musical Sources",J.Audio Eng.Soc 61(11),2013,pp. 917-927.

[20] S. Tolnai, R.Y. Litovsky, A.J. King, "The Precedence Effect and its Buildup and Breakdown in Ferrets and Humans", J. Acoust. Soc. Am., 135(3), March 2014, pp. 1406-1418.

[21] C. Tsakostas and J. Blauert, "Some New Experiments on the Precedence Effect", Fortschritte der Akustik, 27, 2001, pp. 486-487

[22] A. Smimite, K. Chen, and A. Beghdadi, "Next-Generation Audio Networking Engineering for Professional Applications", in Proc. 20th Telcommunications forum (TELFOR 2012), Belgrade, IEEE, 2012, pp. 1252-1255.

[23] M. Rautiainen *et al*, "Swarm Synchronization for Multi-Recipient Multimedia Streaming", in Proc. Multimedia and Expo (ICME2009), NY, IEEE 2009, pp. 786-789.

MUSICMEAN: FUSION-BASED MUSIC GENERATION

Tatsunori Hirai
Waseda University
tatsunori_hirai@asagi.waseda.jp

Shoto Sasaki
Waseda University
/ CREST, JST

Shigeo Morishima
Waseda Research Institute
for Science and Engineering
/ CREST, JST
shigeo@waseda.jp

ABSTRACT

In this paper, we propose *MusicMean*, a system that fuses existing songs to create an "in-between song" such as an "average song," by calculating the average acoustic pitch of musical notes and the occurrence frequency of drum elements from multiple MIDI songs. We generate an in-between song for generative music by defining rules based on simple music theory. The system realizes the interactive generation of in-between songs. This represents new interaction between human and digital content. Using MusicMean, users can create personalized songs by fusing their favorite songs.

1. INTRODUCTION

Because composing songs from scratch is difficult for inexperienced people, musical composition used to be the act of expression only approved for people with musical sense. However, the development of music creation software has made it easy for a wide range of people to create music. Despite easy-to-use music creation software, some people, particularly those who are not accustomed to expressing their feelings through music, have difficulty creating original music. Therefore, we propose *MusicMean*— a music creation system that allows users to create songs by fusing existing songs.

Music evokes subjective impressions. For example, people feel that "this song is good" or "this song lacks meaning." Such impressions are spontaneous and occur even though the listener does not have a good grounding in music theory or does not possess musical sense. These impressions can be a source of creativity. MusicMean enables users to create a new song interactively by fusing songs depending on the user's impressions of existing songs. Thus, people with little musical knowledge or experience can construct personalized music content while listening to the fused music. We call this fused music an "in-between song." While this may not actually be original song writing, it appeals to an essential motivation for music composition. MusicMean represents a small step toward an era in which everyone can create personalized musical content. Our goal is to realize such a content creation environment.

Copyright: ©2015 Tatsunori Hirai 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.

We also propose the concept of an "average song," as a part of in-between song, by calculating the average[1] of musical elements, e.g., notes from several songs.

In most circumstances, listeners cannot change an existing song. Listeners had to listen to the song as it is unless they arrange it. Music has always been what listeners listen to, but what listeners do not create. Our objective is to realize a new interaction between listeners and musical content. MusicMean allows listeners to alter a song to suit their preference via exploration of space in-between existing songs.

In the context of researches about musical experiences for novices, many new instruments, interfaces or applications are proposed. Blaine and Fels explored the context of the research on collaborative musical experiences for novices including music composition experiment and introduced them on [1]. MusicMean also provides user a new musical experience using existing music rather than creating whole new sound from scratch. To support novice users compose music, there are researches area called Computer-Assisted Composition (CAC) [2–5]. MusicMean can also be regarded as an application in the context of CAC with a possibility to expand creativity of novice users. At the same time, MusicMean is music exploration tool so that the user does not have to intend to compose but enjoy the generated song itself.

The remainder of this paper is organized as follows. Related work is discussed in Section 2. We present an outline of the proposed system in Section 3. The averaging concept and its related calculations are discussed in Section 4. Results, conclusions, and suggestions for future work are given in Section 5.

2. RELATED WORK

Research has been performed on the creation of new songs or components of a song, e.g., melody, using existing songs. Melody morphing is a representative technique to fuse two melodies. Hamanaka *et al.* [6] proposed a method to morph two melodies using generative theory of tonal music (GTTM) [7]. The GTTM makes it possible to morph melodies based on notes common to two melodies. However, requiring common notes makes it difficult to generate a fused melody from any two input melodies. In addition, the melody morphing is the method to naturally transform one melody to another melody seamlessly which does not

[1] In this paper, the term "average" refers to the equally mix and the term "in-between" refers to the arbitrary mix.

focus on the intermediate melody. Therefore, our proposed method constructs songs using an averaging operation to generate in-between notes rather than considering generative music models which is used in a melody morphing method. Melody morphing can generate a new melody by fusion; however, our goal is to generate a completely new song. Therefore, we consider fusing rhythm components, e.g., drum sequences and melody.

Wooller and Brown described a music morphing method in their survey paper [8]. The approaches introduced in that paper describe a morphing method for the essence of music rather than morphing the entire song. To the best of our knowledge, the only system that actually morphs songs is *MMorph*, which was proposed by Oppenheim [9]. MMorph provides a music morphing interface with an input of up to four songs and several user input morphing algorithms for each music element. Although the user input can help people fuse music, it may complicate the fusion experience. Therefore, we realize the fusion of songs without the selection of detailed parameters. The proposed system requires only the selection of songs and a mixing rate; thus, users are free to fuse songs intuitively.

Reusing existing content to make new music content is a style of song fusion. For example, Hoffman *et al.* [10] turned a Young MC song into an MC Hammer song using spectral matching with Markov chain Monte Carlo sampling. This approach of taking databases of recorded sounds and attempt to combine them to produce a sound matching a target specification is called audio mosaicing. Hoffman's method is a probabilistic approach that realizes audio mosaicing. In audio mosaicing approach, the audio signals of songs are considered; however, melodies or symbolic notes are not considered. To generate a new song, more semantic and symbolic information such as melody should be considered. To handle melody, we use the MIDI file format as an input music sample.

Reusing existing content to make new content is an established approach in content creation. Some research has generated new music videos by mixing existing music video content. Hirai *et al.* [11] proposed a mashup music video generation system that reuses video content based on audio-visual synchronization. Nakano *et al.* [12] proposed *DanceReProducer*, which is a mashup music video authoring system that employs a statistical audio-visual model. Note that the mashup of content is suitable for beginners, and Davies *et al.* [13] proposed *AutoMashUpper*, a system to mashup multiple songs according to a mashability measure. However, such mashup approaches reuse original content; therefore, the resulting content is a mixture of the original content rather than new and fused content. We aim to make new content which inherit the mood of original songs. In this context, the difference between new and original may seem ambiguous. However, through fusion, the songs MusicMean can generate will bridge the boundary of new and original songs.

Music generation methods based on statistical models have also been proposed [14, 15]. This approach generates new song based on a generative model constructed from training data (i.e., songs). However, such systems are limited in how they can reflect user preference because such statistical models cannot be constructed intuitively. Taking these factors into account, our goal is to realize a system that can fuse existing songs intuitively with minimal user input. The proposed system uses a mathematical averaging operation for the fusion of songs rather than the construction of a model for content generation.

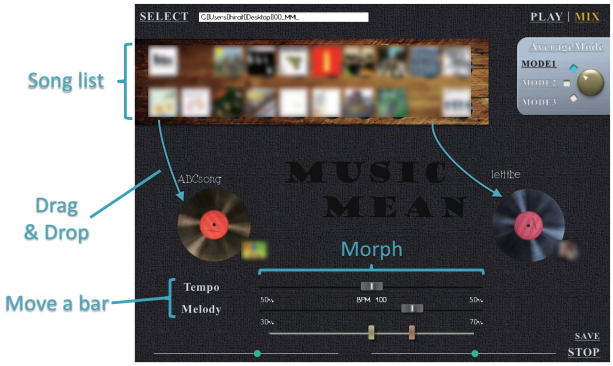


Figure 1. MusicMean screen capture.

3. PROPOSED SYSTEM

Fig. 1 is a screen capture of the proposed MusicMean system. In MusicMean, the MIDI files of each song and a blend rate, which represents a ratio of fusion, are the inputs. The system calculates in-between musical notes using the notes from the source song and the blend rate. Once the user has selected a song from a list, the system plays it. After adding more song and providing a blend rate, the proposed system calculates the in-between musical notes and plays the resulting in-between song in real time. The user can listen to the in-between song and interactively change the blend rate until the system plays a song that satisfies the user. The user can create an in-between song by simply dragging and dropping the songs they wish to fuse and setting the blend rate with a sliding bar. There are two blending parameters that the user can tune, i.e., tempo and melody. By moving a sliding bar, the user can morph songs from one to the other in an aspect of tempo and melody respectively.

MusicMean can also fuse more than two songs. Fig. 2 shows a screen capture of the multisong mixture mode. Each song is allocated to a vertex of an n-sided polygon, and the user can mix songs by moving a control point. The multisong mixture mode allows users to generate an average song for a specific musician or album.

4. FUSION METHOD

We use MIDI files as input to MusicMean. As a result, we do not have to consider sound source separation. MusicMean fuses melody and rhythm parts separately by using an averaging operation to generate an in-between song. The system calculates the in-between musical notes from the input MIDI files and the user-specified blend rate.

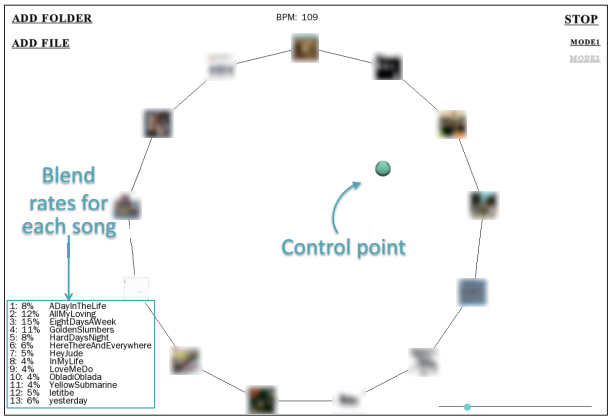


Figure 2. Generating an average song with more than two songs (Making an average song of a specific artist.)

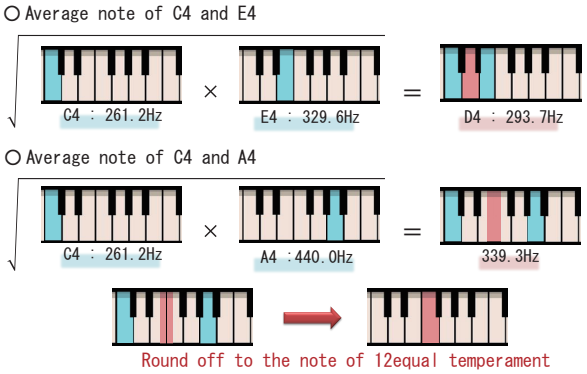


Figure 3. Musical note averaging operation.

4.1 Scope of consideration in MusicMean

Because we handle MIDI files as input, the output sound is also in MIDI format. Therefore, we cannot consider some elements such as timbre of the songs. Although the instrument type can be considered by a MIDI parameter, we are planning to handle these factors in our future research and only focus on the song itself for the current version.

Therefore, MusicMean does not handle singing. In addition, the number of instruments between fusing songs should be same in current implementation. Accordingly, we define the "song" as the music composed of one or more melody tracks (instruments) and a rhythm track.

4.2 Musical note averaging operation

When the user-specified blend rate is 0.5, the system generates average musical notes based on a geometric mean operation. An average note can be calculated using the pitch f_1 of a note of one song and pitch f_2 of a note of another song. The frequency (pitch) $\overline{f}$ of the average note is calculated using the following equation.

$$\overline{f} = \sqrt{f_1 \times f_2} \tag{1}$$

For example, for C4 (261.2 Hz) and E4 (329.6 Hz) notes with a blend rate of 0.5 (50%), the average note will be 293.7 Hz, which is the pitch of D4. Here, the average frequency may be a sound that is not associated with a musical note. In this case, the note will be rounded off to the nearest musical note in 12 equal temperament. Fig. 3 illustrates the averaging of musical notes.

This averaging operation can be extended to generating an in-between note. An in-between note can be calculated using pitch f_1 of one song's note and pitch f_2 of another song's note. With the blend rate α, the frequency (pitch) f' of the average note is calculated using the following equation.

$$f' = f_1^{\alpha} \times f_2^{1-\alpha} \tag{2}$$

When the blend rate α is 0.5, Eq. (2) corresponds to the geometric mean. Here, the output frequency value will be rounded off to obtain a musical note.

4.3 Melody averaging operation

To generate an in-between song, we must consider melodies. Fig. 4 shows the flow for handling melodies. To apply the averaging operation, the system first decomposes all notes of each song into sixteenth notes to align the length of all notes. The proposed system then compares each sixteenth note from the beginning of the both songs. After the averaging operation is performed, the system recombines all the sixteenth notes such that the length of each note is as close as possible to the shortest note among the original notes.

4.4 Generating musical sound

By applying only the averaging operation, the obtained in-between melody will be a series of notes that may not seem to be arranged as a piece of music. MusicMean makes the resulting sound more musical by considering basic music theory. Before calculating an in-between frequency, the system estimates the musical key of the in-between song. By acquiring a histogram of each note from a song, a chromagram for each song can be generated. The weighted sum of the chromagram corresponds to the chromagram of the in-between song. Here, the weight is the blend rate and the musical key of an in-between song is determined based on the top seven notes of the chromagram.

Using the generated musical key, the system estimates chords in each musical bar with reference to music theory. Thus, a musical key of whole song and chords for each bar of music can be determined. Finally, the system rounds off the in-between frequency to the pitch of nearest musical note that composes the chord in each musical bar. Thus, all musical notes in an in-between song become a note from 12 equal temperament and the melody of an average song adheres to the basics of music theory which follow the chord.

4.5 Drum averaging operation

The system also calculates average drum patterns by generating a mixture distribution of probabilities for each type of drum in each song. Drum sounds do not represent a specific musical scale (there are exceptional cases such as the vibraphone or the tom-toms). Therefore, the averaging operation for drum parts differs from the above-mentioned

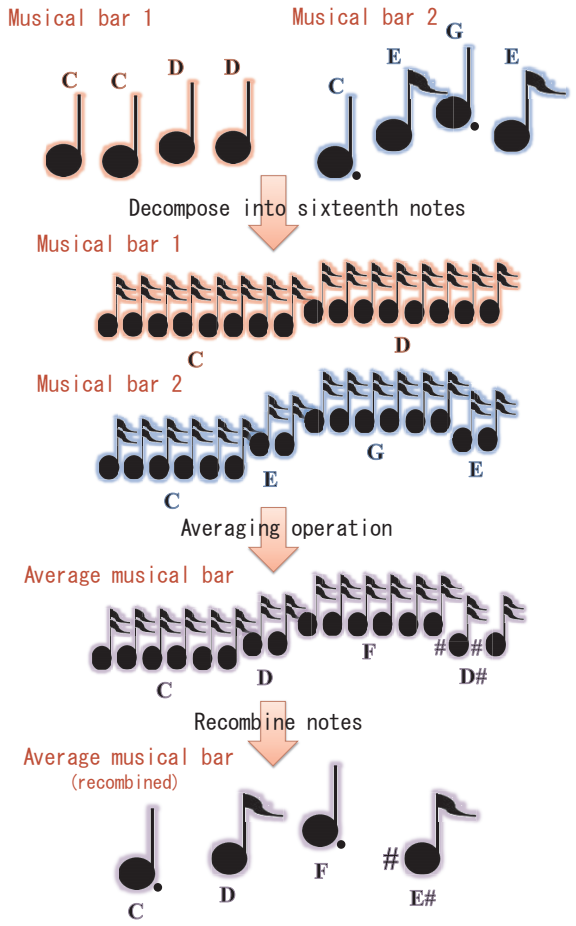


Figure 4. Melody averaging operation.

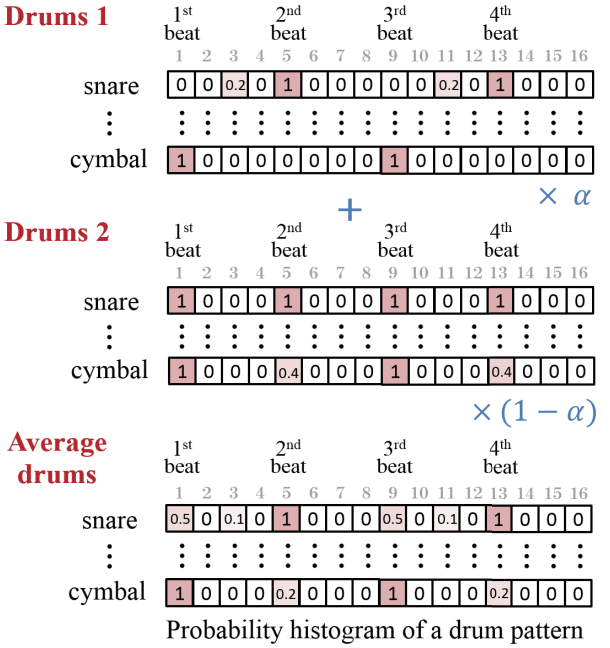


Figure 5. Drum pattern averaging operation.

method. We tested several approaches for averaging drum patterns. We selected a method in which the overall drum pattern does not differ significantly in each musical bar but only changes slightly at regular intervals.

Fig. 5 illustrates the averaging of drum patterns. We describe the drum pattern as a binary sequence of sixteenth beats (0 refers to silence; 1 refers to a sound). For each type of drum, the system generates a drum pattern histogram. The drum pattern histogram describes the time at which a drum sound is produced in a given musical bar. Note that the number of histogram bins is 16. By calculating the weighted sum of the histograms, an in-between drum pattern histogram can be obtained. This histogram indicates the probability of when a drum sound will be produced in a musical bar. If the value is 0.5 for a given timing, the drum will sound at that timing at a 50% probability.

Using this drum pattern histogram, the overall drum pattern will be similar; however, it will differ slightly in each musical bar. In the current implementation of MusicMean, the system does not consider differences in time signature.

Thus, the proposed system outputs an in-between song by calculating the in-between note for each instrument and the in-between drum pattern for each sixteenth note step. This process of generating an in-between song is relatively simple and can run in real time, which allows the user to create and listen to new fused songs interactively.

4.6 "In-betweening" of more than two songs

An in-between song generated from more than two songs can also be obtained as an extension of the above processes. For melody fusing, the in-between frequency (pitch) X of more than two songs can be calculated with the following equation.

$$X = a^{\alpha} \times b^{\beta} \times \cdots \times z^{\zeta} \quad [Hz] \tag{3}$$

Here, a, b, and, z are pitch values, and α, β, and, ζ are blending rates.

For musical key estimation, chord estimation, and drum pattern histogram calculation, the weight in a weight sum corresponds to the blend rates for each song. Note that the sum of the blend rates totals 1.

4.7 Extrapolation song

In the above, we have described a method to generate an in-between song, which corresponds to an interpolation of songs. MusicMean can also generate an extrapolation song, in which a factor of song B can be subtracted from song A. The process for generating an extrapolation song is simple. By setting blend rate to $\alpha > 1.0$ or $\alpha < 0.0$, the system can calculate an external melody and external rhythm pattern.

5. RESULTS AND CONCLUSION

MusicMean allows users to create personalized songs from a selection of songs and blend rate tuning. A song created using the proposed system preserves the characteristics of the original songs. This means that if the user wants to preserve the essence of one song while integrating another song, they can achieve this by manipulating the blend rate by moving a sliding bar. For example, the user can make

rock music quieter by blending ballads. In addition, an average song for a specific musician or album can be generated with MusicMean. The evaluation of the in-between songs and the system is our future work. Early reactions to the proposed system are indicating that users can feel the essence of each original song. However, generated songs often sound strange so that the improvement of the generation quality is a important subject that we have to tackle. In the future, we plan to further analyze in-between songs as well as the evaluation.

Note that an in-between song produced by MusicMean demonstrates some common traits. For example, even after tonality is considered, many minor notes are generated by averaging operation which is not so common in human generated songs. Therefore, MusicMean tends to generate strange melodies. We are planning to solve this problem by considering constraint of more detailed music theory. Another trait is that the melody of an in-between song tends to be flat when mixing many songs because the average converges when many samples are considered. In this case, a statistical model such as HMM based music generation [15] may be better suited for this purpose than the MusicMean fusion approach.

The concept of an average song and an in-between song has great potential for user-personalized music generation. Note that the proposed system represents only an initial phase of our research. We believe that there may be a better method to generate an average song. For example, modulating the keys of an original song, which was not considered in the current study, may be useful for generating a more effective average song. The proposed approach preserves the mood of an original song well; however, there may also be better ways to achieve this. Further the exploration of the averaging method and ongoing development of the proposed system are planned for future work.

As people change the flavor of a dish to their own taste by seasoning, digital content can be equally flexible. MusicMean demonstrates a potential to lead us to next-generation content production and music consumption.

Acknowledgments

We thank Tsukasa Fukusato and Hayato Ohya (Waseda University, Japan) for their advisory about musical theory for MusicMean. This work was supported by IPA Exploratory IT Human Resources Development and partially supported by OngaCREST, CREST, JST

6. REFERENCES

[1] T. Blaine and S. Fels: Collaborative musical experiences for novices, *Journal of New Music Research*, 32.4, 2003, pp.411–428.

[2] A. Andrea, and D. Ghisi: A Max Library for Musical Notation and Computer-Aided Composition, *Computer Music Journal*, 2015, pp.11–27.

[3] J.B. Maxwell, A. Eigenfeldt, and P. Pasquier: ManuScore: Music Notation-Based Computer Assisted Composition, Ann Arbor, MI: MPublishing, University of Michigan Library, 2012.

[4] J. Bresson, M. Stroppa, and C. Agon: Symbolic Control of Sound Synthesis in Computer-Assisted Composition, in *Proc. of ICMC*, 2005, pp.303–306.

[5] G. Assayag, C. Rueda, M. Laurson, C. Agon, and O. Delerue: Computer-assisted composition at IRCAM: from PatchWork to OpenMusic, *Computer Music Journal*, 23.3, 1999, pp.59–72.

[6] M. Hamanaka, K. Hirata, and S. Tojo: Melody morphing method based on GTTM, in *Proc. of ICMC*, 2008, pp.155–158.

[7] F. Lerdahl, and R. Jackendoff: An overview of hierarchical structure in music, *Music Perception*, 1983, pp.229–252.

[8] R.W. Wooller and A.R. Brown: Investigating morphing algorithms for generative music, *Third Iteration: Third International Conference on Generative Systems in the Electronic Arts*, 2005.

[9] D. Oppenheim: Demonstrating MMorph: a system for morphing music in real-time, in *Proc. of ICMC*, 1995.

[10] M. D. Hoffman, P. R. Cook, and D. M. Blei: Bayesian spectral matching: Turning Young MC into MC Hammer via MCMC sampling, in *Proc. of ICMC*, 2009.

[11] T. Hirai, H. Ohya, and S. Morihsima: Automatic Mash up Music Video Generation System by Perceptual Synchronization of Music and Video Features, in *Proc. of SIGGRAPH* posters, 2012.

[12] T. Naknao, S. Murofushi, M. Goto, and S. Morihsima: DanceReProducer: An Automatic Mashup Music Video Generation System by Reusing Video Clips on the Web, in *Proc. of SMC*, 2011, pp.183–189.

[13] M. E. P. Davies, P. Hamel, K. Yoshii and M. Goto: AutoMashUpper: Automatic creation of multi-song mashups, *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(12), 2014, pp. 1726–1737.

[14] J. Sneyers, and S. D. Danny: APOPCALEAPS: Automatic music generation with CHRiSM, in *Proc. of ISMIR*, 2010.

[15] D. Conklin: Music generation from statistical models, in *Proc. of AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, 2003, pp.30–35.

DESIRABLE ASPECTS OF VISUAL PROGRAMMING LANGUAGES FOR DIFFERENT APPLICATIONS IN MUSIC CREATION

Antonio Pošćić
Faculty of Electrical Engineering and Computing, Zagreb, Croatia
antonio.poscic@fer.hr

Gordan Kreković
Faculty of Electrical Engineering and Computing, Zagreb, Croatia
gordan.krekovic@fer.hr

Ana Butković
Faculty of Humanities and Social Sciences, Zagreb, Croatia
abutkovi@ffzg.hr

ABSTRACT

Visual programming languages are commonly used in the domain of sound and music creation. Specific properties and paradigms of those visual languages make them convenient and appealing to artists in various applications such as computer composition, sound synthesis, multimedia artworks, and development of interactive system. This paper presents a systematic research of several well-known languages for sound and music creation. The research was based on the analysis of cognitive dimensions such as abstraction gradient, consistency, closeness of mapping, and error-proneness. We have also considered the context of each analyzed language including its availability, community, and learning materials. Data for the research were collected from a survey conducted among users of the most notable and widespread visual programming languages. The data is presented both in raw, textual format and in a summarized table view. The results indicate desirable aspects along with possible improvements of visual programming approaches for different use cases. Finally, future research directions and goals are suggested in the field of visual programming for applications in music.

1. INTRODUCTION

The usefulness of visual programming for various domains has been researched extensively as well as the motives and reasons which typically drive users to adopt and adapt to visual programming [1]. Research shows that visual paradigms are better suited for novice users and are easier to understand for non-expert users in general. The most often cited reasons for this are the inherent predisposition of the human psychology towards visual, instead of textual representations and the inclusion of domain specific, immediately understandable syntactic elements such as musical instruments [2]. However, the specific links between artists, in this case musicians, and visual programming has not been explored extensively. We think that this is something worth researching as the field itself is interesting and unique, a touching point between arts and engineering, and the visual approach seems to be quite fruitful and well adopted in the field [3]. Languages and tools for computer composition, music creation, and sound synthesis are predominantly of the visual kind and it has been shown that musicians avoid conventional textual languages [4].

The aim and purpose of this paper is to better understand why musicians find visual programming so appealing, which aspects of existing languages they find most useful and for which applications, and what problems do they encounter while using them. Primarily, we wanted to identify why certain languages seem well-suited for specific applications in music, from sound synthesis to algorithmic composition. In a way, this work can be seen as the first step on the journey of trying to understand how and why certain paradigms and approaches inspire artists when dealing with computer created music. The impact of tool choices on creativity is also briefly considered. Additionally, in this paper we present a succinct cross-examination and comparison of selected languages and tools based on their design choices.

Since neither hard data nor related research existed that would help us determine which specific properties and aspects of visual programming users find appealing, we conducted a survey among the users of the selected five most significant languages. The user base we surveyed was predominantly comprised of amateur musicians. The collected data and extrapolated knowledge presented in this paper has been judged, internally, against the cognitive dimensions framework [1]. We used the framework to try and identify key concepts of these languages and we employed its principles to help us deduce the positives and negatives of each aspect of visual programming languages. Detailed analysis based on the cognitive framework is beyond the scope of this paper.

The rest of this paper is divided as follows: the second chapter gives a basic overview of the programming languages and tools that we included in our research. The third chapter discusses the methodology used to collect the data and gives insights into the users' responses. In the fourth chapter we try to identify what and why users value in programming languages for music creation. In closing, we propose topics for future research.

2. VISUAL PROGRAMMING AND MUSIC

As stated previously, visual programming is often used in computer-supported music creation processes. From synthesizing basic sounds to algorithmic composition and complex audiovisual works, the number and type of applications are practically limitless. For our questionnaire we selected five languages and tools that cover all of

Copyright: © 2015 First author et al. 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.

these purposes, but during the selection process we also considered their reach and influence. The final selection was reduced to the following five languages and tools: Pure Data, Max/MSP, OpenMusic, Symbolic Sound Kyma, and Native Instruments Reaktor.

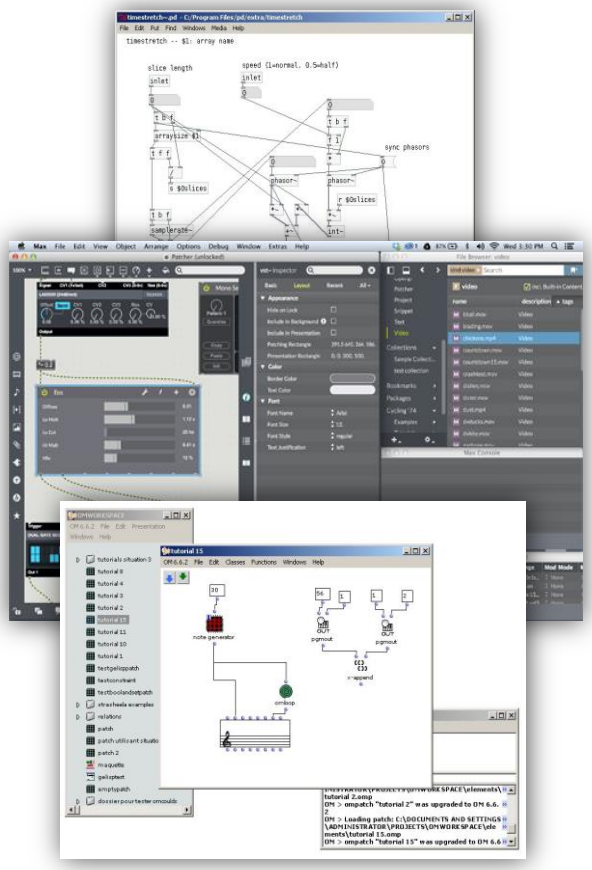


Figure 1. Pure Data, Max/MSP [5], and OpenMusic

In the following text as well as in the surveys, we used nomenclature tied to each of the individual languages. In other words, terms like "class", "object", and "sound" may refer to very similar concepts in the context of Pure Data, OpenMusic, or Kyma respectively. We decided that this approach would help minimize any misunderstandings and uncertainties that would have been introduced by a generalized, common codification.

2.1 Pure Data

Pure Data [6] is an open source visual programming language geared towards sound processing and generation, music creation, and artistic multimedia work in general.

On a paradigm level, Pure Data is a dataflow programming language. The flow consists either of control messages or audio signals which are processed by native objects providing functionalities ranging from simpler (mathematical operations and general programming functionalities) to complex digital signal processing. The feature set available through native objects and extensions called "externals" is very wide making Pure Data usable not only as a language for creating sounds, videos, and other multimedia, but to create fully fledged programs.

This flexibility is twofold. On one side, it is powerful and enables its users to make anything they want, but on the other side it requires a better knowledge and understanding of programming on a lower level. This is especially clear when taking into consideration the signal flow it is based on. Further positives include the variety of available patches and the possibility of interfacing with various other hardware and software tools, libraries, and systems making it suitable for creating interactive systems. Among other negatives, a certain lack of visual polish stands out as well as rudimentary support for higher level functionalities and out of the box functionalities.

2.2 Max/MSP

Max/MSP is a language that is superficially fairly similar to Pure Data since they stem from the same author. Max/MSP and Pure Data fall under the moniker of "patcher" languages [7], and they both share significant features and use cases, while portions of their code can even be interoperated and used with either of the programming languages. One important benefit of Max/MSP is a user interface with many usability improvements, better organization of components, and the availability of additional tools such as tags and searches which improves programming efficiency. The graphical design with customizable elements contributes to better intelligibility and overall impression of the workspace. Some objects have a rich visual representation which illustrates nicely the purpose of the object and sometimes even allow interactivity over its parameters. Since Max/MSP is nowadays a proprietary and commercial product [5], the user base is somewhat smaller, but the users in turn benefit from improvements not available in Pure Data.

2.3 OpenMusic

In basic terms, OpenMusic [8][9] is an object-oriented visual programming language. While it possesses a number of functionalities that make it usable as a general purpose language, Open Music is a tool heavily adapted for algorithmic composition and automation of composition techniques. Standout features include objects that resemble the traditional musical notation as well as a variety of objects that diverge from the underlying signal flow paradigm and enable the manipulation of sounds in time. Being based on Lisp, it retains many of its characteristics.

While not as flexible nor extensible as Pure Data or Max/MSP, it provides many additional features that might make it easier to use when dealing purely with music composition. Users well-versed in Lisp might also find OpenMusic appealing because of the modifications made available through core changes.

2.4 Symbolic Sound Kyma

Kyma, a "sound design" graphic language, is geared primarily towards live, interactive sound generation and manipulation, and post-production [10]. Being concentrated mainly on the manipulation of sounds and disre-

garding other elements of multimedia, it is primarily suited for music composition and a variety of music manipulation techniques.

The paradigm includes many aspects and additional tools that are not available in competing products making it easier to use, more streamlined, and less error prone. Especially notable is the timeline that enables the manipulation of sounds in the domain of time, similar to those found in digital audio workstation software.

Its reach and acceptance were influenced, since the very beginning, by the cost of ownership and mandatory hardware associated with the software. Because of those factors, Kyma has been and, to a lesser degree, still is used mainly in the Academia.



Figure 2. Symbolic Sound Kyma [11] and Native Instruments Reaktor [12]

2.5 Native Instruments Reaktor

Reaktor is a modular studio for designing sound synthesizers, sequencers, samplers, audio effects, and other sound design tools. Unlike all previously mentioned tools and languages, Reaktor is not intended for creating music compositions, interactive systems, or multimedia works. It is focused on developing objects for sound synthesis and processing which then can be combined and superimposed. Additionally to objects created using the visual paradigm on several different abstraction levels, Reaktor also provides the possibility to view and edit building blocks using textual programming. Sound synthesizers and effects developed in Reaktor can be used within host sequencers and digital audio workstations. This kind of a toolset clearly suits a different target audience when compared to the other described languages.

3. RESEARCH AND FINDINGS

In this chapter, after first explaining the methodology used in our research, we present the survey results that point towards some interesting aspects and tendencies regarding the selected languages. Further analysis and conclusions are laid out in the subsequent chapter.

3.1 Methodology

The data presented in this paper was collected in four phases. During the first phase, we created questionnaires for the selected visual programming languages and tools based on existing research and our own previous analysis. The questions were modeled so that they would provide us with knowledge about certain interesting aspects of these languages and their impact on users.

To improve the questionnaires, during the second phase we conducted interviews with experienced users for each of the selected programming languages. The collected data helped us to ensure the correct terminology, intelligibility of questions, and relevance of the answers offered for closed-type questions.

The third phase included posting the questionnaires on a variety of appropriate community web pages, forums, and social networks to be filled by users. The questionnaires were available for one week for collecting answers.

The fourth and final phase consisted of data analysis. For this purpose, we used IBM's SPSS Statistics software package [13]. Each questionnaire was processed individually and no cross-analysis was performed between the sets of data. The output of the software package has been manually corrected for certain outliers.

Also worth noting is that the questions in the questionnaires were divided into two sets. The first set of questions included common questions that were the same for each of the languages. These dealt with demographic data, how the users became accustomed with the specific language, inquiries about other tools that they might have used, etc. On the other hand, the second set was comprised of questions tailored around the specificities of individual languages. For example, the questionnaire for Native Instruments' Reaktor did not include questions about algorithmic composition since it cannot support such constructs.

3.2 Pure Data

There were 56 participants who answered questions about Pure Data, 91% male, aged between 17 and 59 (M=31.37, SD=9.64), with 68% having a formal music education. They found out about Pure Data from a friend or colleague (48%), during university (19%), from a magazine, journal, or a book (14%), and online (14%). They have been using it for more than 2 years (55%), between 1 and 2 years (20%), between 3 months and a year (16%), and less than 3 months (9%). They have been using it every day (18%), few times a week (36%), few times per month (34%), and few times per year (12%). They have made between 0 and 100 works (M=14.49, SD=17.52). Most of them said that it took a few weeks (34%) or a few months (34%) to become productive, for 14% it took a few days and for 18% a year or more. They use Pure Data mostly

for creating interactive systems (70%), musical compositions (68%), and audio effects (66%), automating composition techniques (46%), creating synthesizers that are subsequently used in compositions (43%), and audiovisual works (38%). Participants use Pure Data to create algorithmic compositions (63%), laptop improvisations (59%), acousmatic music (41%), and accompaniments for acoustic instruments or ensembles (36%). They chose Pure Data because, in comparison to other languages/tools, it is better for their needs (38%), they understand it better (34%), it is simpler to use (20%), or because it is free (14%). They have learned how to use it from online tutorials (82%), from a book (34%), during a university course (16%), or with a help of colleague (16%). Learning became easier or easy after they learned the basics (88%). Most of them are satisfied or somewhat satisfied with the visibility and legibility of their programs (61%) and think that other users would significantly or completely understand their programs (59%). They think the most inspiring aspect of Pure Data are the flexibility of the language (support for various usages) (46%), the community and support provided (21%) and cross-platform support (16%). They think there is a subtle (39%) or moderate (38%) difference between their personal way of thinking and the concepts used in Pure Data, and they think what is missing in Pure Data is a temporal dimension (39%), visualization of a timbral characteristics and changes in sound (39%) and visualization of sound spatialization (34%). The most tedious aspect for 34% is program flow control, for 23% synchronization of events in time, for 20% mapping and scaling numerical values and for 16% analysis, synthesis, and processing of audio signals. There was no consensus on limitations. Most of Pure Data users (61%) are also acquainted with Max/MSP, 27% also know NI Reaktor, while 25% know only Pure Data.

When programming in Pure Data, participants have controls and some other pieces of the patch opened in the main patch (39%) or only controls (30%). 71% of them use example patches or other people's patches and 61% annotate them sometimes or often, with 36% saying they use comments for annotations. Most of them think they make mistakes sometimes (55%), that it is moderately difficult to notice errors and mistakes (55%), and that it is moderately difficult to understand roles of certain portions of program (66%). Also, most of them think they have to rearrange elements often or always (70%), and that it is easy or moderately difficult to modify existing programs (71%). For 55% of participants, components sometimes behave in unwanted ways, they test unfinished patches often or always (74%), and find this a useful or very useful possibility (93%). 48% of them think that the debugging tool's limitations hinders the evaluation of unfinished programs, while 39% think that there is a need for temporarily placed connectors and objects.

3.3 Max/MSP

There were 18 participants who answered questions about Max/MSP, they were all male, aged between 22 and 57 (M=33.44, SD=9.67), and 60% had formal music education. They found out about Max/MSP from a magazine, journal or a book (28%), during university (28%), online (22%), from a friend or colleague (17%), and during conferences (5%). They have been using it for more than 2 years (67%), between 1 and 2 years (22%), and less than a year (11%). They have been using it every day (44%), few times a week (17%), and few times per month (33%). Only one participant stated that he uses it few times per year. They have made between 0 and 200 works (M=24.56, SD=46.39). Most of them said that it took a few weeks to become productive (39%), for 33% it took a few months, for 17% a few days, and for 11% a year or more. They use Max/MSP mostly for creating musical compositions (78%), creating audio effects and automating composition techniques (50%), creating interactive systems (44%), audiovisual works (39%), and synthesizers later used in compositions (33%). Participants make algorithmic compositions (67%), laptop improvisations (44%), acousmatic music (28%), and accompaniment for an acoustic instrument or ensemble (11%). They chose Max/MSP because they understand it better than other languages (61%), it is better for their needs (44%) or it is simpler to use (28%). They have learned how to use it with online tutorials (61%) and during a university course (39%), and learning became easier or easy after they learned the basics (94%). 33% of them are satisfied with using Max/MSP, 44% somewhat satisfied, and 22% somewhat unsatisfied. Most of them think that other users would somewhat understand their programs (50%), with additional 39% thinking that other users would understand them significantly or completely. They think the most inspiring aspect of Max/MSP is the flexibility of the language (support for various usages). The most important limitation for 50% is the lack of explicit temporal dimension and for 44% the lack of appropriate tools for exporting MIDI/sheet music. In line with that, 44% say what is missing in Max/MSP is a temporal dimension, 33% say visualization of a timbral characteristics and changes in sound, and 22% visualization of sound spatialization. The most tedious aspect for 39% is synchronization of events in time, for 22% program flow control, and for 17% analysis, synthesis, and processing of audio signals. Most of them (44%) know only Max/MSP, 39% know also Pure Data and 28% know also NI Reaktor.

Participants differed in how they would describe the difference between their visual representation and the one used in Max/MSP: 16% think there is no difference, 28% think there is a subtle, 28% there is a moderate, and 28% there is a significant difference. When working with Max/MSP, most of them (56%) have controls and some other pieces of the patch open in the main patch. 61% of them use example patches or other people's patches and annotate them sometimes or often, with 39% saying they use comments for annotations. Most of them think it is moderately difficult to notice errors and mistakes (72%) and to understand roles of certain portions of program (77%). Also, most of them think that sometimes or often they make mistakes (84%) and have to rearrange elements (77%), and that it is easy or moderately difficult to modify existing programs (77%). For 56% components sometimes behave in unwanted ways, they test unfinished patches often or always (83%) and find this a useful or

very useful possibility (83%). 56% of them think that the debugging tool's limitations hinder the evaluation of unfinished programs, while 33% think that there is a need for temporary connectors and objects.

3.4 OpenMusic

Due to the limited number of responses to this questionnaire, the data presented for OpenMusic is statistically unreliable, but it nonetheless helps better understand some aspects of the language.

There were 3 participants who answered questions about OpenMusic, they were all male, aged between 24 and 51, and all had high levels of formal music education. They found out about OpenMusic during university (67%) and from a friend or colleague (33%). They have been using it from more than 3 months to more than 2 years. They use it from few times per month to everyday. They have made between 1 and 20 works. It took a few weeks for them to become productive. They use OpenMusic for creating musical compositions (acousmatic music and algorithmic compositions) and automating composition techniques. They chose OpenMusic because it is simpler to use and better suited for their needs. They have learned how to use it by consulting written online tutorials and books, and they find that learning became easier after becoming acquainted with the basics. They have different opinions about whether other users would understand their programs, ranging from not at all to completely. Each participant named different limitations of OpenMusic and all of them have different ideas about what is missing from OpenMusic's visual representation, but two of them think that most tedious aspect is the reliance on program flow control. Two of them also think that the most inspiring aspect are visual representations which include sheet music. They all also know Max/MSP, two of them also know Pure Data, and one also knows NI Reaktor.

As mentioned before, the number of respondents was small, so we will only report what the respondents agreed on when discussing the process of working in OpenMusic. They think it is easy or moderately difficult to notice errors and mistakes, that it is moderately difficult to modify existing programs, and that it is useful or very useful that they can evaluate unfinished programs, with debugging tool's limitations hindering the evaluation of unfinished programs the most. In their opinion, programs sometimes behave in unwanted ways and they find it moderately difficult or difficult to understand the roles of certain portions of the program. They are somewhat satisfied with OpenMusic, they often use existing functions, and always use existing classes. They find the possibility to choose icons useful, but that the purpose of existing classes is somewhat unclear from their icons.

3.5 Symbolic Sound Kyma

Similar to OpenMusic, we received a limited number of responses to the questionnaire for Kyma. While it remains statistically unreliable, our confidence in the collected data is higher due to interviews conducted with users who are Kyma experts, highly educated, and have been using the system for many years (more than 10, in some cases).

There were 5 participants who answered questions about Kyma, they were all male, aged between 39 and 61 (M=44.60, SD=9.24), and 60% had formal music education. They found out about Kyma from a magazine, journal, book or radio program (40%), during university courses (40%), and during research (20%). They have been using it either more than 2 years (80%) or between 1 and 2 years (20%). They mostly use it a few times per year (80%) and only one participant stated that he uses it every day. They have made between 3 and 300 works (M=69.20, SD=129.35). It took a few weeks to become productive for 40%, a year or more for 40%, and a few days for 20% of the participants. They use Kyma mostly for creating musical compositions (100%), making synthesizers that are subsequently used in compositions (60%), audio effects (40%), and interactive systems (40%). Participants mostly make acousmatic music (60%) and accompaniment for acoustic instruments or ensembles (60%). They chose Kyma because it is better suited for their needs (60%). They have learned how to use it mostly from a book (80%) and learning became easier or easy after they learned the basics (60%); 40% of them find that other users would completely understand their programs. Each participant named different inspiring aspects and limitations of Kyma, but 40% named parameter automation, 40% the possibility of creating new from existing objects, and 20% mentioned the large number of existing sounds as the most useful aspect of Kyma. 40% of the respondents are also acquainted with NI Reaktor and 40% with Max/MSP.

Kyma users often use existing sounds (60%) and tools (60%), while they never or rarely use colors to annotate sounds (80%). 60% say it is easy to find the right module, 80% say they do not have problems finding the right module, and 60% say that do not have problems with the unavailability of required modules. Regarding errors, 40% say they never make them, 40% that they sometimes make them, 40% think errors and mistakes are very difficult to notice, and 40% sometimes rearrange modules. 80% find it easy to modify existing sounds and 80% think it is easy or moderately difficult to understand the roles of existing sounds. As for satisfaction with the visibility and legibility of their sounds, 40% are somewhat unsatisfied and 40% are somewhat satisfied.

3.6 Native Instruments Reaktor

There were 13 participants who answered questions about NI Reaktor. They were all male, aged between 20 and 59 (M=35.00, SD=12.77), and 54% had no formal music education. They found out about NI Reaktor from a friend or colleague (25%), magazine, journal or a book (25%), online (17%), through another product of the same company (17%), during university courses (8%), or conferences (8%). They have been using it more than 2 years (54%), between 3 months and a year (31%) and between 1 and 2 years (15%). They have been using it every day (38%), a few times a week (31%), and few times per month (23%). Only one participant stated that he uses it a few times per year. They have made between

3 and 352 works (M=56.46, SD=102.86). Most of them said that it took a few months to become productive (62%), for 23% it took a few weeks, and for 15% a year or more. They use NI Reaktor mostly for creating musical compositions (54%), audio effects (62%) and synthesizers subsequently used in compositions (46%). They chose NI Reaktor because it is simpler to use (23%), they understand it better (15%), or it is better for their needs (15%). They have learned how to use it mostly with online tutorials (77%), and learning became easier or easy after they learned the basics (93%). 62% of them are satisfied with using NI Reaktor and find that other users would significantly or completely understand their programs (85%). They use it because it enables them to create works which cannot be made in other tools (39%) and because of its flexibility and creative possibilities (31%). For 54% the most important limitation is that instruments cannot be compiled to work outside Reaktor (e.g. as VST instruments). Most of them (54%) know only NI Reaktor, and 38% know also Max/MSP.

As much as 77% of users often or always use existing ensembles and instruments, and the rest use them at least sometimes. For existing macros, 46% use them rarely or sometimes and 54% use them often or always. Existing cells are used often or always (77%). On the core level NI Reaktor is used rarely or sometimes (54%), but 39% of participants stated that they never use it on the core level. They also rarely or sometimes use play mode (77%), they think that mistakes and errors are only rarely or sometimes a consequence of mixing events and audio signals (70%), and for them most common mistakes are connecting polyphonic macros or modules to monophonic ones (23%) and normalization and scaling of numeric values (23%). When they create new instruments they sometimes or often (70%) encounter components that are not clear to them, they find it moderately difficult to find the components they need (62%), and 54% think that unclear functionalities of some components complicate the process of finding the right component. For 62% of the respondents, components sometimes behave in unwanted ways, it is moderately difficult to modify existing ensembles and instruments and to understand their purpose, and it is somewhat clear what are the functionalities and roles of components. When they want information about components, they search online tutorials (54%) or read user manuals (31%). When it comes to rearranging components, 31% of the participants rearrange them often, 23% always, and 23% sometimes. 70% of participants sometimes or often group their instruments, rewrite significant portions of their instruments, and make mistakes that require significant changes. 77% find it useful to evaluate unfinished instruments and 46% of them think that the need for temporary connectors and components hinders the evaluation of unfinished instruments, while 31% think it is the debugging tool's limitations. They mostly do not think that the experience of using Reaktor would improve if the flow was vertical rather than horizontal (70%).

3.7 Summary

To improve the readability of some key data, we include a summary, in the form of a table (Table 1), of the most important questions and answers. This summary improves the comprehension of the following chapter.

4. ANALYSIS, CONCLUSIONS, AND FUTURE WORK

Analysis of the collected data (see Table 1 for a summarized view) shows many interesting tendencies and patterns. Particularly, this research has confirmed that the visual nature of these languages removes hurdles between artists and their ideas, enabling them to become productive in a relatively short period.

On a more superficial level, it is clear that most users tend to commit to only one language and only sparsely use or try out alternatives, regardless of the type of task they are trying to accomplish. This is true even in those instances when better, more fitting languages are available. For example, while Kyma is best suited for sound design and OpenMusic is oriented towards algorithmic composition, users insist on using Pure Data for these applications. What is obvious from the results and answers is that Pure Data and Max/MSP users are not aware of commercial tools such as Kyma or of more recent and less widespread tools such as Open Music.

Based on the number of responses as well as other metrics such as the Alexa page rank or number of citations in publications, we can conclude that most users gravitate towards open source languages such as Pure Data which, in turn, has the most diverse and largest user base. The lower cost of tools or the availability of online literature can be reasons for that, but this can also be explained by the "do it yourself" nature of the language, the encompassing culture and community, and the vast range of functionalities provided. While there is no consensus as far as its limitations are concerned, it is obvious that many users resent the somewhat dated look and feel of the language/environment. Many of the issues that they mention, such as the lack of a timeline, have already been resolved in languages that they are not acquainted with (most often Kyma). Respondents point out that the lack of a temporal dimension and the over-reliance on signal flow also hinder productivity, but whether or not this has any bearing on the quality of their output cannot be determined. For many, Pure Data requires additional efforts to tackle and harness, asking for deeper knowledge about programming. The question is raised, in the end, whether these obstacles actually have any impact on creativity since almost all users seem satisfied with Pure Data in general.

Similar judgments can be made about the other languages as well. The survey results have shown that some common preconceptions are, indeed, correct. For example, when considering use cases, Pure Data is predominantly used for interactive systems and music composition, Max/MSP for automated composition techniques, Kyma for music compositions and creating synthesizers, Reaktor for sound effects, etc. Whether it is a question of marketing or truly related to the specifics of each language remains unclear, especially when considering the users' reluctance of trying out other software.

	Pure Data	**Max / MPS**	**Open Music**	**Kyma**	**NI Reaktor**
Number of participants	56	18	3	5	13
Age	range: 17 to 59 mean: 31.7	range: 22 to 57 mean: 33.4	range: 24 to 51 mean: 36.3	range: 39 to 61 mean: 44.6	range: 20 to 59 mean: 35.0
Experience in using the language	> 2 years: 55% 1-2 years: 20% < 1 year: 25%	> 2 years: 67% 1-2 years: 22% < 1 year: 11%	> 2 years: 2 < 1 year: 1	> 2 years: 80% 1-2 years: 20%	> 2 years: 54% 1-2 years: 15% < 1 year: 31%
Time to become productive	few days: 14% few weeks: 34% few months: 34% > 1 year: 18%	few days: 17% few weeks: 39% few months: 33% > 1 year: 11%	a few weeks	few days: 20% few weeks: 40% > 1 year: 40%	few weeks: 23% few months: 62% > 1 year: 15%
Frequency of usage	few times per: - day: 18% - week: 36% - month: 34% - year: 12%	few times per: - day: 44% - week: 17% - month: 33% - year: 1	from few times per month to everyday	few times per year: 80% every day: 20%	few times per: - day: 38% - week: 31% - month: 23% - year: 1
Purpose of usage	interactive sys., compositions, automation of comp. techniques	compositions, audio effects, automation of comp. techniques	compositions, automation of comp. techniques	compositions, synthesizers	audio effects, compositions, synthesizers
Knowledge of other languages	Max/MSP: 61% Reaktor: 27%	Pure Data: 39% Reaktor: 28%	Max/MPS: 3 Pure Data: 2 Reaktor: 1	Max/MSP: 40% Reaktor: 40%	Max/MSP: 38%
Frequency of making mistakes	never: 2% rarely: 18% sometimes: 55% often: 16% always: 9%	never: 0% rarely: 17% sometimes: 56% often: 28% always: 0%	rarely: 1 sometimes: 1 always: 1	never: 40% rarely: 0% sometimes: 40% often: 0% always: 20%	never: 8% rarely: 0% sometimes: 23% often: 31% always: 38%
Difficulty of noticing mistakes	easy: 11% moderate: 55% difficult: 30% very difficult: 4%	easy: 11% moderate: 72% difficult: 17% very difficult: 0%	easy: 1 moderate: 2	easy: 20% moderate: 20% difficult: 0% very diff.: 40%	N/A
Most tedious aspects	flow control, synchronization of events	synchronization of events, flow control	flow control	N/A	N/A
Most inspiring aspects	flexibility, community	flexibility	visual representation of sheet music	parameter automation, reuse of objects	N/A

Table 1. Important results obtained by the survey.

Additionally, while most users across the various visual programming languages share the same education and experience levels, the most startling discrepancies are found within Native Instruments Reaktor's user base. In general, they show a lack of formal education, they are even more heavily exclusive to Reaktor than is the general case, and they tend to shy away from using Reaktor's advanced functionalities such as editing the fundamental building blocks with textual programming.

Bearing that in mind, it is not surprising that users usually pick out the most apparent features of each language as those that most inspire them. Max and Pure Data experts note their flexibility, Kyma's user base insists on the potentials of its sound design tools (for example, parameter automatization and the spectrum editor), while OpenMusic users value its sheet input and maquette. We can ascertain that most tools have adapted to the needs of their users.

It appears that most musicians and artists do not think that any radical paradigm shifts are necessary when considering the tools that they use. Still, it is worth noting that many languages are evolving to include a variety of additional tools whilst the main, basic paradigm stays the same. One notable example is the timeline functionality that was added to Kyma in its later years. Refinements and improvements of existing mechanisms are sought out constantly. Another path worth exploring are specific tools that aid the main functionalities of programming languages such as the spectral editor in Kyma or the simple importer of multimedia files in Max/MSP. The questionnaire results confirmed the intuition that such specific tools are considered to be useful in practical and frequent tasks. Regarding other visual programming aspects, Kyma's resistance to errors, which stems from an inherent quality of its paradigm and makes it most suitable for inexperienced users approaching sound design, is one example of such improvements.

As far as problems with the languages are concerned, most of them are inherent to the visual domain [1]. Other than that, the lack of program control flow in Kyma and

Reaktor is identified as a problem by some of their users. While it seems that most of the described languages are relatively easy to start working with, more difficult tasks and complex works require a certain amount of experience. There are some further issues prompted by the users which are common to almost all of the analyzed languages. For example: the synchronization of events in time, the underlying signal flow, and the lack of temporal dimensions. Inevitably, some of these issues cannot be resolved without a paradigm shift. Because of that, we believe that new approaches and paradigms could be beneficial to artists' outputs if they offered attractive advantages and solved enough important problems to make them try something new. One such approach could include using a flow based on timbral attributes instead of a conventional signal flow [14].

In conclusion, several interesting points for future work are revealed. First of all, a deeper and more detailed cross-analysis of these languages based on the cognitive dimensions framework must be considered. This kind of broader inquiries could help pinpoint exact areas that could be improved as well as it could shine light on the means of improving them. Questions of creativity left unanswered by this paper should also become clearer. Secondly, and drawing from the first point, worth considering is research into paradigms that would take into account the historic and current developments in the field, but which would also try to fundamentally challenge and change some basic principles. Finally, issues related to gender inequality in the field (shown by our demographic data) and social and subcultural concerns, which might be tangentially related to questions of creativity, should be explored by researchers from appropriate fields.

5. REFERENCES

[1] T. R. G. Green and M. Petre, "Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework," in J. of Visual Languages & Computing, 1996, vol. 7, no. 2, pp. 131-174.

[2] K. N. Whitley, "Visual Programming Languages and the Empirical Evidence For and Against," in J. of Visual Languages & Computing, 1996, vol. 8, no. 1, pp. 109-142.

[3] A. Blackwell and N. Collins, "The Programming Language as a Musical Instrument," presented at the 17th PPIG Workshop, University of Sussex, Brighton, UK, 2005.

[4] M. V. Mathews and J. R. Pierce, "Composing with Computers: A Survey of Some Compositional Formalisms and Music Programming Languages," in *Current Directions in Computer Music Research*, CA: MIT Press, 1989, pp. 291-396.

[5] Cycling '74, "Max," Internet: https://cycling74.com/products/max/ [Jan. 6, 2015].

[6] M. S. Puckette, "Pure Data: Another Integrated Computer Music Environment," presented at the Second Intercollege Computer Music Concerts, Tachikawa, Japan, 1997.

[7] M. Puckette, "The Patcher," presented at the International Computer Music Conference, San Francisco, CA, 1988.

[8] G. Assayag, C. Rueda, M. Laurson, C. Agon, and O. Delerue, "Computer-assisted composition at IRCAM: From PatchWork to OpenMusic," in Computer Music Journal, 1999, vol. 23, no. 3, pp. 59-72.

[9] J. Bresson, C. Agon, and G. Assayag, "OpenMusic 5: A Cross-Platform Release of the Computer-Assisted Composition Environment," presented at the 10th Brazilian Symposium on Computer Music, Belo Horizonte, MG, Brazil, 2005.

[10] C. Scaletti, *Kyma: An Object-Oriented Language for Music Composition*. Ann Arbor, MI: MPublishing, University of Michigan Library, 1987.

[11] Symbolic Sound Corporation, "Kyma X Sound Design Playground," Internet: http://www.symbolicsound.com/cgi-bin/bin/view/Company/WebHome [Jan. 20, 2015].

[12] Native Instruments GmbH, "Reaktor 5," Internet: http://www.native-instruments.com/en/products/komplete/synths/reaktor-5/ [Jan. 20, 2015].

[13] A. Field, *Discovering Statistics Using IBM SPSS Statistics*. London: Sage, 2013.

[14] A. Pošćić and G. Kreković, "Controlling a Sound Synthesizer Using Timbral Attributes," presented at the Sound and Music Computing Conference, Stockholm, Sweden, 2013.

ONLINE HARMONIC/PERCUSSIVE SEPARATION USING SMOOTHNESS/SPARSENESS CONSTRAINTS

F. Canadas-Quesada, P. Vera-Candeas, N. Ruiz-Reyes
Univ. de Jaén, Spain
fcanadas@ujaen.es

P. Alonso
Univ. Politécnica de Valencia, Spain
palonso@dsic.upv.es

J. Ranilla
Univ. de Oviedo, Spain
ranilla@uniovi.es

ABSTRACT

The separation of percussive sounds from harmonic sounds in audio recordings remains a challenging task since it has received much attention over the last decade. In a previous work, we described a method to separate harmonic and percussive sounds based on a constrained Non-negative Matrix Factorization (NMF) approach. The approach distinguishes between percussive and harmonic bases integrating percussive and harmonic sound features, such as smoothness and sparseness, into the decomposition process. In this paper, we propose an online version of our previous work. Instead of decomposing the whole mixture, the online proposal decomposes a set of segments of the mixture selected by a sliding temporal window. Both percussive and harmonic bases of the next segment are initialized using the bases obtained in the decomposition of the previous segment. Results show that an online proposal can provide satisfactory separation performance but the sound quality of the separated signals depends inversely on the computation time of the system.

1. INTRODUCTION

Separating percussive sounds from harmonic sounds in music remains a challenging problem since it has received much attention over the last years. Percussive sounds, e.g. snare drum, are impulsive and have a structure that is vertically smooth in frequency and sparse in time. Harmonic sounds, e.g. bass, are quasi-stationary and have a structure that is horizontally smooth in time and sparse in frequency (see Fig 1). Several music information retrieval applications could benefit from this separation such as music transcription or onset detection.

Although many algorithms have been developed to separate percussive and harmonic sounds from monaural music [1] [2] [3] [4] [5], one of the trends in percussive and harmonic separation is based on the concept of anisotropic smoothness which is related to the difference in the directions of continuity between the spectrograms of harmonic and percussive sounds. Ono et al. [6] [7] separated harmonic and percussive sounds by exploiting the anisotropy of harmonic and percussive sounds in a maximum a posteriori (MAP) framework. Fitzgerald's system [8] extracted percussive sounds using the anisotropy smoothness by means of a median filtering. In this manner, the harmonics are considered to be outliers in a temporal slice. Recently, Canadas et. al [9] proposed a NMF approach that automatically distinguishes between percussive and harmonic bases by integrating spectro-temporal features, such as anisotropic smoothness or time-frequency sparseness, into the factorization process. Results were promising but the approach requires offline processing since it is necessary to decompose the whole mixture signal.

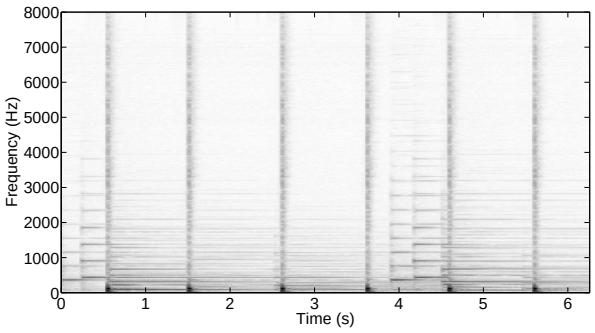


Figure 1. Magnitude spectrogram of a mixture composed of percussive and harmonic sounds. It can be seen that percussive sounds form vertical lines while the harmonic sounds form horizontal lines.

In this paper, we propose an online version of our previous work [9] where instead of decomposing the whole mixture, a set of segments of the mixture are decomposed using a sliding temporal window. Once a new segment is selected and decomposed by a constrained NMF, the sliding window is shifted by one segment. Using a small size of segment implies a faster NMF convergence. Percussive and harmonic bases are initialized randomly and updated in the decomposition of the first segment. However, the bases of the next segments are initialized using the bases obtained in the previous segment.

Consider the term latency as the time elapsed between receiving the input audio mixture and starting to perform separation in order to clarify the terms *offline*, *online* and *realtime*. The term *offline* indicates a latency equal to the duration of the whole input mixture because the whole input mixture is necessary to apply the constrained NMF [9]. The term *online* indicates a latency equal to the duration of the segment because only each segment is necessary

Copyright: ©2015 F. Canadas-Quesada, P. Vera-Candeas, N. Ruiz-Reyes 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.

to apply the constrained NMF [9] and obtain both percussive and harmonic signals related to the segment processed. However, none of the aforementioned terms provide a realtime separation. The term *realtime* indicates that the latency plus the computation time is about 30-40 milliseconds providing to the user the sense of an immediate output. The computation time is the time elapsed between starting to perform separation and obtaining each separated percussive and harmonic signals.

The remainder of the paper is organized as follows. Section 2 introduces NMF and its application to sound source separation. Section 3 describes briefly our previous offline harmonic/percussive separation work. Section 4 details the online harmonic/percussive separation proposal. Experimental results and performance analysis are shown in Section 5. Finally, conclusions are reported in Section 6.

2. NON-NEGATIVE MATRIX FACTORIZATION (NMF) FOR SOUND SOURCE SEPARATION

Non-negative Matrix Factorization (NMF) [10] is a technique for multivariate data analysis which aims to obtain a parts-based representation of objects, by imposing non-negative constraints. Given a matrix $\mathbf{X}$ of dimensions $F \times T$ with non-negative entries, it is possible to model it as linear combinations of K elementary non-negative spectra. Therefore, NMF is the problem of finding a factorization:

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{WH}, \tag{1}$$

where $\hat{\mathbf{X}}$ is the estimated matrix, $\mathbf{W} \in R^{F \times K}$ is the matrix whose columns are the bases or components. These bases represent characteristic spectral patterns active in the input spectrogram. $\mathbf{H} \in R^{K \times T}$ is a matrix of component gains for all frames. These gains represent the temporal interval in which the spectral patterns are active. In typical audio applications, the matrix $\mathbf{X}$ is chosen as a time-frequency representation (e.g., magnitude or power spectrogram), $f = 1, ..., F$ denoting the frequency bin and $t = 1, ..., T$ the time frame.

In the case of magnitude spectra, the parameters are restricted to be non-negative, then, a common way to compute the factorization in eq. (1) is generally obtained by minimizing a cost function defined as

$$D(\mathbf{X}|\hat{\mathbf{X}}) = \sum_{f=1}^{F} \sum_{t=1}^{T} d(X_{ft}|\hat{X}_{ft}), \tag{2}$$

where $d(a|b)$ is a function of two scalar variables, d is typically non-negative and takes value zero if and only if $a = b$. Using the $\beta-$divergence cost [11], some of the most popular cost functions are the Euclidean distance (β=2), the generalized Kullback-Leibler divergence (β=1) and the Itakura-Saito divergence (β=0). The cost functions are non-increasing using $1 \leq \beta \leq 2$ [12]. In practice, Févotte et.al [11] observed that the criterion is still non-increasing for β <1 and β >2 but no proof is available. An iterative algorithm based on multiplicative update rules is proposed to obtain the model parameters that minimize the cost function. In general, the update rules can be defined as follows [11],

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T((\mathbf{WH})^{\beta-2} \odot \mathbf{X})}{\mathbf{W}^T(\mathbf{WH})^{\beta-1}}, \tag{3}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{((\mathbf{WH})^{\beta-2} \odot \mathbf{X})\mathbf{H}^T}{(\mathbf{WH})^{\beta-1}\mathbf{H}^T}, \tag{4}$$

where $\mathbf{W}$ and $\mathbf{H}$ are initialized as random positive matrices, ^{T} is the transpose operator, $\odot$ represents the Hadamard (element-wise) multiplication and the division is also element-wise.

Considering that the source z is composed of a set of L components, the separated magnitude spectrogram of the source z can be reconstructed as follows,

$$X_z = \frac{\sum_{i=1}^{L} \mathbf{W_i H_i}}{\mathbf{WH}} \odot \mathbf{X}, \tag{5}$$

where the temporal signal $x_z(t)$ is computed inverting the spectrogram X_z to the time-domain using the phase of the original mixture.

3. OFFLINE HARMONIC/PERCUSSIVE SEPARATION

Unconstrained NMF cannot discriminate between percussive and harmonic bases. To overcome this problem, we proposed [9] an unsupervised system that can separate percussive and harmonic sounds in monaural music integrating percussive and harmonic sound features into the NMF decomposition. For that purpose, an objective function is defined to decompose a mixture spectrogram X into two separated spectrograms, X_P (a percussive spectrogram) and X_H (a harmonic spectrogram). Each separated spectrogram exhibits specific spectro-temporal features for percussive or harmonic sounds. The factorization model is given in eq. (6),

$$X \approx X_P + X_H = W_P H_P + W_H H_H, \tag{6}$$

where X_P, X_H, W_P, H_P, W_H and H_H are non-negative matrices.

The percussive constraints used to model percussive sounds assume smoothness in frequency (the energy slowly decreases in frequency) and sparseness in time (most of the signal energy is concentrated over short time intervals). Two constraints, spectral smoothness SSM and temporal sparseness TSP, are associated to the percussive matrix W_P.

The harmonic constraints used to model harmonic sounds assume smoothness in time (amplitudes that vary slowly in time) and sparseness in frequency (spectral peaks). Two constraints, spectral sparseness SSP and temporal smoothness TSM, are associated to the harmonic matrix W_H.

The global cost function D uses the β-divergence cost d_β, the percussive constraints (SSM, TSP) and the harmonic constraints (SSP, TSM),

$$\begin{aligned} D = d_\beta(X|(X_P + X_H)) + K_{SSM} SSM + \\ + K_{TSP} TSP + K_{TSM} TSM + K_{SSP} SSP, \end{aligned} \tag{7}$$

where the parameters $K_{SSM}, K_{TSP}, K_{TSM}, K_{SSP}$ determine the degree of control of each constraint in the NMF procedure. However, the system requires offline processing since it is necessary to decompose the whole mixture signal X. More details can be found in [9].

4. ONLINE HARMONIC/PERCUSSIVE SEPARATION

We extend our offline harmonic/percussive separation work to the case online. In the online proposal, a constrained NMF [9] is not applied to the whole mixture spectrogram X. The whole mixture signal of duration T seconds is segmented in $L = \left\lceil \frac{T}{T_i} \right\rceil$ segments S_i using a non-overlapped sliding window of duration T_i seconds as can be seen in Fig. 2.

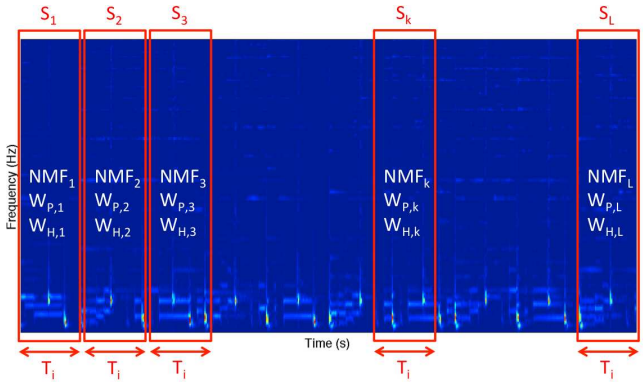


Figure 2. Online percussive/harmonic proposal. It can be seen that each segment S_i is decomposed using a constrained NMF [9] obtaining the percussive matrix $W_{P,i}$ and the harmonic matrix $W_{H,i}$ related to the magnitude spectrogram of the segment S_i.

When the magnitude spectrogram X_1 of the first segment S_1 is computed, the matrices $W_{P,1}$, $W_{H,1}$, $H_{P,1}$ and $H_{H,1}$ are initialized randomly and then are updated using [9]. In this manner, we obtain the matrices of the percussive $W_{P,1}$ and harmonic $W_{H,1}$ bases related to the first segment S_1. Next, the sliding window is shifted by one segment and a new segment is selected to apply the method [9]. Taking into account the next segment S_i, the gains matrices are randomly initialized but the bases are initialized using the bases obtained from the previous segment S_{i-1}, that is $W_{P,i} = W_{P,i-1}$, $W_{H,i} = W_{H,i-1}$ for $i = 2...L$. The idea is to use a better initialization in the next segment from the values of the bases obtained in the previous segment. The reason is because these previous bases reflect properties of percussive and harmonic sounds and it could find a better minimum local in the NMF decomposition [13]. A small number of iterations is sufficient to the NMF convergence due to the size of the spectrogram related to the segment S_i is relatively small compared to the whole spectrogram X of the mixture.

Instead of reconstructing the whole mixture as occurs in [9], each percussive $x_{p,i}(t)$ or harmonic $x_{h,i}(t)$ temporal signal related to the segment S_i is reconstructed. The separated percussive signal $x_{p,i}(t)$, composed of $L_{p,i}$ percussive components, can be synthesized inverting into the time domain via inverse Short Time Fourier Transform (STFT) using the phase of the segment S_i of the mixture. In a similar way, the harmonic signal $x_{h,i}(t)$ is synthesized taking into account the harmonic components $L_{h,i}$.

Considering a segment S_i, the time T_r defines the computation time to obtain both separated percussive $x_{p,i}(t)$ and harmonic $x_{h,i}(t)$ signals associated with the segment S_i.

5. EXPERIMENTAL RESULTS

5.1 Data set, metrics and State-of-the-art methods

A data set, composed of nine monaural real-world music excerpts, taken from the Guitar Hero game [14] [15], has been created to evaluate the performance of the proposed method as can be seen in Table 1. Each music excerpt contains percussive and harmonic instruments and a duration about T=30 seconds. All of the signals were converted from stereo to mono and sampled at 16 kHz.

Identifier	Title	Artist
$M1$	Hollywood Nights	Bob Seger & The Silver Bullet Band
$M2$	Hotel California	Eagles
$M3$	Hurts So Good	John Mellencamp
$M4$	La Bamba	Los Lobos
$M5$	Make It Wit Chu	Queens Of The Stone Age
$M6$	Ring of Fire	Johnny Cash
$M7$	Rooftops	Lost prophets
$M8$	Sultans of Swing	Dire Straits
$M9$	Under Pressure	Queen

Table 1. Identifier, Title and Artist of the files of the database

The assessment of the performance of the online proposal has been performed using the metrics Signal to Distortion Ratio (SDR), Signal to Interference Ratio (SIR) and Signal to Artifacts Ratio (SAR) [16] [17] widely used in the field of sound source separation. Higher values of these ratios indicate better separation quality.

The separation performance of the online proposal is evaluated using different durations T_i (seconds) of the segment: Online-1 ($T_i = 1$), Online-2 ($T_i = 2$), Online-3 ($T_i = 3$), Online-5 ($T_i = 5$), Online-10 ($T_i = 10$) and Online-15 ($T_i = 15$). Moreover, these online proposals are compared with the offline method [9] (the offline method assumes that the whole mixture has a duration of T seconds) and the two recent state-of-the-art percussive and harmonic sound separation methods. The first one is the method HPSS [7] and the second one is the method MFS [8].

5.2 Parameters

The normalization process [9] is applied taking into account not the size of the whole mixture but the size of the segment. Most of the parameters used in [9] are also used in the online proposal. Specifically, a frame size N =1024 samples, a time shift J =512 samples and the optimum values β = 1.5, K_{TSP} = K_{SSP} = 0.1 and K_{TSM} = K_{SSM} = 0.2. The convergence of the online proposal is

empirically observed using a number of iterations $MaxIter$ = 50. Compared to the offline version, the online proposal needs a lower number of iterations to converge due to the lower size of the spectrogram of the segment to decompose.

Considering the optimum number of percussive $R_{p_{offline}}$ and harmonic $R_{h_{offline}}$ components used in [9], the number of percussive $R_{p_{online}}$ and harmonic $R_{h_{online}}$ components used in the online proposal is computed as,

$$R_{p_{online}} = \left\lfloor \frac{T_i \cdot R_{p_{offline}}}{T} \right\rfloor, \quad (8)$$

$$R_{h_{online}} = \left\lfloor \frac{T_i \cdot R_{h_{offline}}}{T} \right\rfloor, \quad (9)$$

It seems that a lower number of percussive and harmonic components will be necessary to decompose a segment of shorter duration since a lower number of sources will be active. More details can be found in [9].

5.3 Results

Fig. 3 shows SDR (top figure) and SIR (bottom figure) percussive results evaluating the database for the online proposals and the offline method in function of the duration of the segment. Percussive SIR results improve using a very long duration ($T_i \geq 5$) of segment. Initially the percussive SDR and SIR improves using short segments but this improvement cannot be compared to the other online proposals with a long segment (see percussive SIR results). The initial SDR improvement of the Online-1 reports that using bases that contains typical features of percussive and harmonic sounds achieves to find a better minimum local in the NMF decomposition. This minimum local obtains a set of bases that reflect higher musical sense similar as percussive and harmonic sounds are perceived in the nature. The improvement in SIR between the proposal Online-1 and Online-2 is about 4dB in average so, SIR results improve significantly using a segment of duration $T_i > 1$. A drawback of the proposal Online-1 is that it captures a high amount of harmonic sounds in the separated percussive signal but these harmonic sounds are attenuated if a longer segment is used. The reason is because a longer segment provides more useful information to model correctly the sounds active in the mixture. The improvement in SIR between the proposal Online-2 and the others is approximately 2.5dB in average and approximately 1dB between the proposal Online-3 and the other online methods. Finally, the percussive performance using a segment of duration $T_i > 3$ is similar.

Fig. 4 shows SDR (top figure) and SIR (bottom figure) harmonic results evaluating the database for the online proposals and the offline method in function of the duration of the segment. Unlike in the percussive separation (as can be seen in Fig. 3), the online proposals show a slight negative slope in the harmonic SDR and SIR results. This behavior could indicate that some harmonic bases that correctly model the harmonic sounds of the previous segment are replaced by the new bases obtained in the update process of the next segment providing a fluctuation of the harmonic

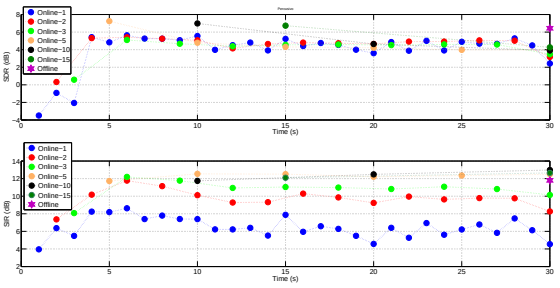


Figure 3. SDR (top) and SIR (bottom) percussive separation performance related to the offline method and online proposals.

SDR and SIR along the segments. This effect of replacement seems to be more critical in the separation of harmonic sounds compared to percussive sounds (see Fig. 3 and Fig. 4).

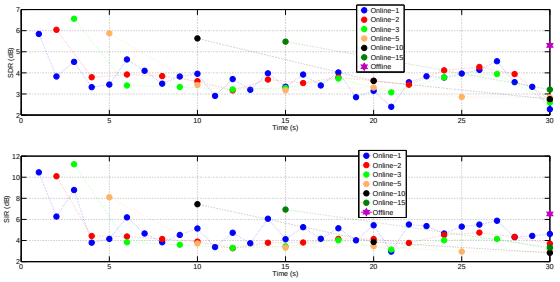


Figure 4. SDR (top) and SIR (bottom) harmonic separation performance related to the offline method and online proposals.

Fig. 5 shows SDR and SIR separation performance related to the offline method, online proposals and the two state-of-the-art percussive and harmonic sound separation methods. Each percussive bar is computed using the mean of all of the separated percussive signals of the database evaluated. In a similar way, each harmonic bar is computed taking into account the separated harmonic signals. Each group of bars in the left figure refers to percussive results and each group of bars in the right figure refers to harmonic results. Comparing the offline method and the online proposals, the best percussive and harmonic SDR and SIR results are obtained by the offline method. As can be seen, a longer duration of the segment shows a better separation providing higher quality of the separated signals. It means that a constrained NMF improves the separation performance using sufficient information of the mixture in order to model correctly the sounds active. Although the percussive SDR and the harmonic SDR and SIR are similar using the method Online-1 and Online-2, the method Online-2 achieves a significant percussive SIR improvement of about 4dB compared to the method Online-1. This improvement reports that a segment of duration equal to one second cannot model correctly harmonic sounds. For this reason, a high amount of harmonic sounds are captured in the separated percussive signal by the method Online-1

minimizing the percussive SIR as shown in Fig. 5 (bottom).

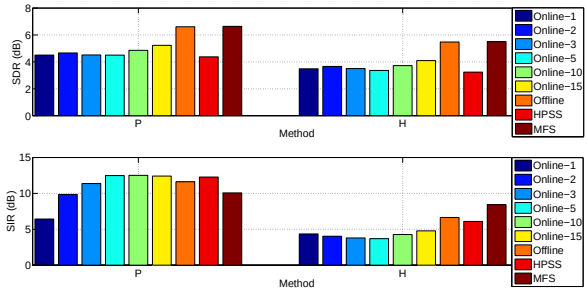


Figure 5. SDR (top) and SIR (bottom) percussive and harmonic separation performance related to the online, offline and state-of-the-art percussive and harmonic sound separation methods. The letter P in the *x-axis* refers to percussive results while the letter H in the *x-axis* refers to harmonic results.

The computation time of the online proposals is shown in Table 2 which has been computed using Matlab on a PC with Intel Core i5 CPU of 2.5 GHz and 4 GB of RAM. It can be observed that the computation time of the online proposals increases with the size of the segment. Moreover, the processing factor P_F defined as the ratio between the computation time and the duration of the segment also increases with the size of the segment. Specifically, it ranges approximately from a $1/5$ to $2/3$ of the duration of the segment processed. Although the method Online-15 obtains the best SDR and SIR results comparing the online proposals, the method Online-3 can be considered the best choice because it provides the best trade-off between sound quality and computation time.

Online proposal	Computation time T_r (sec)	$P_F = \frac{T_r}{T_i}$
Online-1 (1sec)	0.25	0.25
Online-2 (2sec)	0.43	0.22
Online-3 (3sec)	0.64	0.22
Online-5 (5sec)	1.12	0.22
Online-10 (10sec)	3.02	0.30
Online-15 (15sec)	5.85	0.39
Offline (30sec)	19.82	0.66

Table 2. The computation time of each online proposal

6. CONCLUSIONS

We extend our offline harmonic/percussive separation work [9] to the case online to separate harmonic and percussive sounds in monaural music. Instead of decomposing the whole mixture, a set of segments of the mixture are decomposed using a sliding temporal window. Once a new segment is selected and decomposed by a constrained NMF, the sliding window is shifted by one segment. Using a small size of segment implies a faster NMF convergence. Percussive and harmonic bases are initialized using the bases obtained in the NMF decomposition of the previous segment. The idea is to use a better initialization in the next segment with bases that reflect properties of percussive and harmonic sounds.

Percussive and harmonic SDR and SIR results show that a longer duration of the segment shows a better separation providing higher quality of the separated signals. It means that a constrained NMF improves the separation performance using sufficient information of the mixture in order to model correctly the sounds active.

The initialization of percussive and harmonic bases using spectral patterns that model the energy distribution of these types of sounds seems to find a better minimum local in the NMF decomposition. This better minimum local means that it provides bases that reflect higher musical sense similar as percussive and harmonic sounds are perceived in the nature. A drawback of the online proposals is that some bases that correctly model the percussive or harmonic sounds of the previous segment are replaced by the new bases obtained in the update process of the next segment providing a fluctuation of the separation performance. This replacement of bases with "good properties" is more critical in the separation of harmonic sounds compared to percussive sounds.

In the future, we plan to work on two extensions to improve the sound quality of the online proposal. The first extension is based on measuring the similarity between consecutive segments. In this manner, the percussive and harmonic bases of a segment will be initialized with random values if a low similarity is obtained regarding to the previous segment. However, the percussive and harmonic bases of a segment will be initialized with the harmonic and percussive bases computed in the previous segment if a high similarity is obtained regarding to the previous segment. The second extension is based on a new update of percussive and harmonic bases. The idea is to keep fixed along the segments those bases that have correctly model percussive or harmonic sounds in previous segments.

Acknowledgments

This work was supported by the Andalusian Business, Science and Innovation Council under project P2010- TIC-6762 (FEDER) and the Spanish Ministry of Economy and Competitiveness under Projects TEC2012-38142-C04-01, TEC2012-38142-C04-03 and TEC2012-38142-C04-04.

7. REFERENCES

[1] L. Daudet, "Review on techniques for the extraction of transients in musical signals," in *Proceedings of the Third international conference on Computer Music Modeling and Retrieval*, 2005, pp. 219–232.

[2] M. Helen and T. Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorisation and support vector machine," in *Proceedings of European Signal Processing Conference*, Anatalya, Turkey, 2005, pp. –.

[3] O. Gillet and G. Richard, "Transcription and separation of drum signals from polyphonic music," in *IEEE*

Transactions on Audio, Speech, and Language Processing, vol 3, no. 16, 2008, pp. 529–540.

[4] A. Ozerov, E. Vincent, and F. Bimbot, "A general flexible framework for the handling of prior information in audio source separation," in *IEEE Transactions on Audio, Speech, and Language Processing, vol 20, no. 4*, 2012, pp. 1118–1133.

[5] J. Yoo, M. Kim, K. Kang, and S. Choi, "Nonnegative matrix partial co-factorization for drum source separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2010, pp. –.

[6] N. Ono, K. Miyamoto, H. Kameoka, and S. Sagayama, "A real-time equalizer of harmonic and percussive components in music signals," in *Proceedings of the Ninth International Conference on Music Information Retrieval (ISMIR)*, 2008, pp. 139–144.

[7] N. Ono, K. Miyamoto, J. L. Roux, H. Kameoka, and S. Sagayama, "Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram," in *Proceedings of the European Signal Processing Conference*, 2008, pp. 25–29.

[8] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *Proceedings of Digital Audio Effects (DAFX)*, 2010, pp. –.

[9] F. Canadas, P. Vera, N. Ruiz, J. Carabias, and P. Cabanas, "Percussive/harmonic sound separation by non-negative matrix factorization with smoothness/sparseness constraints," in *Journal on Audio, Speech, and Music Processing, vol. 2014, no. 26*, 2014, pp. 1–17.

[10] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," in *Proc. of Advances in Neural Inf. Process. System*, 2000, pp. 556–562.

[11] C. Févotte, N. Bertin, and J. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence. with application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[12] R. Kompass, "A generalized divergence measure fon nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 3, pp. 780–791, 2007.

[13] B. Zhu, W. Li, R. Li, and X. Xue, "Multi-stage non-negative matrix factorization for monaural singing voice separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2096–2107, 2013.

[14] Activision, "Guitar hero world tour," in *http://en.wikipedia.org/wiki/Guitar_Hero_World_Tour*.

[15] ——, "Guitar hero 5," in *http://en.wikipedia.org/wiki/Guitar_hero_5*.

[16] E. Vincent, C. Févotte, and R. Gribonval, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[17] C. Févotte, R. Gribonval, and E. Vincent, "Bss_eval toolbox user guide - revision 2.0, technical report 1706," in *IRISA, April*, 2005, pp. –.

Wave Voxel Synthesis

Anis Haron
Media Arts & Technology, UCSB
anisharon@umail.ucsb.edu

Matthew Wright
CREATE / Media Arts & Technology, UCSB
matt@create.ucsb.edu

ABSTRACT

We present research in sound synthesis techniques employing lookup tables higher than two dimensions. Higher dimensional wavetables have not yet been explored to their fullest potential due to historical resource restrictions, particularly memory. This paper presents a technique for sound synthesis by means of three-variable functions as an extension to existing multidimensional table lookup synthesis techniques.

1. INTRODUCTION

Table lookup techniques are computationally efficient methods for oscillators employed in many sound synthesis applications, including wavetable synthesis, vector synthesis, wave stacking, wave terrain synthesis, scanned synthesis, and many others [1–3]. These techniques employ table indexing operations in one- and two-dimensional spaces.

Our proposed method extends current table-lookup sound synthesis techniques, particularly wave terrain synthesis, with three-variable functions. We introduce the term ***wave voxel*** to denote three-dimensional lookup tables for sound synthesis.

2. AN OVERVIEW OF 1D AND 2D WAVETABLES

A wavetable of D dimensions consists of precomputed amplitude values stored in a D-dimensional array. An indexing operation takes D indices and retrieves the desired value stored at that particular location of the array.

2.1 1D wavetable

A 1D wavetable is a length N lookup table with sampled amplitude values for one cycle of an arbitrary wave. It is visually represented in two axes, where x is a time axis representing index number from 0 to $N-1$ and the y axis denotes amplitude. Algorithm 1 writes (one period of) a single sine wave into a wavetable, as in Figure 1.

Algorithm 1

for $i = 0\,, i < N\,, i++$ **do**
$table[i] = sin(2\pi(i/N))$

Copyright: ©2015 Anis Haron 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.

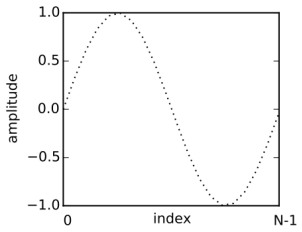


Figure 1. Wavetable of size N.

Indexing a 1D wavetable selects a point on the x axis. As a wavetable contains one cycle of a wave, indexing in this instance is essentially a continuous modular arithmetic phase increment looping through the wavetable.

Algorithm 2

```
phase = 0
increment = (frequency / samplerate) × tablesize
  while true do
      phase = phase + increment
      while (phase >= tablesize) do
          phase = phase - tablesize
      output = table.interpolate(phase)
```

Algorithm 2 generates incremental phase with the appropriate increment values based on desired frequency, table size N, and sampling rate [4, 5]. Frequency should be between 0 and the nyquist frequency. In general the resulting phase values are floating point numbers, hence the need for some sort of interpolated read from the table (even if just truncating phase to an integer).

Dannenberg demonstrated that the table size required to achieve a given signal-to-noise ratio (SNR) is highly dependent on interpolation method employed [5]. The noise comes from quantization errors both in time and amplitude. A large enough table size would yield a high SNR even without interpolation. Interpolation improves SNR, thus allowing for smaller table sizes, although computation cost will increase. Truncation could produce results that err at most by almost an entire sample, while rounding could produce results that err at most by half a sample. Linear interpolation approximates a waveform curve better, thus producing a lower error, while higher order interpolation methods such as cubic and Catmull-Rom produce even lower error [4].

2.2 2D Wavetable (Wave Terrain)

An extension to the wavetable was formally introduced by Mitsuhashi in [6]. This technique is also known as "wave terrain synthesis," named after Gold's use of the term [7, 8]. Wave terrains are generally illustrated as a three-dimensional surface where the Z axis (height) represents amplitude as a function of 2D location X and Y [1], as in Figure 3. Terrains can also be visualized in a two dimensional plot, with color at a pixel location representing the amplitude value at that location, as in Figure 2. Both axes (X and Y) are used for indexing operations in wave terrain synthesis. The path of an indexing operation in a wave terrain is called an *orbit* [9].

This technique of using multidimensional surfaces, introduced by Mitsuhashi, was originally intended for hardware implementation as an alternative for Chowning's method of FM synthesis [6, 10, 11]. However, it was in 1978 that Gold, a member of the League of Automatic Music Composers, first used the term *wave terrain* to describe a two dimensional map implemented in his instrument, *A Terrain Reader* [7, 8]. In 1986, Borgonovo and Haus implemented a software version of Mitsuhashi's two-variable technique [10].

Mitsuhashi proposes restrictions for the functions that can be used for terrain generation. He recommended that terrains be continuous in the area of definition and on its boundaries [6, 9]. While Mitsuhashi's, Borgonovo's, and Gold's implementation focused on trigonometric polynomials for terrain generation, latter researches explored other means. Di Scipio experimented with functional iterations [12], Mikelson uses the Julia set as terrains [13], Overholt fabricated a hardware interface for generation of user defined terrains [14], while Dannenberg and Neuendorffer used real-time video images [15].

To illustrate an example of a wave terrain, we referred to an implementation by Comajuncosas in CSound. The pseudocode shown in algorithm 3 generates an N_x by N_y wave terrain containing one cycle of sine wave in each direction, as implemented in [16] and illustrated in Figure 2.

Algorithm 3

for $y = 0$, $y < N_y$, $y++$ **do**
  $siny = sin(2\pi(y/N_y))$
  for $x = 0$, $x < N_x$, $x++$ **do**
    $sinx = sin(2\pi(x/N_x))$
    $table[x][y] = sinx \times siny$

Amplitude values from -1 to 1 at each location are mapped to color from black to white. Figure 2 visualizes a wave terrain in two dimensions, using color values to represent amplitude values. The same wave terrain, visualized in three dimensions is illustrated in figure 3.

Just as with terrain definitions, there are many different implementations of orbit trajectories. As an example, equations 1 and 2 shows orbit trajectories as implemented by Mitsuhashi [6] and Borgonovo [9].

$$x = 2f_x t + \phi_x + Asin(2\pi F_x t + \varphi_x) \quad (1)$$

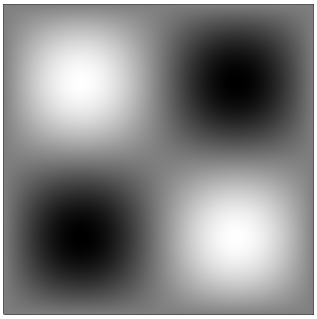


Figure 2. Wave terrain of size N_x by N_y, using pixel brightness to represent amplitude.

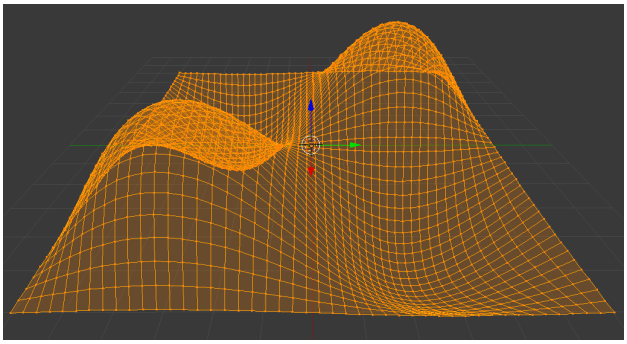


Figure 3. Wave terrain in figure 2 visualized in 3D.

$$y = 2f_y t + \phi_y + Bsin(2\pi F_y t + \varphi_y) \quad (2)$$

In the equations above, t denotes time, f_x, f_y, F_x and F_y are frequencies, A and B are amplitudes, and $\phi_x, \phi_y, \varphi_x$, and φ_y are initial phases. These equations can be broken down into two terms: the first, $2ft + \phi$, describes linearity (starting at a given point on the plane and moving in a straight line over time), while the second, $Asin(2\pi Ft+\varphi)$, describes the nonlinear portion of the equation (Lissajous curves). 2D table indexing, as with 1D, should be wrapped (modulo the table size) so that orbits can go past the boundaries of the terrain. Time-varying values for any one or combinations of variables generate time-varying waveforms.

Wave terrains are by no means restricted to only direct synthesis for sound generation. Other applications have seen the use of wave terrains as a two dimensional surface for dynamic nonlinear distortion or waveshaping, as a control method for multichannel panning, and even as a basis for generative scored compositions [17].

3. 3D WAVETABLE (WAVE VOXEL)

Borgonovo mentioned a higher dimensional approach to wave terrain synthesis [9] but did not implement it at the time due to computational expense. A higher dimensional implementation of wave terrains was implemented by Mikelson in [18]. Mikelson named his implementation *Terrain Mapping Synthesis*, which was written in CSound. His instrument, *Deep Space Growl*, uses a terrain based on a four dimensional polygonal modeling of a torus with spiral orbits in three dimensions.

In this section, we present our approach towards a technique for sound synthesis using three dimensional waveta-

bles. We propose the term ***wave voxel*** [1] to denote a three dimensional wavetable, and name this technique ***wave voxel synthesis***. Our current version is a real-time implementation written in C++ using the openFrameworks [2] open source toolkit.

Voxel models are widely used in computer graphics for volumetric imaging such as in visualizing scientific data (simulation of smoke), medical applications (MRI, ultrasound), and in video games. A single voxel represents a value on a regular grid in three-dimensional space. A regular grid of N_x by N_y by N_z voxels is called a *voxel stack*, and a voxel stack contains $N_x \times N_y \times N_z$ individual voxels, each with its own value represented by a color or in some instances by opacity [19]. Wave voxels are extensions of wave terrains just as wave terrains are extensions of wavetables. Figure 4 illustrates a $12 \times 12 \times 12$ voxel stack; note that for sound synthesis applications, such small tables generate signals with low SNR.

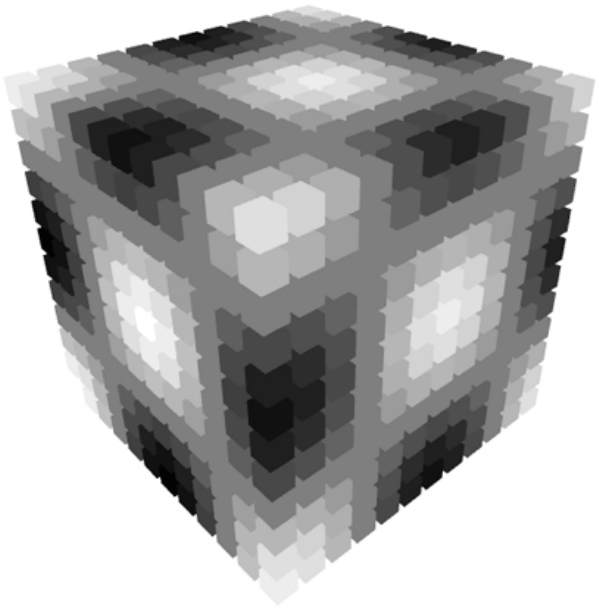


Figure 4. Stack of wave voxels of size N_x by N_y by N_z. [$N_x = N_y = N_z = 12$]

3.1 Voxel stack

For illustrative purposes, we used cosine waves at each axis in the voxel stack shown in figure 4. [3]

A stack of wave voxel of size N_x by N_y by N_z, storing amplitude values for a single cycle of a cosine wave as illustrated in figure 4 was generated using the pseudocode shown in algorithm 4.

A three dimensional wavetable can be viewed as layers of wave terrains, stacked on top of one another. Figure 5 illustrates three slices along the Z axis at locations 0, N_z/2 and N_z-1.

[1] "Voxel" is a portmanteau of "volume" and "pixel."

[2] `http://www.openframeworks.cc/`

[3] As sine waves begins at amplitude 0.0, each voxel at all six faces of the stack will store the same amplitude value of 0.0. Value of 0.0 would be graphically represented by the same gray hue, not very interesting to visualize.

Algorithm 4

for $z = 0$, $z < N_z$, $z++$ **do**
$cosz = cos(2\pi(z/N_z))$
  for $y = 0$, $y < N_y$, $y++$ **do**
  $cosy = cos(2\pi(y/N_y))$
    for $x = 0$, $x < N_x$, $x++$ **do**
    $cosx = cos(2\pi(x/N_x))$
    $table[x][y][z] = cosx \times cosy \times cosz$

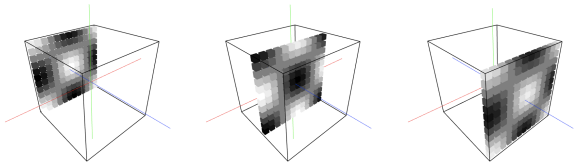


Figure 5. Slices along the Z axis of a $12 \times 12 \times 12$ voxel stack. [Left to right] $table[x][y][0]$, $table[x][y][N_z/2]$ and $table[x][y][N_z - 1]$

3.2 Indexing 3D wavetables

In our current implementation, our approach for three dimensional indexing is shown in equations 3, 4 and 5, which is largely based on equations 1 and 2. We've implemented orbits of finite length. Orbit length (I) is the length of the longest axis in the voxel stack (N_x, N_y or N_z).

$$x = 2f_x(i/I) + \phi_x + \mu_x A sin(2\pi F_x(i/I) + \varphi_x) \quad (3)$$

$$y = 2f_y(i/I) + \phi_y + \mu_y B sin(2\pi F_y(i/I) + \varphi_y) \quad (4)$$

$$z = 2f_z(i/I) + \phi_z + \mu_z C sin(2\pi F_z(i/I) + \varphi_z) \quad (5)$$

As in equations 1 and 2, the variables f_x, f_y, f_z, F_x, F_y and F_z denote frequencies, A, B, and C denote amplitude, and ϕ_x, ϕ_y, ϕ_z, φ_x, φ_y, and φ_z are initial phases. I represents length of orbit, while i is the index or position along an orbit where $0 <= i < I$.

The variables μ_x, μ_y and μ_z are amplitude envelopes. In our current version, it adds the option to have a constant (equation 6), incremental (equation 7) or triangularly windowed amplitude (equation 8).

$$const(i) = 1.0 \quad (6)$$

$$incr(i) = i/I \quad (7)$$

$$tri(i) = 1.0 - \mid 2i - (I-1) \mid \quad (8)$$

With equations 3, 4, and 5, an orbit now inhabits three dimensional space. We've implemented three dimensional transformations to maximize control and the capability to generate a variety of waveforms using the same orbit definition. Orbit transformations would also be possible for two dimensional orbits in wave terrain synthesis, although it would be rather restricted compared to transformations in three dimensions.

3.3 Orbit transformations

We apply the standard transformation matrices commonly used in computer graphics [19] to modulate orbits such as those defined above. This changes the spectral characteristics of a signal using a relatively small number of control parameters (particularly with rotation and shearing transformations).

3.3.1 Scale

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{9}$$

Equation 9 shows the scaling matrix implemented where s_x, s_y and s_z are the scaling factor on x, y and z axis respectively.

3.3.2 Translate

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a_x \\ 0 & 1 & 0 & a_y \\ 0 & 0 & 1 & a_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{10}$$

Equation 10 shows translation matrix used to enable orbit translation along x, y and/or z axis where a_x, a_y and a_z are translation amount (in voxels) on x, y and z axis respectively. a_x, a_y and a_z are values denoting offset along a particular axis.

3.3.3 Rotate

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix}$$

$$R_y = \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix}$$

$$R_z = \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' & y' & z' \end{bmatrix} = R_x R_y R_z \begin{bmatrix} x & y & z \end{bmatrix} \tag{11}$$

For rotations along x, y, and z axes, we've implemented rotation matrices as shown in equation 11 where α, β and γ are rotation angles when rotating on x, y or z axis respectively. Sign of the value for rotation angle determines direction of rotation.

3.3.4 Shear

$$H_x = \begin{bmatrix} 1 & 0 & 0 \\ tan(a) & 1 & 0 \\ tan(b) & 0 & 1 \end{bmatrix}$$

$$H_y = \begin{bmatrix} 1 & tan(c) & 0 \\ 0 & 1 & 0 \\ 0 & tan(d) & 1 \end{bmatrix}$$

$$H_z = \begin{bmatrix} 1 & 0 & tan(e) \\ 0 & 1 & tan(f) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' & y' & z' \end{bmatrix} = H_x H_y H_z \begin{bmatrix} x & y & z \end{bmatrix} \tag{12}$$

In a two dimensional plane, a set of vectors could be sheared along the x axis, in which case its y values remains the same after applying transformation. Conversely, shearing is also possible along the y axis in which case its x values are retained after a transformation. Viewing from the z axis in a three dimensional space, we are essentially looking at a two dimensional plane where x axis is the abscissa and y axis the ordinate. Looking at it this way, shearing in a three dimensional space when viewed from the z axis is similar to shearing in two dimensions. Similarly, when viewed along y axis, shearing is possible along x and z axes while viewing from the x axis enables shearing along y and z axes.

Equation 12 shows shearing matrices used, where a and b are shearing degrees on y and z axis respectively when viewed from x axis, c and d are shearing degrees on x and z axis respectively when viewed from y axis and e and f are shearing degrees on x and y axis respectively when viewed from z axis.

As before, the variables a, b, c, d, e and f are angles in degrees and sign of the value for shearing angle determines shearing direction.

3.4 Orbit's origin and modulus

An orbit's trajectory starts at the origin, which is defined to be the center point of a voxel stack at coordinate (0, 0, 0). Modulus for orbits are set to be the size of voxel stack's length, width and height for orbits to wrap around if it exceeds the size of a voxel stack.

As an example, in a voxel stack with 512 voxels on its length, height and width (cube), a trajectory of length 512 traversing in a straight line parallel to the x axis will start at the origin, continues on until it reaches the boundary of the voxel stack at orbit length 256, wraps around to the other end of the axis, continues towards the origin and stops one voxel shy from the origin at orbit length 512. Applying translation on x axis for 256 or -256 voxels would move the orbit so that the trajectory starts at one boundary and ends at the opposite boundary. Figure 6 illustrates this example.

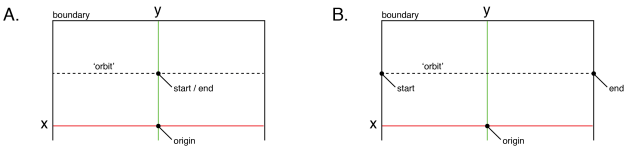


Figure 6. [A] Orbit's start and end point by default. [B] Orbit's start and end point, translated by $-N_x/2$ on x axis.

3.5 Examples of orbits and its resulting waveform

In this section, we present a selection of waveforms generated by specific orbit trajectories. Voxel stack used to generate the waveforms presented in this section are as illustrated in figure 4, but with sine waves instead. Size of voxel stack and orbit length are 512.

For each example presented in this section, both time and frequency domain representations of the waveforms are illustrated, accompanied with orbit trajectory plots (Figure

8) and a table listing values used for each variables (Table 1).

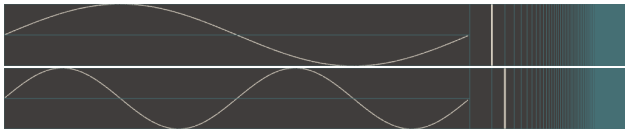


Figure 7. Time and frequency domain representations of a sine wave. [Top] 1st harmonic, [Bottom] 2nd harmonic.

Figure 7 illustrates both time and frequency domain plots as it appears in our current software implementation. The larger portion on the left illustrates time domain representation of the waveform, while the smaller portion on the right illustrates its harmonic contents.

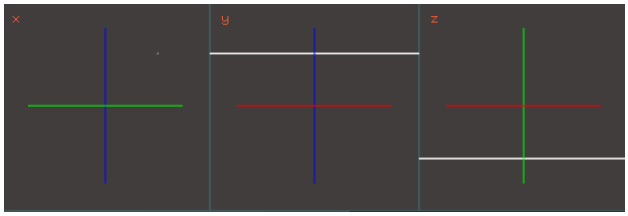


Figure 8. Orbit viewed from x, y and z viewports.

Table 1. Values for orbit shown in figure 8.

linear frequency (f_x, f_y, f_z)	**linear phase** (ϕ_x, ϕ_y, ϕ_z)	**amplitude** (A, B, C)	**amp. envelope** (μ_x, μ_y, μ_z)	**frequency** (F_x, F_y, F_z)	**phase** $(\varphi_x, \varphi_y, \varphi_z)$
1, 0, 0	0, 0.5, -0.5	0, 0, 0	a, a, a	0, 0, 0	0, 0, 0
scale (s_x, s_y, s_z)	**translate** (a_x, a_y, a_z)	**rotate** (α, β, γ)	**shear x** (a, b)	**shear y** (c, d)	**shear z** (e, f)
1, 1, 1	0, 0, 0	0, 0, 0	0, 0	0, 0	0, 0

Figure 8 illustrates trajectory of orbit viewed from x, y and z axis viewports. The color coded crosshairs are axis reference, where colors red, green and blue represents x, y and z axis respectively.

Waveform shown in figure 7 [Top] were generated by trajectory shown in figure 8, using values listed in table 1. For the waveform shown in figure 7 [Bottom], x axis linear frequency (f_x) is set to 2.

Linear phases (ϕ_x, ϕ_y and ϕ_z) and phases (φ_x, φ_y and φ_z) are normalized to 1.0. Both rotation angles (α, β and γ), and shearing angles (a, b, c, d, e and f) are in degrees, while translation values (a_x, a_y and a_z) are by number of voxels.

Amplitude envelope variables (μ_x, μ_y and μ_z) are listed as characters 'a', 'b' or 'c' representing either a constant value (equation 6), a linear ramp (equation 7) or a triangular window (equation 8) respectively.

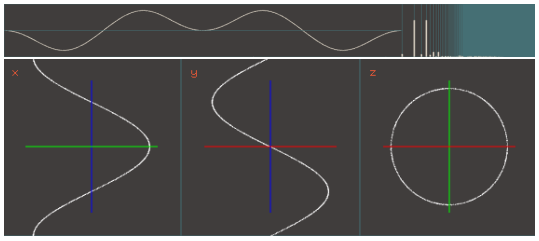


Figure 9. Orbit/Waveform example01

Table 2. Values for orbit/waveform shown in figure 9.

linear frequency (f_x, f_y, f_z)	**linear phase** (ϕ_x, ϕ_y, ϕ_z)	**amplitude** (A, B, C)	**amp. envelope** (μ_x, μ_y, μ_z)	**frequency** (F_x, F_y, F_z)	**phase** $(\varphi_x, \varphi_y, \varphi_z)$
0, 0, 1	0, 0, 0	0.65, 0.65, 0	a, a, a	1, 1, 0	0, 0.25, 0
scale (s_x, s_y, s_z)	**translate** (a_x, a_y, a_z)	**rotate** (α, β, γ)	**shear x** (a, b)	**shear y** (c, d)	**shear z** (e, f)
1, 1, 1	0, 0, 0	0, 0, 0	0, 0	0, 0	0, 0

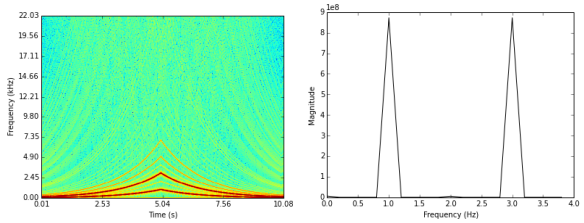


Figure 10. Waveform example01 (Figure 9) [Left] frequency sweep (50hz to 1khz), [Right] harmonic contents at 1hz.

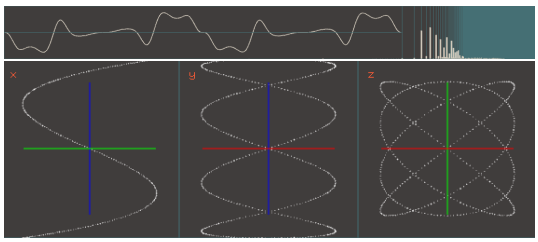


Figure 11. Orbit/Waveform example02

Table 3. Values for orbit/waveform shown in figure 11.

linear frequency (f_x, f_y, f_z)	**linear phase** (ϕ_x, ϕ_y, ϕ_z)	**amplitude** (A, B, C)	**amp. envelope** (μ_x, μ_y, μ_z)	**frequency** (F_x, F_y, F_z)	**phase** $(\varphi_x, \varphi_y, \varphi_z)$
0, 0, 0	0, 0, 0	0.75, 0.75, 1	a, a, a	4, 3, 1	0, 0, 0
scale (s_x, s_y, s_z)	**translate** (a_x, a_y, a_z)	**rotate** (α, β, γ)	**shear x** (a, b)	**shear y** (c, d)	**shear z** (e, f)
1, 1, 1	0, 0, 0	0, 0, 0	0, 0	0, 0	0, 0

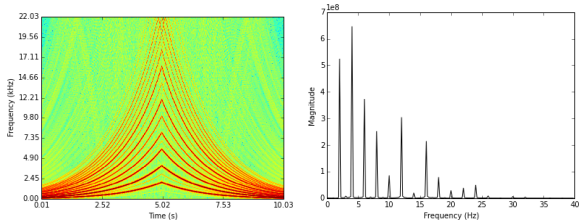


Figure 12. Waveform example02 (Figure 11) [Left] frequency sweep (50hz to 1khz), [Right] harmonic contents at 1hz.

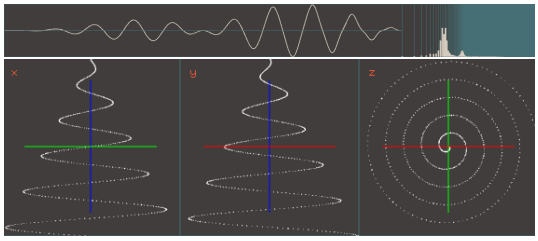


Figure 13. Orbit/Waveform example03

Table 4. Values for orbit/waveform shown in figure 13.

linear frequency (f_x, f_y, f_z)	**linear phase** (ϕ_x, ϕ_y, ϕ_z)	**amplitude** (A, B, C)	**amp. envelope** (μ_x, μ_y, μ_z)	**frequency** (F_x, F_y, F_z)	**phase** $(\varphi_x, \varphi_y, \varphi_z)$
0, 0, 1	0, 0, 0	1, 1, 0	b, b, a	5, 5, 0	0.25, 0, 0
scale (s_x, s_y, s_z)	**translate** (a_x, a_y, a_z)	**rotate** (α, β, γ)	**shear x** (a, b)	**shear y** (c, d)	**shear z** (e, f)
1, 1, 1	0, 0, 256	0, 0, 0	0, 0	0, 0	0, 0

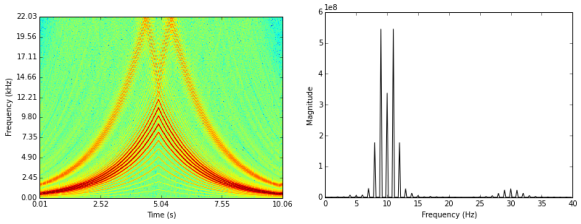


Figure 14. Waveform example03 (Figure 13) [Left] frequency sweep (50hz to 1khz), [Right] harmonic contents at 1hz.

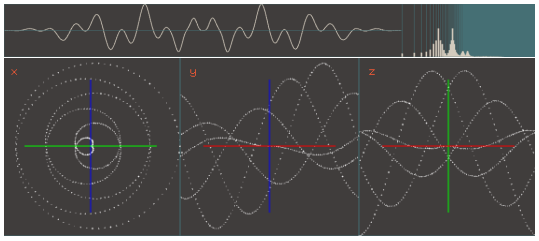


Figure 15. Orbit/Waveform example04

Table 5. Values for orbit/waveform shown in figure 15.

linear frequency (f_x, f_y, f_z)	**linear phase** (ϕ_x, ϕ_y, ϕ_z)	**amplitude** (A, B, C)	**amp. envelope** (μ_x, μ_y, μ_z)	**frequency** (F_x, F_y, F_z)	**phase** $(\varphi_x, \varphi_y, \varphi_z)$
5, 0, 0	0, 0, 0	0, 1, 1	a, c, c	0, 6, 6	0, 0.25, 0
scale (s_x, s_y, s_z)	**translate** (a_x, a_y, a_z)	**rotate** (α, β, γ)	**shear x** (a, b)	**shear y** (c, d)	**shear z** (e, f)
1, 1, 1	0, 0, 0	0, 0, 0	0, 0	0, 0	0, 0

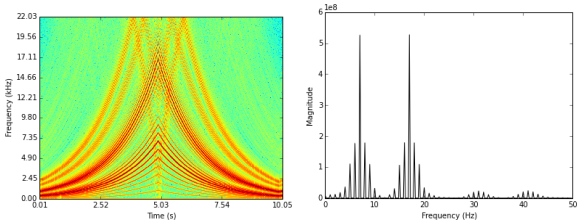


Figure 16. Waveform example04 (Figure 15) [Left] frequency sweep (50hz to 1khz), [Right] harmonic contents at 1hz.

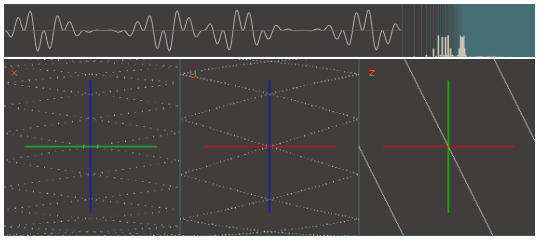


Figure 17. Orbit/Waveform example05

Table 6. Values for orbit/waveform shown in figure 17.

linear frequency (f_x, f_y, f_z)	**linear phase** (ϕ_x, ϕ_y, ϕ_z)	**amplitude** (A, B, C)	**amp. envelope** (μ_x, μ_y, μ_z)	**frequency** (F_x, F_y, F_z)	**phase** $(\varphi_x, \varphi_y, \varphi_z)$
10, 10, 0	0, 0, 0	0, 0, 1	a, a, a	0, 0, 1	0, 0, 0
scale (s_x, s_y, s_z)	**translate** (a_x, a_y, a_z)	**rotate** (α, β, γ)	**shear x** (a, b)	**shear y** (c, d)	**shear z** (e, f)
1, 1, 1	0, 0, 0	0, 0, 0	45, 0	0, 0	0, 0

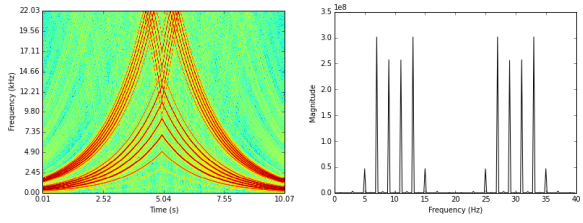


Figure 18. Waveform example05 (Figure 17) [Left] frequency sweep (50hz to 1khz), [Right] harmonic contents at 1hz.

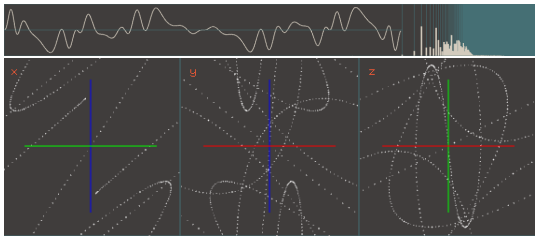


Figure 19. Orbit/Waveform example06

Table 7. Values for orbit/waveform shown in figure 19.

linear frequency (f_x, f_y, f_z)	**linear phase** (ϕ_x, ϕ_y, ϕ_z)	**amplitude** (A, B, C)	**amp. envelope** (μ_x, μ_y, μ_z)	**frequency** (F_x, F_y, F_z)	**phase** $(\varphi_x, \varphi_y, \varphi_z)$
0, 0, 0	0, 0, 0	1, 1, 1	a, a, a	2, 1, 3	0, 0, 0
scale (s_x, s_y, s_z)	**translate** (a_x, a_y, a_z)	**rotate** (α, β, γ)	**shear x** (a, b)	**shear y** (c, d)	**shear z** (e, f)
1.5, 1, 1.75	0, 0, 0	45, 0, 0	0, 0	30, 0	45, 0

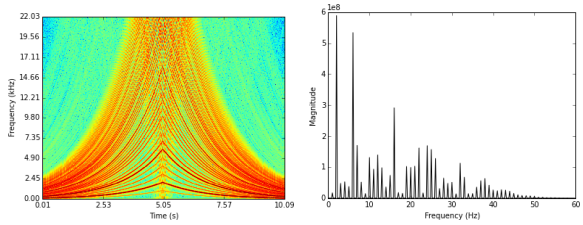


Figure 20. Waveform example06 (Figure 19) [Left] frequency sweep (50hz to 1khz), [Right] harmonic contents at 1hz.

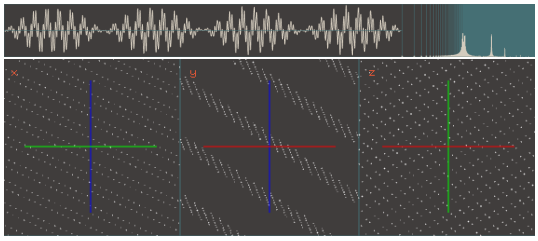


Figure 21. Orbit/Waveform example07

Table 8. Values for orbit/waveform shown in figure 21.

linear frequency (f_x, f_y, f_z)	**linear phase** (ϕ_x, ϕ_y, ϕ_z)	**amplitude** (A, B, C)	**amp. envelope** (μ_x, μ_y, μ_z)	**frequency** (F_x, F_y, F_z)	**phase** $(\varphi_x, \varphi_y, \varphi_z)$
1, 1, 1	0, 0, 0	0, 0, 0	a, a, a	0, 0, 0	0, 0, 0
scale (s_x, s_y, s_z)	**translate** (a_x, a_y, a_z)	**rotate** (α, β, γ)	**shear x** (a, b)	**shear y** (c, d)	**shear z** (e, f)
5, 2, 3	0, 0, 0	0, 15, 0	0, 33	0, 0	89, 0

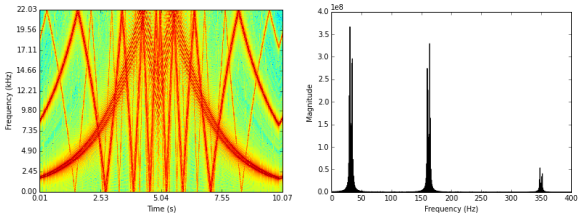


Figure 22. Waveform example07 (Figure 21) [Left] frequency sweep (50hz to 1khz), [Right] harmonic contents at 1hz.

At the moment we have only explored time-invariant orbits with waveforms generated using sine waves that is isomorphic in x, y and z axes to generate voxel stack content. Using pure tones allows us to better understand harmonic contents of a waveform generated by a particular orbit. To fully maximize the potential of this technique, a voxel model should store data that differs at each axis with dynamic orbits trajectories.

Sound files for each example presented in this section are available for streaming online. [4]

3.6 Aliasing

As visually apparent in the frequency sweep plots shown in the previous section, we've ignored aliasing issues in our current software implementation.

One solution for this issue would be to increase its sampling rate. As an alternative, another approach would be to implement multiple sub-wavetables [20]. For example, suppose the desired frequency range is from 20hz to 22.05khz. This range corresponds to roughly ten octaves from the lowest frequency (20hz to 20.48khz).

In this instance, we could construct ten sub-wavetables, one for each octave. Harmonic contents of each sub-wave table are reduced as the octave gets higher. Ultimately, the sub-wavetable for the highest octave contains only a single period of sine tone. Based on the desired output frequency, a specific sub-wavetable is chosen out of the ten created. This approach guarantees a bandlimited output.

3.7 Resizing of orbit for synthesis

In order to generate audio signals with an acceptable SNR, wavetables are generally implemented in powers of two with 2048 or 4096 samples in size. As in wave terrain synthesis, orbit length correlates to wavetable length. In our current implementation, length of orbit is the same length as voxel stack size (N=512), which is insufficient to produce signals with an acceptable SNR. To construct a larger wavetable, we resized the orbit using catmull-rom spline interpolation [19].

[4] `https://soundcloud.com/anisharon/sets/smc2015-wave-voxel-synthesis`

4. CONCLUSIONS & FUTURE WORK

We introduced the term wave voxel to denote a three dimensional lookup table for sound synthesis. We proposed a new method for sound synthesis by means of three-variable functions as an extension to existing multidimensional table lookup synthesis techniques, presented a method for visualizing three dimensional lookup tables using volumetric representations, and demonstrated a table indexing method for use with three dimensional wavetables based on previous research in multidimensional scanning by Mitsuhashi and Borgonovo/Haus.

A study on comparison with other synthesis techniques is ongoing at the time of writing. Part of our ongoing work is exploring other means in creating orbit trajectories. Besides implementing other types of affine transformations, we are interested in experimenting with time-varying orbits, dynamic modulation of orbits using transformations, orbit trajectories based on summation of spherical curves [21], three dimensional knight's tour [22] and dynamic vector fields.

We are also interested in exploring the possibility of using dynamic voxel stack contents, for instance by traversing a voxel stack through samples of recorded audio, where each axis would store different snippets of recordings. In this instance, a voxel stack can be thought of as a three dimensional rectangular granulation window. We will also look into volumetric scene reconstruction from multiple cameras [23] as a method for generating dynamic voxel stack contents. Also pertinent to our research are interfacing and control methods to effectively control our proposed technique for use as an instrument in live setups.

Acknowledgments

The authors would like to thank Curtis Roads for his support with this research. We would also like to extend our sincerest gratitude to Andrés Cabrera, Yuan-Yi Fan, members of Media Arts & Technology's Write Club and the CREATE Ensemble for providing us with valuable feedback.

5. REFERENCES

[1] C. Roads, *The Computer Music Tutorial*. Cambridge, Massachusetts. London, England: The MIT Press, 1996.

[2] M. M. Bill Verplank, "Scanned synthesis," *Interval Research Corporation*, 1999.

[3] J. F. Richard Boulanger, Paris Smaragdis, "Scanned synthesis: An introduction and demonstration of a new synthesis and signal processing technique," *International Computer Music Conference*, pp. 372–375, 2000.

[4] G. Loy, *Musimathics. The Mathematical Foundations of Music*. Cambridge, Massachusetts. London, England: The MIT Press, 2007, vol. 2.

[5] R. B. Dannenberg, "Interpolation error in waveform table lookup," in *Proceedings of the International Computer Music Conference*, 1998, pp. 240–243.

[6] Y. Mitsuhashi, "Audio signal synthesis by functions of two variables," in *Journal of the Audio Engineering Society*, vol. 30, no. 19, 1982, pp. 701–706.

[7] R. Gold, *A Terrain Reader (The BYTE Book of Computer Music)*, C. P. Morgan, Ed. Byte Publications Inc, 1979.

[8] J. H. John Bischoff, Rich Gold, "Music for an interactive network of microcomputers," in *Computer Music Journal*, vol. 2, no. 3, 1978, pp. 24–29.

[9] G. H. Aldo Borgonovo, "Sound synthesis by means of two-variable functions: Experimental criteria and results," in *Computer Music Journal*, vol. 10. The MIT Press, 1986, pp. 57–71.

[10] C. d. S. Anderson Mills, "Gestural sounds by means of wave terrain synthesis," *Congresso Nacional da Sociedade Brasileira de Computação XIX*, 1999.

[11] J. M. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *Journal of the Audio Engineering Society*, vol. 21, no. 7, pp. 526–534, September 1973.

[12] A. D. Scipio, "The synthesis of environmental sound textures by iterated nonlinear functions, and its ecological relevance to perceptual modeling," *Journal of New Music Research*, vol. 2, no. 32, pp. 109–117, 2002.

[13] H. Mikelson, "Sound generation with the julia set," *CSound Magazine*, vol. Summer, 1999.

[14] D. Overholt, "New musical mappings for the matrix interface," *International Computer Music Conference*, pp. 486–488, 2002.

[15] T. N. Roger B. Dannenberg, "Sound synthesis from real-time video images," in *International Computer Music Conference*. International Computer Music Association, 2003, pp. 385–388.

[16] J. M. Comajuncosas, *Wave Terrain Synthesis (The CSound Book, CD-Rom Chapter 22)*, R. Boulanger, Ed. The MIT Press, 2000.

[17] C. H. Stuart James, "Multidimensional data sets: Traversing sound synthesis, sound sculpture, and scored composition," *Australasian Computer Music Conference*, 2011.

[18] H. Mikelson, *Terrain Mapping Synthesis (The CSound Book, CD-Rom Chapter 26)*, R. Boulanger, Ed. The MIT Press, 2000.

[19] J. Huges, A. Dam, M. McGuire, D. Sklar, J. Foley, S. K. Feiner, and K. Akeley, *Computer Graphics: Principles and Practice*, 3rd ed. Addison-Wesley Professional, 2013.

[20] (2013, March). [Online]. Available: http://www.earlevel.com/main/2013/03/03/replicating-wavetables/

[21] L. Putnam, "A method of timbre-shape synthesis based on summetion of spherical curves," in *Proceedings ICMC\SMC\2014*. Athens, Greece: International Computer Music Association, Sound and Music Computing, 2014, pp. 1332 – 1337.

[22] S. Bai, X.-F. Yang, G.-B. Zhu, D.-L. Jiang, and J. Huang, "Generalized knight's tour on 3d chessboards," *Discrete Applied Mathematics*, vol. 158, pp. 1727–1731, 2010.

[23] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer, "A survey of methods for volumetric scene reconstruction from photographs," *The International Workshop on Volume Graphics*, 2001.

Mozart is still blue: a comparison of sensory and verbal scales to describe qualities in music

Maddalena Murari, Antonio Rodà, Osvaldo Da Pos
University of Padova
maddalena.murari@unipd.it

Emery Schubert
University of New South Wales
e.schubert@unsw.edu.au

Sergio Canazza, Giovanni De Poli
University of Padova
canazza@dei.unipd.it

ABSTRACT

An experiment was carried out in order to assess the use of non-verbal sensory scales for evaluating perceived music qualities, by comparing them with the analogous verbal scales. Participants were divided into two groups; one group (SV) completed a set of non-verbal scales responses and then a set of verbal scales responses to short musical extracts. A second group (VS) completed the experiment in the reverse order. Our hypothesis was that the ratings of the SV group can provide information unmediated (or less mediated) by verbal association in a much stronger way than the VS group. Factor analysis performed separately on the SV group, the VS group and for all participants shows a recurring patterning of the majority of sensory scales versus the verbal scales into different factors. Such results suggest that the sensory scale items are indicative of a different semantic structure than the verbal scales in describing music, and so they are indexing different qualities (perhaps ineffable), making them potentially special contributors to understanding musical experience.

1. INTRODUCTION

Traditionally, studies on music qualities such as perceived emotions, performance styles, or timbre nuances are communicated through words. A sophisticated example of how words can be used to generate an understanding of underlying semantic constructs is the semantic differential [1], a tool that allows the measure of the connotative meaning of music through bipolar rating scales. A novel approach, based on non-verbal sensory scales applied to music, was proposed by Murari et al. [2]. Sensory scales were first introduced by [3] with the aim to study perceived qualities of colours by substituting Osgood's verbal scales with sensory ones. This approach makes use of multisensorial scales instead of the corresponding verbal scales: for instance, instead of asking the observer where he/she would place his/her impression about a piece of music along the continuum between "warm" to "cold" expressed by words, the observer immerses his/her hands in cold and warm water, deciding which sensation best "describes" the music. In Murari et al. [2], musically trained and untrained listeners were required to listen to 12 music excerpts (two experiments were carried out, involving 6 excerpts each) and to evaluate each along seven different non-verbal scales (see [2] for a detailed definition of the scales and materials used). The data showed that subjects' ratings on non-verbal sensory scales are consistent, offering interesting possibilities about the relationship between music and other sensorial information. One could speculate that the consistency was due to the direct link between the sensory experience and the verbal analog (or verbal "equivalent"), such as sensorial warmth from warm water, and the word "warm" in describing a section of music. Alternatively, the results may suggest that non-verbal scales convey specific sensations that cannot be described verbally. In other words, asking a subject to evaluate a piece of music according to the sensation of warmth felt by immersing ones hand in water does not give the same results as verbally asking whether that music piece is warm or cold. This explanation assumes that evaluation based on sensorial information is not (or is less) mediated by verbal associations. To better explain this concept, consider the word "blue". This word can mean sadness (I feel blue), a musical style (the blues), a colour etc. Therefore, the subject's evaluation may be influenced by a specific but non-unique association between the word and one of its meanings. On the contrary, if the sensory perception of the colour blue has a more limited number of interpretations/representations, the subject's evaluation may be less influenced than its verbal counterpart. Unfortunately, the results of [2] does not offer evidences supporting one or the other alternative, because the experimental design did not include a comparison between sensory scales and the verbal analogs.

This paper presents the result of a new experiment, designed to better understand the relation between sensory and verbal scales. 25 participants were asked to evaluate six musical excerpts using non-verbal sensory scales (visual, auditory, tactile, haptic and gustative) and the "equivalent" verbal scales. Three additional verbal scales were introduced to reflect Osgood's semantic differential dimensions of evaluation (using a scale with poles pleasant-unpleasant), potency (strong-weak), and activity (active-passive). Moreover, we added another three scales (very familiar-very unfamiliar; I like this piece a lot-I dislike this piece a lot; happy-sad) to see if familiarity has a systematic influence on the responses, and to determine whether a large amount of variance in the verbal and sensory ratings could be accounted for by how much the listener liked or did not like

Copyright: ©2015 Maddalena Murari, Antonio Rodà, Osvaldo Da Pos 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 piece and, finally, to cover a contemporary perspective of emotional dimensions [4, 5].

The aims of the paper are: (i) to evaluate the reliability of the non-verbal measures by comparing them with the results reported in [2]; (ii) to evaluate if and which differences can be found between non-verbal sensory scales and the analogous verbal scales.

2. METHOD

2.1 Participants

Twenty-five participants were recruited on a voluntary basis, of whom 13 rated themselves as "musician" (age range 17-49, mean age 32,15; 4 women and 9 men) and 12 as "non-musician" (age range 17-73, mean age 42,7; 6 women and 6 men). The SV group (completing the set of sensory scales before) was made up of 14 participants of whom 4 were musicians (age range 22-36, mean age 26; 4 men) and 10 were non-musicians (age range 17-56, mean age 28,4; 4 women and 6 men). The VS group (completing the set of verbal scales before) was made up of 11 participants of whom 9 were musicians (age range 17-49, mean age 38,3; 4 women and 5 men) and 2 were non-musicians (age range 41-73, mean age 57; 2 women). The SV and VS groups are not balanced in the number of musicians and non-musicians subjects; however, previous experiments [2] didn't show significant differences between musicians and non-musicians during a similar evaluation task. A questionnaire was administered to determine the participants' musical background and experience, including listing the instruments played and number of years trained in each instrument.

2.2 Stimuli

For the present study, six music pieces, representing the three main clusters of Rodà et al. [6], were chosen. Three selected stimuli were in a major tonality, while the other three were in a minor tonality. Each excerpt had a duration of about 30 seconds. A list of the stimuli is reported in Appendix A.

2.3 Materials

We prepared the following material to use for the bipolar sensory scales:

1. maluma - takete [7], two pieces of paper with the two visual forms (cm 4,3 x 4, 3);
2. blue - orange, two cards with the two colours (NCS notation: S 2055-B10G, S 1080-Y70R, cm 4,3 x 4, 3);
3. hard - soft, a piece of wood of cylindrical shape and a cylinder of polystyrene foam;
4. smooth - rough, N 1200 and N 30 sandpapers;
5. bitter - sweet: a bitter substance (Zefirus Calma Plant, 2 drops in a small cup) and water with sugar (1 teaspoon of sugar in a small cup);
6. heavy - light: a dark plastic bottle full of liquid and the same bottle without liquid;

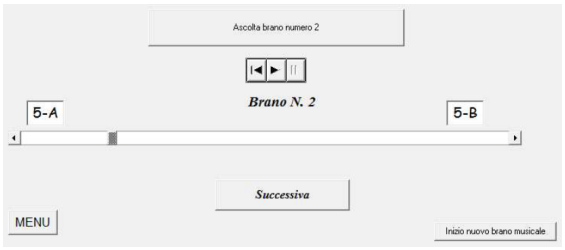


Figure 1. User interface employed in the experiment.

7. cold - warm: one cup of cold water and one cup of warm water;
8. tense - relaxed: iron wire covered with cloth and rubber band covered with cloth.

2.4 Procedure

Some participants were selected at random to complete the set of sensory scales before, and the verbal ones later; the other participants completed the study in the reverse order: verbal first, then sensory. Within the two groups of scales, the order of the scales, the order of the poles of each scale and the order of the stimuli were randomised repeatedly. The participants could listen to the stimulus as frequently as they wished. The verbal scales used had poles for item as follows:

1. maluma-takete
2. blue-orange
3. hard-soft
4. smooth-rough
5. bitter-sweet
6. heavy-light
7. cold-warm
8. tense-relaxed
9. active-passive
10. strong-weak
11. pleasant-unpleasant
12. very familiar-very unfamiliar
13. I like this piece a lot-I dislike this piece a lot
14. happy-sad

Each sound file stimulus was initiated by clicking a button on the computer screen. Each musical excerpt was heard over headphones.

A research assistant ensured that the correct pairs of materials were presented in the given order for each sensory scale item, based on a code displayed on the computer screen. In this way, non-verbal sensory scales were never explicitly associated to verbal descriptors. The procedure consisted of expressing a subjective evaluation on the characteristics of the stimulus heard by placing the indicator inside a slider at the position that was considered representative of the association strength either with the sensations on which the listener was focused (for the sensory items), or with the meaning of the verbal terms proposed (for the verbal items). All 6 excerpts were rated in one block and then again in another block. One block contained the sensory scale items and the other block contained the verbal scale items.

3. RESULTS

A two-way multivariate analysis of variance (MANOVA) was carried out with the six musical excerpts and the two groups (SV and VS) as independent variables and the 22 scales (8 sensory and 14 verbal) as dependent variables. Significant main effects were found for musical excerpt ($F(22; 120) = 13.2$ $p < .001$) and group ($F(22; 116) = 2.92$ $p < .001$). Moreover a significant interaction was found between musical excerpts and groups ($F(22; 120) = 2.26$ $p < .001$).

A post-hoc pairwise comparison was carried out using FDR correction. Table 1 shows the significance levels of the differences between pairs of musical excerpts along the eight sensory scales.

The strongest juxtaposition is represented by the couple Brahms-Chopin which was significantly different along every sensory scale except blue-orange, and by the couple Chopin-Bach, significantly different along every scale except the maluma-takete and blue-orange. Chopin's Prelude is significantly different from all the other five excerpts for hardness, roughness and tension. A similar juxtaposition is also shared by the couples Bizet-Brahms, Bizet-Vivaldi and Bizet-Mozart which are significantly different for hardness and roughness. The most striking similarity is represented by the couple Vivaldi-Bach which doesn't differ significantly in any sensory scale. Inside the cluster represented by the excerpts Brahms, Vivaldi, Mozart and Bach, we notice the strong similarity between the couples Bach-Mozart, Bach-Brahms and Vivaldi-Mozart with a significant difference only in one sensory scale. In particular, the bitterness of Mozart is a quality that significantly differentiates this stimulus from Bach, Vivaldi and Brahms. No significant difference appears for the scale blue-orange.

3.1 Factor analysis

A factor analysis with a four factor solution was conducted and the solution was rotated according to the Varimax method. This was applied three times, once using the whole group of participants and once for each of the separate groups (SV, VS). In the first case, the explained variance is 67,63%. As can be seen from Table 2, seven sensory scales (maluma-takete, hard-soft, smooth-rough, bitter-sweet, heavy-light, cold-warm, tense-relaxed) are better explained by the first factor. The second factor comprises five verbal scales (hard-soft, smooth-rough, bitter-sweet, heavy-light, tense-relaxed) together with the scales pleasant-unpleasant, very familiar-very unfamiliar, I like this piece a lot-I dislike this piece a lot. The third factor comprises the two verbal scales active-passive and strong-weak and the fourth factor comprises the scales blue-orange, both sensory and verbal. The scales maluma-takete (verbal), cold-warm (verbal) and happy-sad are not well aligned with any of the four factors, since they appear with relatively low factor loadings in every factor.

As regards factor analysis performed on separate groups, the explained variance for the SV group is 73,43% (see Table 3). Factor 1 includes eight verbal scales: hard-soft, smooth-rough, bitter-sweet, heavy-light, tense-relaxed, pleasant unpleasant, very familiar-very unfamiliar, I like this piece a lot-I dislike this piece a lot; factor 2 is made up of the sensory scales maluma-takete, hard-soft, smooth-rough, heavy-light, tense-relaxed, and the verbal scale maluma-takete; factor 3 puts together the verbal scales blue-orange, cold-warm, active-passive, strong-weak, happy-sad; factor 4 includes the remaining sensory scales: blue-orange, bitter-sweet and cold-warm.

Regarding the VS group, the explained variance is 66,57% (see Table 4). In this case, sensory scales are less consistently grouped into one factor, since they split between factor one (cold-warm, tense-relaxed), factor two (maluma-takete, blue-orange, hard-soft, bitter-sweet) and factor three (heavy-light). Verbal scales are mainly grouped into the first factor, apart from the scales blue-orange, heavy-light and cold-warm belonging to factor three and the scales active-passive and strong-weak shaping factor four.

	Factor			
	1	2	3	4
mal-tak	**-.675**	-.155	-.150	.337
blu-ora	-.062	.029	-.071	**.836**
har-sof	**.808**	.276	.027	-.188
smo-rou	**-.613**	-.332	-.429	-.172
bit-swe	**.709**	.085	-.042	.352
hea-lig	**.705**	.204	-.030	.126
col-war	**.590**	.319	-.211	.197
ten-rel	**.736**	.269	.166	-.093
mal-tak	-.399	-.439	-.121	.439
blu-ora	.221	.188	-.305	**.707**
har-sof	.529	**.632**	.275	-.045
smo-rou	-.422	**-.688**	-.210	-.018
bit-swe	.407	**.725**	-.086	.051
hea-lig	.494	**.615**	.039	.167
col-war	.379	.417	-.491	.316
ten-rel	.472	**.671**	.223	-.050
act-pas	.070	-.088	**.892**	-.163
str-wea	.229	.118	**.868**	-.129
ple-unp	-.241	**-.837**	.135	-.156
fam-unf	.017	**-.687**	.046	.003
lik-dis	-.149	**-.788**	.194	-.035
hap-sad	-.445	-.382	.457	-.328

Table 2. Scores of the coefficients of the evaluation scales and their assignment to the respective factor. The upper rows refer to non-verbal sensory scales, the lower ones to the verbal scales. All subjects.

Both the analyses on all subjects (Table 2) and on the SV group (Table 3) show a quite clear distinction between sensory and verbal scales, apart from the scale blue-orange in Table 2 and the scale maluma-takete in Table 3. For the VS group (Table 4), the first and third factors are characterized by a less accentuated division between sensory and verbal scales.

In order to determine in which way each musical excerpt is represented by the different factors, factor-based scores were generated (see Fig. 2) for each different factor analysis (all participants, SV group, VS group). An ANOVA was carried out and a graphic representation of the mean

	Mal/Tak	Blu/Ora	Har/Sof	Smo/Rou	Bit/Swe	Hea/Lig	Col/War	Ten/Rel
Bra-Viv				.028				.026
Bra-Biz	.000		.000	.000		.008		.000
Bra-Moz					.000	.013		
Bra-Cho	.000		.000	.000	.000	.000	.000	.000
Bra-Bac	.024							
Viv-Biz	.045		.009	.024		.000		
Viv-Moz					.003			
Viv-Cho			.000	.000	.000	.000		.000
Viv-Bac								
Biz-Moz	.000		.003	.004				.004
Biz-Cho			.047	.038	.006			
Biz-Bac			.006	.003		.017		.004
Moz-Cho	.024		.000	.000		.003		.000
Moz-Bac					.005			
Cho-Bac			.000	.000	.000	.000	.000	.000

Table 1. Significance p-values with FDR correction of the differences between pairs of excerpts along the sensory scales. Blank cells mean $p > .05$.

values of factor scores was created. Analysis on the results deriving from the whole group of participants shows that Chopin is significantly different from Brahms, Vivaldi, Mozart and Bach, but not from Bizet. The order of musical excerpts inside factor 1 from the highest average value to the lowest (see Fig. 2) is Brahms, Vivaldi, Bach, Mozart, Bizet and Chopin, showing that Brahms, Vivaldi and Bach share the qualities maluma, soft, smooth, sweet, light, and relaxed representing a sensation of gracefulness and gentleness as opposed to Bizet and Chopin sharing the sensory qualities takete, hard, rough, bitter, heavy, and tense which express a sensation of harshness and roughness. This fac-

	Factor			
	1	2	3	4
mal-tak	-.184	**-.710**	.239	-.118
blu-ora	.077	-.034	.503	**.686**
har-sof	.209	**.867**	.105	.114
smo-rou	-.479	**-.498**	.231	-.425
bit-swe	.147	.344	.277	**.625**
hea-lig	.140	**.638**	.206	.308
col-war	.270	.214	.155	**.734**
ten-rel	.212	**.796**	-.115	.196
mal-tak	-.317	**-.619**	.053	.374
blu-ora	.348	-.002	**.734**	.322
har-sof	**.659**	.517	-.153	.243
smo-rou	**-.806**	-.398	.029	-.169
bit-swe	**.735**	.381	.255	.162
hea-lig	**.759**	.257	.226	.165
col-war	.351	.212	**.727**	.313
ten-rel	**.757**	.446	-.064	.182
act-pas	.061	.218	**-.868**	-.059
str-wea	.296	.447	**-.722**	-.005
ple-unp	**-.819**	-.200	-.358	-.206
fam-unf	**-.777**	.088	.052	-.002
lik-dis	**-.722**	-.260	-.433	-.025
hap-sad	-.385	-.170	**-.621**	-.369

Table 3. Scores of the coefficients of the evaluation scales and their assignment to the respective factor. The upper rows refer to non-verbal sensory scales, the lower ones to the verbal scales. SV Group.

	Factor			
	1	2	3	4
mal-tak	-.212	**-.748**	-.187	-.040
blu-ora	-.108	**-.894**	.128	.015
har-sof	.551	**.592**	.236	.064
smo-rou	-.471	-.184	-.429	-.439
bit-swe	.144	**.596**	.538	.129
hea-lig	.422	.242	**.655**	.190
col-war	**.622**	.320	.325	-.274
ten-rel	**.633**	.306	.384	.009
mal-tak	**-.564**	-.517	-.065	-.120
blu-ora	-.297	-.106	**.816**	-.100
har-sof	**.631**	.437	.255	.217
smo-rou	**-.640**	-.262	-.122	-.241
bit-swe	**.557**	.377	.260	-.175
hea-lig	.500	.371	**.561**	.133
col-war	.408	.016	**.504**	-.324
ten-rel	**.649**	.310	.212	.183
act-pas	-.064	.020	-.011	**.901**
str-wea	.065	.079	-.016	**.896**
ple-unp	**-.759**	-.167	-.011	.083
fam-unf	**-.598**	.144	-.153	.334
lik-dis	**-.809**	-.057	.282	.065
hap-sad	-.256	-.340	**-.510**	.417

Table 4. Scores of the coefficients of the evaluation scales and their assignment to the respective factor. The upper rows refer to non-verbal sensory scales, the lower ones to the verbal scales. VS Group.

tor appears to be mainly related to aspects connected with arousal. Factor 2 includes 6 of the verbal sensory scales and the verbal scales pleasant-unpleasant, very familiar-very unfamiliar and I like this piece a lot-I dislike this piece a lot. It is best represented by the scales pleasant-unpleasant, and I like this piece a lot-I dislike this piece a lot. Once more it discriminates excerpt 5 as opposed to excerpt 1, 4, and 6 and the order of musical excerpts inside this factor is the same as for factor 1 (Brahms, Vivaldi, Bach, Mozart, Bizet, Chopin). It appears to be related to aspects connected mainly with valence. In particular, the qualities hard, rough, bitter heavy, tense, unpleasant, unfamiliar and "I dislike this piece a lot" convey a sensation of repulsion. On the other hand, the qualities soft, smooth, sweet, light, warm, relaxed, pleasant, familiar and "I like this piece a lot" represent a sensation of attraction. Factor 3 includes the scales active-passive and strong-weak. It discriminates Brahms as opposed to Bizet, Chopin, Mozart and Bach; and Mozart as opposed to Brahms, Bizet, Chopin and Bach. The order of the excerpts inside factor 3 is Mozart, Vivaldi, Brahms, Bach, Chopin and Bizet. This factor is mostly related with aspects connected with potency and activity. If we consider also the happy-sad scale (related to evaluation), which has on this factor the highest factor loading, factor 3 would include all three of Osgood's dimensions [1]. Factor 4 comprises the two scales (both sensory and verbal) connected with colours and the order of the excerpts along this factor is Vivaldi, Bizet, Bach, Chopin, Brahms and Mozart, with Vivaldi the most orange and Mozart the bluest.

Following Fig. 2, the scores along factors 1 and 2 (the first associated to sensory scales, the second to the equivalent verbal ones) are quite correlated even if differences can be found observing the couple of excerpts Brahms-Bach and Vivaldi-Bizet. We need to remember that this analysis includes all subjects and about an half of them evaluated the verbal scales before the sensory ones, with the possibility that the evaluation along the latter could be influenced by an association with the former. This hypothesis is supported by the factorial scores related to the SV group only: Fig. 3 shows that in this case the scores of the factors associated to sensory scales (factor 2 and 4) are quite different from the factors associated to the verbal ones (1 and 3).

4. DISCUSSION

The analysis of variance showed that subjects are able to significantly differentiate musical excerpts by evaluating them along non-verbal sensory scales. As seven of the eight sensory scales were also used in [2] and the six musical excerpts are a subset of the stimuli used in that work, it makes sense to compare the results of our previous and current experiments (subjects were of course different) in order to verify if the associations between musical excerpts and sensory scales are consistent. It is important to be aware that the experimental designs are different because in [2] no equivalent verbal scales were included and no division of groups (SV and VS) was carried out. Tables 5 and 6 show the qualities of the six excerpts based on the subjects' evaluations in the current and previous [2] experiments respectively. In particular, only ratings significantly different (at $p < .05$) from 50 (the mid-point of the evaluation scale) are reported according to t-tests.

It can be noted that the subjects' evaluations are mostly in agreement. In particular, Brahms, Vivaldi, Mozart and Bach are characterized by the qualities soft and smooth, both sensory and verbal (see Table 5). Each of these excerpts received very high scores in the scale "I like this piece a lot", and Bach's Badinerie was the most appreciated piece in this respect according to the mean score for the "liking" rating scale. Brahms and Bach also share the qualities sweet, light, warm and relaxed both from the sensory scales and from the verbal ones. Also in our previous experiment, Brahms' violin concerto and Bach's Badinerie had 6 significant features in common. In the current experiment, Bach's Badinerie is characterized by the orange verbal quality. This outcome differs from our previous study in which Bach was rated highest in "blue" according the evaluation on sensory scales. This same stimulus was analyzed also in a study by Palmer at al. [8], where the Badinerie was associated with a combination of orange and blue colours. The range of results across the studies could be attributed to qualities of the stimulus, together with a difference in liking/familiarity between the participant cohort.

Comparing evaluations on sensory scales and their equivalent verbal scales, we see that Mozart is characterized by the quality bitter only from a sensory point of view. This outcome is probably due to the fact that sensory scales allow a more direct appreciation of the musical excerpt which is not mediated by evaluative thoughts. Distinctive verbal attributes are passive, weak and the apparently incongruent coupling sad and pleasant. As pointed out by Taruffi and Koelsch [9], people appreciate sadness in music, since it seems to have a rewarding effect. Emotional responses to sad music are multifaceted and linked to a multidimensional experience of pleasure. Paradoxically, listening to sad music can lead to beneficial emotional effects since it provides a form of consolation and regulation of negative mood and emotion. Panksepp [10] found that sad music is more effective for arousing "chills" (i.e., intensely pleasurable responses to music) than happy music. Furthermore, Huron [11] proposed that the pleasure experienced through sad music is due to the consoling effects of prolactin, a hormone usually released when people are sad or weeping, and Schubert [12, 13] has argued that absorption with music allows a separation of negative emotions such as sadness from pleasure.

Regarding the factor analysis, interesting similarities can be found with the results of Da Pos and Pietto [3], who carried out an experiment using non-verbal sensory scales applied to evaluation of colours. In particular, sensory scales were grouped into factors different from verbal scales, similarly to what we found in our experiment. Moreover, one factor included all the three verbal scales deriving from the main Osgood's dimensions [1], as did the third factor of Table 3. The two maluma-takete scales (sensory and verbal) in the factor analysis of the VS group were assigned to

Brahms	Vivaldi	Bizet	Mozart	Chopin	Bach
maluma		takete	maluma	takete	
			blue		
soft	soft	hard	soft	hard	soft
smooth	smooth		smooth	rough	smooth
sweet	sweet		bitter	bitter	sweet
light	light			heavy	light
warm		warm		cold	warm
relaxed		tense	relaxed	tense	relaxed
maluma			maluma	takete	
			blue	blue	orange
soft	soft		soft	hard	soft
smooth	smooth		smooth	rough	smooth
sweet	sweet			bitter	sweet
light	light			heavy	light
warm		warm			warm
relaxed	relaxed		relaxed	tense	relaxed
		active	passive	active	active
weak		strong	weak	strong	strong
pleasant	pleasant	pleasant	pleasant		pleasant
			familiar		familiar
I like	I like	I like	I like		I like
	happy	happy	sad	sad	happy

Table 5. The qualities of the six excerpts based on the subjects' evaluation. Blank cells mean that no significant ($p > .05$) trend has been found. The upper rows refer to non-verbal sensory scales, the lower ones to the verbal scales.

Brahms	Vivaldi	Bizet	Mozart	Chopin	Bach
maluma	takete	takete	maluma	takete	maluma
			blue		blue
soft				hard	soft
smooth	smooth		smooth	rough	smooth
sweet				bitter	sweet
light				heavy	light
warm					warm

Table 6. The qualities of the six excerpts as evaluated in the experiments presented in [2]. Blank cells mean that no significant ($p > .05$) trend has been found.

the same factor (the second one in Table 3). The alignment of the verbal scale with the non-verbal sensory scale is indicative of the lack of a semantic association for the words maluma and takete (both "nonsense" words, but with their sound and orthographic shape bearing a resemblance to the shapes they indicate. For more details, see [14]).

5. FUTURE WORK AND CONCLUSIONS

Future work will continue to examine the relationships between sensory and verbal scales in describing musical qualities. The focus of the present research program involves examining semantic relationships, however we acknowledge that different variants for relationships may also exist and interact. For example, cross-modal psychophysical relationships may influence responses. Studies by S. S. Stevens and J. C. Stevens [15, 16] have found relationships between intensity of audio signals with intensity of sensations such as grip strength, redness saturation and so forth. For the current stimuli we did not do a direct comparison of the psychophysical intensity of the complex, realistic musical stimuli with the sensory scales, but some influence may be present and factors such as this may explain some of the less straight forward results we found.

In conclusion, subjects' ratings show notable consistency when compared with the results obtained in previous experiments [2, 3, 17], providing evidence for the reliability of the measurements obtained through sensory scales. Regarding the relation between sensory scales and their verbal equivalent, the order in which the rating task was completed (verbal scale first, versus sensory scale first) impacted on the ratings. Sensory scales appear to have less influence on "equivalent" verbal scales, but verbal scales do not seem to influence sensory scales. This provides weak, but important evidence that sensory scales are not, or need not be mediated by language. And so together, sensory

scales provide new perspectives in rating phenomena such as music, which are also distinct from verbal scales, and made reliably.

From the applicative point of view, such research can foster the development of innovative interfaces to browse audio digital collections. These new devices will allow users to interrelate in a spontaneous and even expressive way with interactive multimedia systems, relying on a set of advanced musical and gestural content processing tools, adopting descriptions of perceived qualities, or making expressive movements.

6. REFERENCES

[1] C. E. Osgood, G. J. Suci, and P. H. Tannenbaum, *The measurement of meaning*. Urbana, University of Illinois Press, 1957.

[2] M. Murari, A. Rodà, O. Da Pos, S. Canazza, G. De Poli, and M. Sandri, "How blue is mozart? non verbal sensory scales for describing music qualities," in *11th Sound and Music Computing Conference*, Athens, Greece, 14-20 September 2014.

[3] O. Da Pos and M. Pietto, "Highlighting the quality of light sources," in *Proc. of the 2nd CIE Expert Symposium on Appearance - When Appearance meets Lighting*, Ghent, 2010, pp. 161–163.

[4] S. Vieillard, I. Peretz, N. Gosselin, S. Khalfa, L. Gagnon, and B. Bouchard, "Happy, sad, scary and peaceful musical excerpts for research on emotions," *Cognition & Emotion*, vol. 22, no. 4, pp. 720–752, 2008.

[5] E. G. Schellenberg, I. Peretz, and S. Vieillard, "Liking for happy-and sad-sounding music: Effects of exposure," *Cognition & Emotion*, vol. 22, no. 2, pp. 218–237, 2008.

[6] A. Rodà, S. Canazza, and G. De Poli, "Clustering affective qualities of classical music: beyond the valence-arousal plane," *IEEE Trans. on Affective Computing*, vol. 5, no. 4, pp. 364–376, 2014.

[7] W. Köhler, *Gestalt psychology*, 2nd ed. New York, Liveright, 1929.

[8] S. E. Palmer, K. B. Schloss, Z. Xu, and L. R. Prado-León, "Music–color associations are mediated by emotion," *Proceedings of the National Academy of Sciences*, vol. 110, no. 22, pp. 8836–8841, 2013.

[9] L. Taruffi and S. Koelsch, "The paradox of music-evoked sadness: An online survey," *PLoS ONE*, vol. 9, no. 10, p. e110490, 2014.

[10] J. Panksepp, "The emotional sources of "chills" induced by music," *Music perception*, pp. 171–207, 1995.

[11] D. Huron, "Why is sad music pleasurable? a possible role for prolactin," *Musicae Scientiae*, vol. 15, no. 2, pp. 146–158, 2011.

[12] E. Schubert, "Loved music can make a listener feel negative emotions," *Musicae Scientiae*, vol. 17, no. 1, pp. 11–26, 2013.

[13] ——, "Enjoyment of negative emotions in music: An associative network explanation," *Psychology of music*, vol. 24, no. 1, pp. 18–28, 1996.

[14] E. Milán, O. Iborra, M. de Cordoba, V. Juárez-Ramos, M. R. Artacho, and J. Rubio, "The kiki-bouba effect a case of personification and ideaesthesia," *Journal of Consciousness Studies*, vol. 20, no. 1-2, pp. 84–102, 2013.

[15] S. S. Stevens, "Matching functions between loudness and ten other continua1," *Perception & Psychophysics*, vol. 1, no. 1, pp. 5–8, 1966.

[16] J. C. Stevens and L. E. Marks, "Cross-modality matching of brightness and loudness." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 54, no. 2, p. 407, 1965.

[17] O. Da Pos, P. Fiorentin, A. Scroccaro, A. Filippi, C. Fontana, E. Gardin, and D. Guerra, "Subjective assessment of unique colours as a tool to evaluate colour differences in different adaptation conditions," in *Proc. of CIE Centenary Conference "Towards a New Century of Light"*, Paris, 2013, p. 488.

Appendix A

Description of the musical excerpts:
1 - J. Brahms - Violin Concert in D major, op. 77, Adagio. Thematic exposition on the oboe of a slow, pure melodic line, built on the tonic major chord, and standing apart above a timbrally rich, sustained orchestra. The doubling of lines serves to reinforce the fullness of sound of the whole.
2 - A. Vivaldi - Trio Sonata in C major, RV82, Allegro. Vigorous and cheerful passage, characterized by a thematic development combining lute and violin. The violin plays rapid trills, thus complementing the lute's quick, athletic ornaments with its own sharp notes. The ascending tone is emphasized by the intensive use of progressions enriched by the continuous dialogue between lute and violin.
3 - G. Bizet - Symphony no. 1 in C major, Allegro vivo. The work starts with an opening tutti full of strength and force, like a brisk announcement. This bold first idea is answered by a timid *pp* reply by the winds which are soon harassed again by the tutti repeating the same announcement this time leading to G major.
4 - W. A. Mozart, Piano concerto Adagio, K 488. Theme in a minor key, played at a very slow tempo. Melancholic trochaic rhythm characterized by a large intervallic distance between sounds grouped by the left hand, and the melody in the high register of the right hand, creating a void in the middle of the range which reinforces the desolate aspect of the theme.
5 - F. Chopin, Prelude 22. Motif in the low register of the piano repeated obsessively and characterized by pounding octaves in the left hand, dissonant harmonies, and accompanied in the right hand by a panting rhythm, accentuating the weak part of the beat, and breaking up the violent and hopeless discourse of the left hand.
6 - J. S. Bach, Badinerie from Orchestral Suite n. 2 BWV 1067. Exposition of the main theme by the flute in the typical dance rhythm characterized by a joyous and light feeling. The orchestral accompaniment is very simple and elegant.

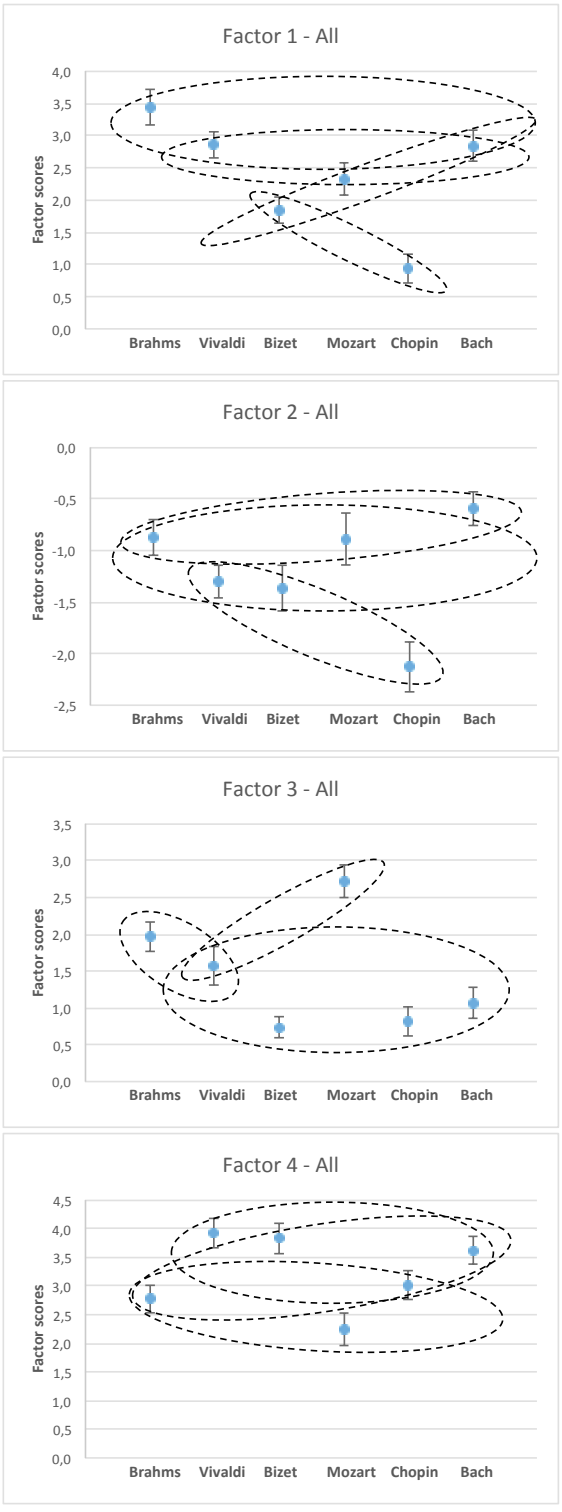


Figure 2. Scores of the music stimuli along the four main factors. Dashed ellipses group together excerpts that are not significantly different according to the ANOVA. All subjects.

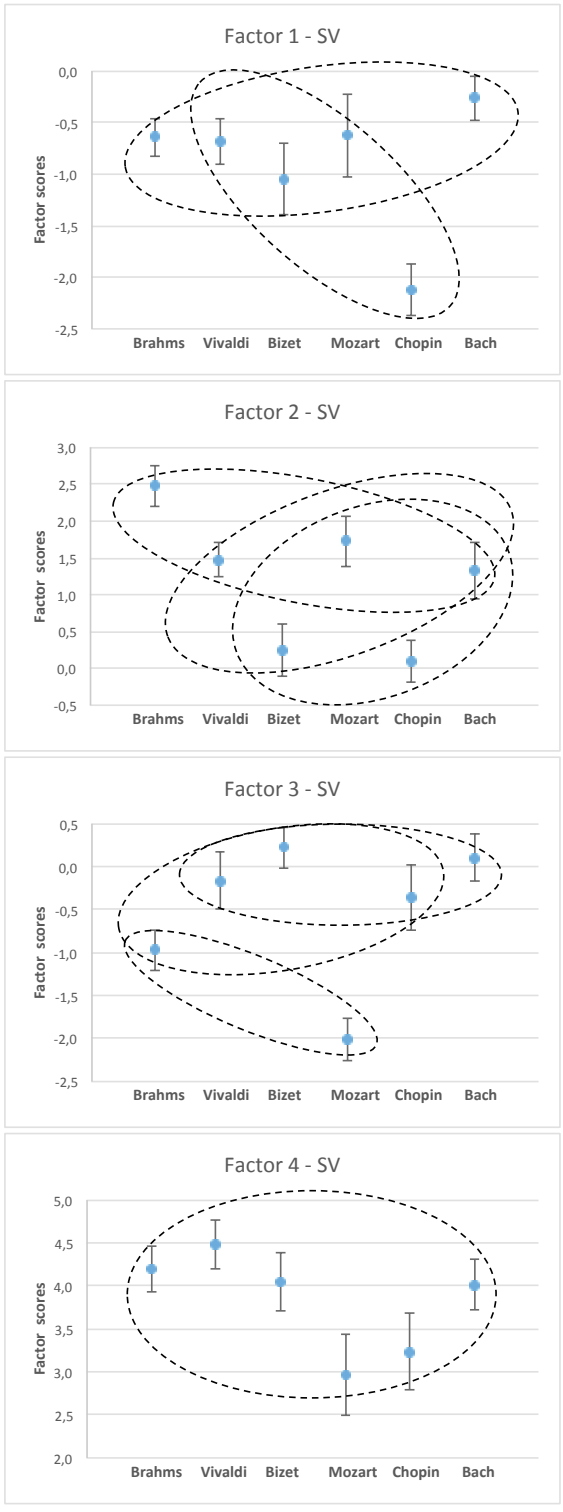


Figure 3. Scores of the music stimuli along the four main factors. Dashed ellipses group together the excerpts that are not significantly different according to the ANOVA. SV Group.

Vibrotactile Discrimination of Pure and Complex Waveforms

Gareth W. Young
Dept. Computer Science / Music
University College Cork
Cork, Ireland
g.young@cs.ucc.ie

Dave Murphy
Dept. Computer Science
University College Cork
Cork, Ireland
d.murphy@cs.ucc.ie

Jeffrey Weeter
Dept. Music
University College Cork
Cork, Ireland
j.weeter@ucc.ie

ABSTRACT

Here we present experimental results that investigate the application of vibrotactile stimulus of pure and complex waveforms. Our experiment measured a subject's ability to discriminate between pure and complex waveforms based upon vibrotactile stimulus alone. Subjective same/different awareness was captured for paired combinations of sine, saw, and square waveforms at a fixed fundamental frequency of 160 Hz (f_0). Each arrangement was presented non-sequentially via a gloved vibrotactile device. Audio and bone conduction stimulus were removed via headphone and tactile noise masking respectively. The results from our experiments indicate that humans possess the ability to distinguish between different waveforms via vibrotactile stimulation when presented asynchronously at f_0 and that this form of interaction may be developed further to advance digital musical instrument (DMI) extra-auditory interactions in computer music.

1. INTRODUCTION

Acoustic instruments provide vibratory feedback that is tightly coupled with the sound-generating module of the instrument. That is to say, the mechanisms for creating sound and audible resonances are often the same as those that are initiated by the musician. The relationships between physical interaction and the generation of sound are inseparable, and vibrations that are introduced outside of this interaction are sometimes considered as distracting or noisy. With respect to DMI design, we can no longer apply what is perceived as the innate vibrational properties of an acoustic device to a digital one, as the sound generating module is no longer tightly coupled with the gestural interface. However, with DMIs we can extend the vibrotactile feedback element beyond that of the acoustic experience.

The findings of Gillmeister and Eimer [1] have highlighted the function of vibrotactile intensity enhancements when tactile stimulus is presented synchronously with auditory stimulus. The interactions between the two stimuli produce mutual benefits and follow principles of inverse effectiveness, as well as the spatial and temporal rules of multisensory integration [2]. In the principle of inverse effectiveness, it is accepted that multisensory integration is more likely to present a stimulus as perceptually stronger than when the same unisensory stimuli are applied in isolation. Further to this, the spatial and temporal rules of multisensory integration state that the advantages of multisensory integration are strengthened when the stimuli arise from approximately the same place and in relative synchrony. Therefore, the parameters of vibrotactile feedback in DMIs can be used to support auditory output, but also expanded to include other complimentary information, such as score data, or other abstract cues from within an ensemble, with care taken not to distract or confuse the user. The application of this vibratory signal will depend ultimately on the musician's ability to process the information in relation to the audio/visual feedback they are already receiving concurrently.

Copyright: © 2015 First author et al. 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.

2. BACKGROUND

Observing the similarities between touch and hearing, we can see indications of a cross modal sensory interaction. This is apparent in terms of the type of physical energy captured; the receptors used in their detection and the relatively short overlap of the frequency domains. This is prevalent in most musical performance, the sound generation and tactile analysis frequently occur in tandem. In tasks that involve textural analysis of an object, the tactile system is dominant; however, in musical tasks, the auditory modality takes precedence. Due to the sensory dominance of hearing over tactile, the interaction between both generally goes unnoticed.

The sensations of tactile signals are bounded to a limited range (approximately 0.3 to 1000 Hz), and an individual's sensitivity to a stimulus. Following this, it can be said vibrotactile feedback from a musical instrument is secondary to that of auditory feedback in a multimodal signal. Moreover, vibrotactile feedback in a musical performance is not the primary source of feedback, but it operates in support of the auditory cues received. Most musical instruments are played with the hands, fingers, or mouth, which have the highest concentration of tactile receptors in the body. This enables fine-grained manipulation of the playing of the instrument. Further studies have shown that other parts of the body are sensitive to vibrotactile stimulus, but to a much lesser extent. The subdivisions of the vibrotactile response of the cutaneous

system are due to the arrangement of four major types of receptors in the skin. These being: the Meissner Corpuscles, the Merkel Corpuscles, the Ruffini Corpuscles, and the Pacinian Corpuscles. The upper region of the skin contains the Meissner corpuscles. These corpuscles are responsible for the transduction of light touch, stretching, and texture stimuli. Within the same region the Merkel corpuscles function to detect sustained pressure and low frequency vibration. Deeper within the skin lies the Ruffini corpuscles, which also detect sustained pressure. The deepest of the corpuscles are the Pacinian corpuscles. These are responsible for the detection of deep pressure and high frequency vibrations that are applied to the surface of the skin. The Pacinian corpuscles respond to high-speed displacement of the skin, but not when under sustained pressure.

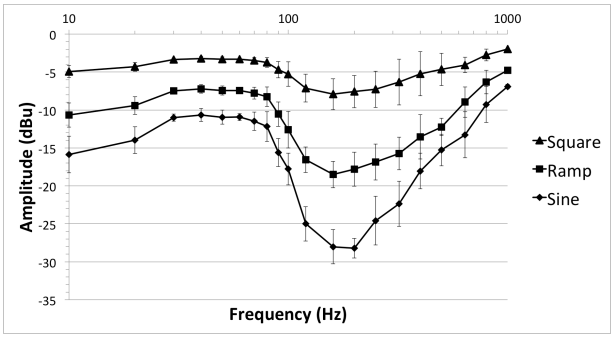


Figure 1: Threshold of perception of vibration applied via the Audio-Tactile Glove [6].

Recent psychophysical studies have focused on the human ability to discriminate between vibrotactile tonalities whilst being masked from an auditory source [3, 4, 5]. These experiments concentrate on the amplitude of fundamental sine waves and the point of which a subject can sense a vibrotactile signal of this sort. These experiments distinguish themselves from the work described here by focusing on pure tone detection or musical timbres. Our experiments have also resulted in similar findings in tactile detection levels, but include controlled complex waveforms containing a fundamental with odd harmonics, or odd and even harmonics. The sub-threshold of detection for each of these wave-shapes has been previously measured as output amplitudes in dBu (Figure 1). The sub-threshold of vibrotactile stimulus detection can be divided into distinct ranges, pertaining to the frequencies that are cutaneously detectable and the waveform of the stimulus. The main range considered is that from 0.3 Hz to 1000 Hz, which corresponds with the response range of the tactile system. Within this range, the region of 100 to 500 Hz is the most sensitive [7]. Other studies have divided this range even further [8], stating that within the span from 20 Hz to 40 Hz, the threshold for vibration detection is independent of the frequency of vibration. However, between the frequencies of 40 Hz to 700 Hz our threshold of sensitivity is a function of frequency, with peak sensitivity around 250 Hz [9]. With the amplitude of a tactile signals detection being dependent on frequency and the waveform shape being delivered, we have attempted to reduce our subject's perception of waveform intensity differences by using a fixed fundamental frequency and adhering to the waveform sub-threshold values discovered during our earlier experiments with vibrotactile feedback [6].

3. DISCRIMINATION OF PURE AND COMPLEX WAVEFORMS

Our experiment sought to investigate the relationship between the tactile receptors of the skin, and to determine if it is possible to use these to distinguish between pure and complex waveforms. As musicians are regularly exposed to combined audio and vibrotactile stimuli, we also aimed to compare musicians with non-musicians to determine if the increased exposure to combined multisensory feedback presents with increased sensitivity to vibrotactile feedback. Our previous research findings regarding stimulus amplitude detection were used to present each waveform at a relative perceptual level [6].

3.1 Stimuli

The vibrotactile stimuli applied during all experimental conditions were sine, saw and square waveforms of 160 Hz (S_1, S_2, and S_3 respectively). This frequency was chosen as it was found to have the lowest sub-threshold of perception in earlier experiments conducted with the Audio-Tactile Glove [6]. In addition, the chosen frequency lies between the musical notes D3# and E3 (on an equal temperament scale), removing any advantage a musician may have had through experience.

The output amplitude of each waveform sample was adjusted to fit within the tactile sensitivity range of 160 Hz (Figure 1). Output levels from the test equipment to the vibrotactile gloves were pre-set to the following parameters: S_1 = -25 dBu, S_2 = -17 dBu, and S_3 = -8dBu. These values were derived from group averages in our previous study. Our participants were asked to verbally verify that the amplitudes were perceptibly equal during the initial trial period. This additional consideration was given in order to assess the individual participant's ability to perceive the amplitude differences between stimuli at differing intensities. We thusly attempted to best control for this confounding influence by incorporating a subjective analysis. Relative perceptual levels of equality in amplitude were important to confirm as instead of the waveform complexities this may have had a significant influence on the individual participant's ability to discriminate between waveforms.

Waveforms were outputted via a digital-analogue audio converter (Avid Fast Track C400) with a sampling frequency of 96 kHz and 24-bit resolution. The audio output was routed through output channel one of the converter, split to the left and right glove in parallel. Participants were presented with digitally generated waveforms using Audacity (an open source wave editing software) at the pre-set fundamental (f_0 = 160 Hz). Waveform clips were recorded and then randomly selected from an audio library. Each clip consisted of a 2-second waveform sample, a one second inter-stimulus time (IST), followed by a second 2-second waveform sample.

Stimulus Pair	Response Different	Same	Same-Different (Independent Observation) Hit	False-alarm	z(H) - z(F)	p (c)$_{unb}$	d`
S_1- S_2 or S_2- S_1	0.89	0.11	0.89	0.07	2.67	0.91	3.33
S_1- S_1	0.07	0.93	0.93	0.11			
S_2 - S_3 or S_3 - S_2	0.96	0.04	0.96	0.04	3.57	0.96	4.16
S_2 - S_2	0.04	0.96	0.96	0.04			
S_1- S_3 or S_3 - S_1	0.81	0.19	0.81	0.07	2.34	0.88	3.03
S_3 - S_3	0.07	0.93	0.93	0.19			

Table 1: Proportion correct for independent observations of same-different experiment

Participants wore the Audio-Tactile Gloves, each constructed of six voice-coil actuators that are capable of outputting vibrotactile signals simultaneously at frequencies that the hand is most sensitive to. These actuators are located on each finger and on the palm of each hand. The vibrotactile waveforms were delivered to each actuator in unison. The signal was applied to both hands simultaneously in order to control for increased dominant hand sensitivity and other variances of hand sensitivity that may have existed. In order to mask incidental sound production from the glove, and bone conduction through the skeletal structure, a white noise signal was presented over Sennheiser HD 215 headphones at 60 dB SPL. The same white noise signal was applied to the lower mastoids via HiWave HIHX14C2-8 audio exciters contained within a specially constructed collar. Validated bone conduction masking parameters were followed [10].

3.2 Participants

Thirty participants partook in this experiment; three were subsequently removed as outliers. Physiological pretesting was not performed on individual participants; however, participants self-reported as having no reduced feeling or other impairments of their hands. Three participants were removed from the study as they presented with reduced sensitivity to vibrotactile stimuli. All participants were recruited from University College Cork and the surrounding community area. 17 of the participants classified themselves as musicians, having been formally trained or regularly performing in the last five years. The participants who identified as being musicians were aged 21 to 35 (MD = 24, SD = 7.23). This group consisted of 10 males and 7 females. The participants who identified as being non-musicians were aged 23 to 49 (MD = 35.5, SD = 8.15). This group consisted of 5 males and 5 females.

3.3 Experimental Conditions

The experiment examined the ability of participants to discriminate between different vibrotactile stimuli presented at the appropriate sub-threshold for the waveform type. For all experimental conditions, participants were seated in a soundproof room with both forearms resting on armrests, and hands in a relaxed position. Participants were asked to make same-different judgments for each trial. This experimental procedure was chosen to remove any ambiguity in participants explaining the differences they experienced between the three waveforms presented. Participants were asked to indicate if the two stimuli were the same or different by saying "Same" or "Different". Our objectives were not to determine the specific cue of the stimuli, but to simply determine the discriminability of each waveform. Three blocks of recorded trials followed a practice period of two blocks. Whilst providing our participant with the opportunity to familiarise themselves with the experimental procedure, the practice period also reaffirmed sensory 'equal loudness', insuring that signal intensity was not a discriminating factor. All participants indicated that the signals presented equal loudness at this stage. Each of the recorded trials consisted of the presentation of two stimuli, which were either the same or different. The waveform pairs were presented in counterbalanced order. All possible waveform pairs were presented within each block. Each block of samples contained three matched and six mismatched pairs. Thus, the recorded results consisted of 27 clips in total; 9 matched and 18 mismatched paired samples.

4. RESULTS

A Wilcoxon Signed Rank Test revealed that there was no statistically significant effect in the order of waveform presentation; S_1- S_2 / S_2 - S_1 (z = 0, p = ns), with no significant effect size (r < 0.00); S_2 - S_3 / S_2 - S_3 (z = 1.13, p = .26), with a small effect size (r = 0.14); S_3 - S_1/ S_1- S_3 (z = 1.73, p = .083), with a medium effect size (r = 0.22). There was also no change in the median for each waveform pair. Therefore, it was deemed possible to collapse the proportion of correct response results across these complementary pairs. Table 1 shows the same-different responses for each stimulus pair after collapsing. This Data was subjected to a signal detection theory analysis and the effects of bias were removed. Specifically, hit and false alarm rate data were analysed to calculate a sensitivity measure of d` and an unbiased proportion correct probability, determined from table 5.3 in MacMillan and Creelman's textbook [11]. An independent-samples t-test was conducted to compare the adjusted mean percentage correct of musicians and non-musicians. There was no significant difference in scores for musicians (M = 0.89, SD = 1.15) and non-musicians (M = 0.94, SD = 0.1; t (14.09) = -1.06, p = .3, two-tailed). The magnitude of the differences in the means (mean difference = 0.17, 95% CI: –0.17 to 0.06) was small (eta squared = 0.04).

5. DISCUSSION

The results from our experiment identified how the participants successfully recognised different waveforms based on waveform shape (as distinct from envelope) when presented in isolation to the hand. These findings support previous research findings undertaken by Russo et al. relating to the vibrotactile discrimination of musical timbres [5]. However, our experiment here has expanded some of these findings further; by applying the stimuli directly to the subject's hands via voice-coil transducers; applying waveforms that have a controlled waveform envelope; and finally compared musicians with non-musicians. The data gathered from this experiment supports a theoretical operation of combined critical band filtering that may be carried out by the sensory receptor arrays within human glabrous skin; specifically, in the ventral portion the fingers and the palmer surfaces of the hand at a fixed fundamental of 160 Hz. We predict that the stimulus of the four main types of mechanoreceptors outlined earlier, and their individual responses to mechanical displacement, function as frequency-tuned filters whilst experiencing complex tones. This filtering of complex tonality into component frequencies, with relative intensities, contributes to our tactile perception of differing timbres.

Studying the subjective, contextual, and physiological gestural characteristics of musical instrument interactions, highlights the importance of feedback via primary, secondary, and other lesser pathways from instrument to musician. The tactile component of haptic feedback, which is considered in this research, provides an insight into the complexity of primary/secondary and passive/active feedback in multimodal communications. During the playing of musical instruments, the auditory system takes on the role of primary feedback processor. In this context, the role of the visual and haptic senses operates on the secondary feedback signals, primarily relating to the instrument's physical response to gestural inputs. Also worthy of note is the difference between active and passive feedback, as passive feedback was solely applied in our experiments. Passive feedback relates to the feedback provided through the physical characteristics of the system in use, i.e. the manner in which the system input responds when affected. Active feedback is produced by the system in response to a specific user action, a sound produced within for example. Further experimentation may reveal supplementary information about the role of active feedback in musical performance.

6. CONCLUSIONS

We have concluded from our experiments that humans possess the ability to distinguish between different waveforms via vibrotactile stimulation alone when presented asynchronously at a fundamental frequency of 160 Hz. We conducted an experiment to confirm that humans are capable of distinguishing between pure sinusoidal and complex waveforms with non-sinusoidal periodic shape containing odd only (square) and odd and even (saw) harmonic content at f_0. Our experiment yielded positive results, with 92% of participants successfully identifying waveforms when presented asynchronously. From this, it can be argued that the adoption of a combined psychophysical approach is required to reinforce the role of somatosensory integration in timbral discrimination tasks that are carried out on digital devices. This will hopefully allow researchers and DMI designers to combine multisensory interfaces that are transparent and intuitive to operate during musical tasks. The linking of tactile feedback to audio output can also assist in reducing computer-processing power that may be required in outputting extra channels of feedback in haptic systems.

7. REFERENCES

[1] Gillmeister, H., Eimer, M., "Tactile enhancement of auditory detection and perceived loudness," Brain Research, 1160, 2007, pp. 58-68.

[2] Meredith, M. A., and Stein B. E., "Visual, auditory, and somatosensory convergence on cells in superior colliculus results in multisensory integration," In Journal of neurophysiology 56 (3), 1986, pp. 640-662.

[3] Soto-Faraco, S., Deco, G., "Multisensory contributions to the perception of vibrotactile events," Behavioural Brain Research, 196 (2), 2009, pp. 145-154.

[4] Wilson, E. C., Reed, C. M., Braida, L. D., "Integration of auditory and vibrotactile stimuli: Effects of phase and stimulus-onset asynchrony," The Journal of the Acoustical Society of America, 126, 2009, pp. 960-1974.

[5] Russo, F. A., Ammirante, P., Fels, D. I., "Vibrotactile discrimination of musical timbre," Journal of Experimental Psychology: Human Perception and Performance, 38(4), 2012, p. 822.

[6] Young, G., Murphy, D., Weeter, J., "Audio-tactile glove," In Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13), Maynooth, Ireland, 2013.

[7] Chafe, C., "Tactile audio feedback," in Proc. of the Int. Computer Music Conference, September 1993, pp. 76-76.

[8] Verrillo, R. T., "Vibration sensation in humans," Music Perception, 1992, pp. 281-302.

[9] Birnbaum, D. M., Wanderley, M. M., "A systematic approach to musical vibrotactile feedback," in Proc. of the Int. Computer Music Conference (ICMC), Vol. 2, August 2007, pp. 397-404.

[10] Wilson, E. C., Reed, C. M., Braida, L. D., "Integration of auditory and vibrotactile stimuli: Effects of frequency," The Journal of the Acoustical Society of America, 127(5), 2010, pp. 3044-3059.

[11] Macmillan, N. A., Creelman. C. D., "Detection theory: A user's guide," Psychology press, 2004.

An Exploration of Mood Classification in the Million Songs Dataset

Humberto Corona, Michael P. O'Mahony
Insight Centre for Data Analytics
School of Computer Science and Informatics
University College Dublin, Ireland
firstname.lastname@insight-centre.org

ABSTRACT

As the music consumption paradigm moves towards streaming services, users have access to increasingly large catalogs of music. In this scenario, music classification plays an important role in music discovery. It enables, for example, search by genres or automatic playlist creation based on mood. In this work we study the classification of song mood, using features extracted from lyrics alone, based on a vector space model representation. Previous work in this area reached contradictory conclusions based on experiments carried out using different datasets and evaluation methodologies. In contrast, we use a large freely-available dataset to compare the performance of different term-weighting approaches from a classification perspective. The experiments we present show that lyrics can successfully be used to classify music mood, achieving accuracies of up to 70% in some cases. Moreover, contrary to other work, we show that the performance of the different term weighting approaches evaluated is not statistically different using the dataset considered. Finally, we discuss the limitations of the dataset used in this work, and the need for a new benchmark dataset to progress work in this area.

1. INTRODUCTION

Most of the research on music classification is based on features obtained by audio analysis [1–3]. However, previous work by Besson et al. [4] concluded that semantic (lyrics) and harmonic (tunes) information are processed independently by the brain, even when these information sources are closely related to each other. This indicates the relevance of lyrics in music classification, as it can be complementary to the study of harmonic information.

It is noteworthy that the results obtained by different previous work are not consistent in their methodology or outcomes. For example, [5, 6] found that lyrical features can outperform audio features in music mood classification in certain categories. However, [7] suggests that lyrics perform worse than cultural or audio features. Moreover, comparing results from previous work is difficult, as the class labels selected for classification are not consistent, and the datasets used are different and/or are not publicly available.

In this paper we focus on the analysis of lyrics for mood classification. We follow state-of-the-art approaches to infer music mood using social tags, and study three different levels of granularity for mood classification. We study a vector space model (VSM) representation of songs using different term-weighting approaches, with the aim of establishing a comprehensive benchmark for music mood classification using lyrical features. Moreover, we present a feature analysis to further explain the main findings of this work.

As lyrics are copyrighted material, it is difficult to legally obtain a large dataset. Contrary to previous work, which are based on different small-scale datasets [8,9], we use the *Million Song Dataset (MSD)* [10], a large freely-available dataset which facilitates the reproducibility and comparison of the findings presented in this work. With this goal in mind, we also make our source code publicly available [1] .

The remainder of this paper is organised as follows. First, Section 2 describes the related work. Section 3 describes the datasets used and Section 4 presents the term-weighting metrics studied in this work. Section 5 presents a feature analysis of song lyrics. Section 6 introduces the classification approach studied in this paper and the results obtained. Finally, in Section 7, conclusions are presented.

2. RELATED WORK

Most of the existing music classification approaches rely on audio analysis to infer mood or genres [11, 12], while other approaches combine audio analysis with other features, such as cultural features [7] or lyrics [13]. However, the exploration of lyrics alone as a source of information for music classification is an interesting problem and it has not been widely explored. In this section we present an overview of the main approaches for mood representation and music mood classification using lyrics.

2.1 Mood Representation

Music moods are difficult to infer: people perceive them differently [14] and they are culturally dependent [15]. Moreover, some songs (e.g., *Bohemian Rhapsody* by *Queen* [2]) express a wide range of moods over the course of the song.

Copyright: ©2015 Humberto Corona 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] http://github.com/hcorona/SMC2015
[2] http://open.spotify.com/track/1fNo4jzUtg9EC0yyHcZY5j

Most of the proposed mood ontologies rely on models developed in the psychology field — *Russell's model of affect* [16] being one of the most widely used. This model is based on the evidence that the affective dimensions are built in a *highly systematic fashion*, instead of being independent dimensions. Each mood $m \in M$ is mapped onto a two-dimensional space defined by valence v (which measures the good–bad dimension of sentiment) and arousal a (which measures the active–passive dimension of sentiment). Therefore, in this model each mood can be represented by a vector in the two-dimensional valence-arousal space $m \in M = (v, a)$.

Russell's theoretical model has been adapted to the music classification problem [17] using social tags to infer the categories. The authors also expose the lack of consensus on the names for concepts to be learned; some authors refer to *mood* while others refer to *sentiment* to define the same concept. There is also little consensus in defining the mood categories to classify, which makes comparing research output in this area problematic.

In this work, we use the term *mood* to refer to the categories to be learned in the classification problem. Moreover, we infer mood from social tags as proposed in [17, 18]. Then, we group those moods into four categories, following Russell's model of affect. This allows us to build a large dataset in which the mood groups are clearly defined. Moreover, the approach facilitates different levels of granularity that can be used in the classification task.

2.2 Music Classification using Lyrics

Hu et al. [8] propose a method for detecting mood for 500 manually labeled Chinese songs using lyrics. The approach maps mood into a two-dimensional space of valence and arousal and uses a translated and expanded version of the ANEW (Affective Norms for English Words) dataset [19]. Then a fuzzy clustering method is used to group the lyrics' sentences according to their mood and to extract one prominent mood from each song. The results show that lyric mood are more correlated to valence than to the arousal dimension.

Downie et al. [18] propose a lyric-based approach to mood classification using a binary SVM classifier. The article proposes a vector space model feature set, combined with other statistical textual features such as part-of-speech tags. The results are evaluated using a private dataset with 5,585 songs and using the 18 mood categories presented in [20]. The results show that the combination of both audio and lyrical features can improve classification performance. In later work [21], the authors explore different lyrical features and modifiers, such as stylistic features or features obtained from the ANEW dataset. The results show that a lyrics-based classifier can outperform an audio based classifier for some mood categories.

Kim et al. [22] also explore lyric-based mood classification. The approach uses partial syntactic analysis to extract emotions or mood from songs, achieving an accuracy of around 60% when evaluated in a manually labeled dataset of 500 Korean songs. This paper proposes an approach which includes novel features such as negation detection, time of emotion and change of emotion. A different approach is adopted by Kumar et al. [23], who use SentiWordNet [3] to extract mood features from 185 lyrics labeled with one of four mood categories: *happy, angry, love* and *sad*. This work compared three classifiers: KNN, SVM, and Naïve Bayes; the latter classifier performed best, achieving a classification accuracy of up to 81%.

Dodds et al. [24] use features extracted from lyrics and the ANEW dataset to measure the sentiment of songs, (also from blogs and State of the Union presidential speeches). The aim of the work is to quantify the evolution of the overall happiness in the different contexts. The approach calculates the average valence of each instance (song, blog post or speech) as a measure of happiness. The results show that, for example, valence can help distinguish between genres, when a large number of songs are considered.

From the related work it is clear that mood classification of music using lyrics is an emerging and interesting problem. However, it is difficult to compare previous findings since different works have reached contradictory conclusions based on experiments carried out on different private datasets using different evaluation methodologies. Thus, in this paper we present a comparison of different approaches for classifying music mood using lyrical features, and evaluate them using a large freely-available dataset using categories derived from Russell's model of affect.

3. DATASET

We perform our experimental studies using the *Million Song Dataset (MSD)* [10]. It is a large, freely-available dataset, which contains rich metadata and audio features for one million contemporary popular music tracks. We also use the *LastFm* [4] and *MusixMatch* [5] datasets, which expand the original *Million Song Dataset* providing metadata and lyrics for a subset of tracks.

The *LastFM dataset* contains song-level tags for more than 500,000 songs. The mood categories are derived using the social tags found in this dataset, following the approach proposed in [17, 18].

The *MusixMatch dataset* contains lyrics for 237,662 songs. Each song is described by word-counts of the top 5,000 stemmed terms across the set [6]. Specifically, we use the songs from the MSD for which social tags and lyrics are available. Furthermore, we only consider English language lyrics in this study. Thus, the resulting dataset used in this work contains 32,302 songs.

The use of this dataset is key regarding the reproducibility of the work presented here. However, given the format of the dataset (only word-counts for the top 5,000 terms are

[3] SentiWordNet: a database of sentiment information for english words, designed for opinion mining. `http://sentiwordnet.isti.cnr.it/`

[4] Last.fm dataset, the official song tags and song similarity collection for the Million Song Dataset, available at: `http://labrosa.ee.columbia.edu/millionsong/lastfm`.

[5] musiXmatch dataset, the official lyrics collection for the Million Song Dataset, available at: `http://labrosa.ee.columbia.edu/millionsong/musixmatch`.

[6] The terms were selected by its document frequency, normalised by the term frequencies in each song. We do not perform any post-processing on this set of terms (i.e. stop-words are not removed).

provided), our analysis is limited to a vector space model representation of songs, and more sophisticated natural language processing techniques [25] cannot be considered. This is a significant limitation of this particular dataset from a classification perspective as we will discuss further below.

3.1 Building the Mood Dataset

Three levels of granularity are considered for mood classification. To build the dataset we select a subset of songs for which the mood-related tags described in [18] are available. We select songs using the same criteria as used for the *MIREX 2009 mood multi-tag dataset* [7]; a song has to be tagged at least twice with one term in a tag group, or with at least two terms in a tag group, each at least once. Moreover, we remove repeated songs, (i.e. songs which have the same title and lyrics, but different ids in the dataset).

The mood groups are inferred as described in [18], where different *LastFM* tags are grouped to form a subset of predefined groups. For example, the group *G29* contains songs tagged as *aggression* and *aggressive*.

Finally, the mood quadrants as described in [8] are considered, where each quadrant represents a positive or negative value for valence and arousal. Table 1 shows the mood tags, groups and quadrants used in this work [8].

Tag	Group	Quadrant
aggression, aggressive.	G29	v^-a^+
angst, anxiety, anxious, etc.	G25	
anger, angry, choleric, fury, etc.	G28	
excitement, exciting, thrill, etc.	G1	v^+a^+
upbeat, gleeful, enthusiastic, etc.	G2	
cheerful, festive, jolly, etc.	G6	
happy, happiness, happy music, etc.	G5	
depressed, blue, dark, gloom, etc.	G16	v^-a^-
sad, sadness, unhappy, etc.	G15	
grief, heartbreak, sorrow, etc.	G17	
brooding, contemplative, etc.	G8	v^+a^-
alm, comfort, quiet, etc.	G12	

Table 1. Mood tags, groups and quadrants.

To illustrate the above, consider the song *Orchestra of Wolves* [9], by the British hardcore-punk band *Gallows*. This song is tagged as *aggressive* in the *LastFM* dataset, and therefore it is included in mood group $G29$ and quadrant v^-a^+ (given its negative valence and positive arousal values).

4. TERM-WEIGHTING SCHEMES

In this work, the vector space model is used to represent documents (songs), where each document $d = (t_1, t_2, \cdots, t_y)$ is represented by a vector in the y-dimensional term space.

[7] http://www.music-ir.org/mirex/wiki/2013:Audio_Tag_Classification

[8] Only tags which are associated with at least 100 songs are considered. Moreover, we discard groups G7, G9, G11, G14, G31 and G32 because they do not have enough tags or they can not be easily described in the valence-arousal space.

[9] http://open.spotify.com/track/5BorbORef4VQUlNOjAjoDT

The basic term weighting scheme we consider is the binary approach, in which each element of the vector is set to 1 or 0 to indicate the presence or absence of the corresponding term. A number of other term weighting schemes have been proposed in the literature [26,27]; in what follows, we describe some well known term-weighing schemes which are used in this work.

Term Frequency (tf) [27] accounts for the number of times a term t occurs in document d (denoted by $tf_{t,d}$). The rationale for this scheme is to assign higher weights to frequently occurring terms, since such terms are likely to be more characteristic of document content. Several normalisation approaches have been proposed for the original term frequency metric. Here, we use a standard logarithm normalisation, as shown in Equation 1.

$$ntf_{t,d} = \log\left(1 + tf_{t,d}\right). \tag{1}$$

Term frequency – inverse document frequency (tf-idf) combines the *tf* metric described above, with inverse document frequency (*idf*), which gives higher weights to terms which are rare in the collection. The *tf-idf* metric [28, 29] for a term t in document d is calculated as the product of $tf_{t,d}$ and $idf_{t,D}$, as shown in Equation 2.

$$tf\text{-}idf_{t,d,D} = tf_{t,d} \cdot \log \frac{|D|}{df_{t,D}}, \tag{2}$$

where the document frequency ($df_{t,D}$) is the number of documents in the collection D that contain the term t.

BM25 [30], is a sophisticated term-weighting scheme which has been widely used in text classification and retrieval. It is computed as per Equation 3:

$$BM25_{t,d,D} = \log \frac{|D| - df_{t,D} + 0.5}{df_{t,D} + 0.5} \frac{(k_1+1)tf_{t,d}}{k_1((1-b)+b\frac{L}{\bar{L}})+tf_{t,d}}, \tag{3}$$

where L is the document length and $\bar{L}$ is the average document length in the collection D. In this work, the parameters k_1 and b are set to typical values of 1.20 and 0.75, respectively [27].

Delta tf-idf [31] is a scheme specifically proposed for sentiment classification. As shown in Equation 4, the term frequency of a term is multiplied by the δ function (Equation 5), which measures the relative document frequencies of a term in positive and negative instances. Thus, higher weights are assigned to terms which appear primarily in one class [10].

$$delta\ tf\text{-}idf_{t,d,D} = tf_{t,d} \cdot \delta_{t,D}. \tag{4}$$

$$\delta_{t,D} = \log_2 \left(\frac{df_{t,D^+} + 1}{df_{t,D^-} + 1} \right). \tag{5}$$

In the above, df_{t,D^+} and df_{t,D^-} are the document frequencies for term t in documents labeled as positive and negative, respectively.

[10] The original weight results in an infinite or undefined value if a particular term does not appear at least once in both classes. Thus, we modify the original equation by adding 1 to the document frequencies, as shown in Equation 5.

5. FEATURE ANALYSIS

In this section, we perform a preliminary feature analysis of song lyrics, examining how does the term distributions and different term weighting schemes affect the classification performance. For the sake of clarity, we perform the analysis on the mood quadrants dataset; however, similar trends are found in the mood groups and mood tags datasets.

We first study the term distribution across documents (songs) and classes (mood quadrants). To achieve high classification performance using lyrics alone, the vocabulary should be very different across moods. If many of the same terms occur in all classes, it will be difficult to classify those songs that contain these terms.

In total, there are 4,481 distinct terms in the dataset (i.e., vocabulary size). From Table 2, it can be seen that the majority of these terms (between 3,903 and 4,248 terms) occur in all classes. The overall distribution of terms across classes is as follows: 365 terms appear in a single class, 328 terms appear in two classes, 300 terms appear in 3 classes and 3,488 terms are common to all four classes. Thus, only a very small fraction of terms are unique to a single class, between 25 (class v^-a^+) and 180 (class v^+a^-) terms, indicating that the vocabulary of lyrics is, to a high degree, common across the four moods considered.

	v^+a^+	v^+a^-	v^-a^+	v^-a^-
Number of instances	6,973	14,685	1,958	8,686
Number of distinct terms per class	3,903	4,248	3,616	4,106
Number of unique terms per class	57	180	25	103
Mean (std. dev.) number of distinct terms per song	65 (48)	46 (50)	78 (40)	61 (49)
Mean (std. dev.) number of terms per song	189 (147)	134 (160)	224 (139)	180 (160)

Table 2. Term statistics for the mood quadrant dataset.

Table 3 shows the top ten terms for each mood quadrant, where the rank is produced by measuring the correlation between the term and the class, using Pearson correlation [32]. Nine of the top terms of the v^-a^+ quadrant shown in the table are intuitively related with moods from this quadrant (*aggressive, angry, etc.*). However, while some terms are correlated to one particular class, the same terms (*got, get, yeah*) are most highly correlated to both the v^+a^+ and v^-a^- mood quadrants. These are *connector* terms that are not related to mood. Moreover, a number of the top terms shown in Table 3 are stop-words (or at least would be considered as such in traditional information retrieval contexts), indicating the relative lack of discriminating terms in the dataset.

Table 2 also presents statistics on the total number (song length) and the number of distinct terms per song per class. While differences in these statistics are apparent — for example, on average, songs in class v^+a^- tend to be short while those in class v^-a^+ are the longest — there is significant variance evident in these statistics, thereby limiting their value from a classification perspective.

Rank	v^+a^+	v^+a^-	v^-a^+	v^-a^-
1	got (0.14)	dead (0.075)	love (0.231)	got (0.113)
2	get (0.14)	f**k (0.068)	f**k (0.217)	get (0.094)
3	yeah (0.136)	death (0.067)	hate (0.157)	yeah (0.09)
4	it (0.119)	love (0.062)	dead (0.155)	die (0.082)
5	oh (0.112)	die (0.061)	kill (0.149)	pain (0.08)
6	gonna (0.11)	scream (0.057)	blood (0.143)	it (0.077)
7	up (0.108)	blood (0.055)	s**t (0.13)	babi (0.074)
8	a (0.104)	hate (0.05)	burn (0.124)	tear (0.074)
9	do (0.101)	the (0.048)	death (0.122)	you (0.074)
10	you (0.1)	hell (0.047)	die (0.119)	up (0.073)

Table 3. Top terms ranked by Pearson correlation.

Figure 1 presents a histogram of term frequency (*tf*) values per song in the dataset, calculated over all songs. The graph shows the term frequency values on the horizontal axis and the count for each value on the vertical axis (both axis are presented in logarithmic scale). As can be seen, the vast majority of terms (98%) occur just once in songs, while only 1.1% of terms occur twice. Given these findings, little or no difference in classification results can be expected when the binary or term frequency weighting schemes are applied.

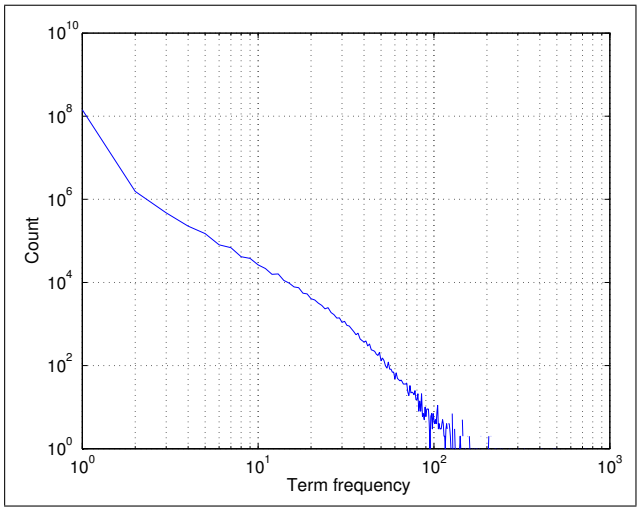


Figure 1. Term frequency distribution.

Figure 2 shows a histogram of the document frequency values (*df*) for all terms and documents. The horizontal axis shows the document frequency value for each term, with corresponding counts shown on the vertical axis (both axis are presented in logarithmic scale). The figure shows that the document frequency histogram follows a long-tail distribution, where most terms appear in a small subset of documents; for example, 1,044 terms (23%) appear in 20 documents (songs) or less, while 272 terms (6%) appear in more than 2000 (out of a total of 32,302) documents. Thus, this distribution of terms across documents is likely to limit the effect of *idf* term weighting. Moreover, the *idf* scheme does not consider term distribution with respect to class, and hence we also consider the *delta tf-idf* term weighting scheme, which does take class distribution into account.

The distribution of values for the δ function (Equation 5) is shown in Figure 3, where the horizontal axis represents δ values, while the vertical axis shows the count for each

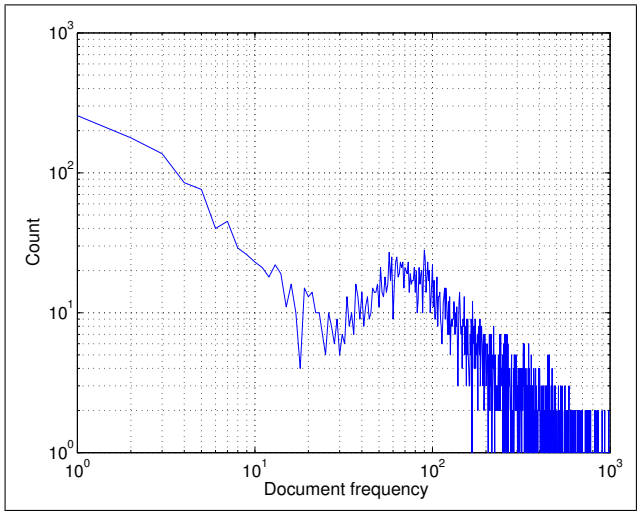


Figure 2. Document frequency distribution.

value, presented in a logarithmic scale. Each of the lines in the graph corresponds to a mood quadrant [11] . From the figure it is clear that a large number of terms are evenly distributed among the classes (i.e. at $\delta \approx 0$), while, on average across the mood quadrants, only 13% and 2% of term $delta$ values are beyond ± 1 (i.e. corresponding to a ratio of 2:1 or above of term distribution across classes) and ± 2 (i.e. a ratio or 4:1 or above), respectively. Thus, given the distribution of terms across classes, the *delta tf-idf* weighting scheme may not appreciably affect classification performance.

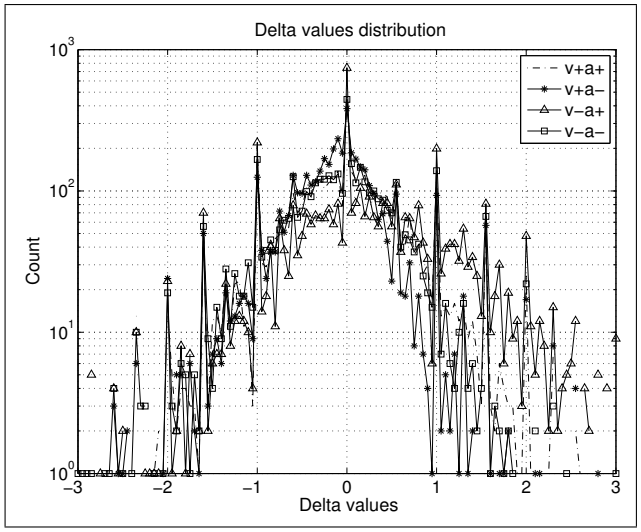


Figure 3. Distribution for δ values.

The analysis presented in this section, in particular the relative lack of discriminating terms in the dataset, may be an artefact of how the 5,000 terms were selected for inclusion in the *MusixMatch* dataset. As such, the analysis indicates a limitation in the use of this dataset for lyrics-based mood classification, a point to which we will return later in the paper.

[11] For each mood quadrant, δ values for terms are computed based on a random selection of 1,000 positive songs (i.e. from the mood quadrant in question) and 1,000 negative songs (i.e. from other mood quadrants).

6. MOOD CLASSIFICATION

We adopt a supervised classification approach where songs are represented using the vector space model. We experiment with the term weightings approaches described in Section 4 (*binary*, *tf*, *tf-idf*, *BM25* and *delta tf-idf*), comparing their performance using the three different mood granularities (i.e. class labels) described in Section 3.1. With this experiment, we aim to present a comprehensive and reproducible evaluation of music mood classification based on lyrics using the large, publicly available *Million Song Dataset*.

6.1 Experimental Methodology

We experiment with three different datasets in this evaluation, where each song is labelled according to one of the three mood granularities (i.e. mood quadrants, groups or tags) as shown in Table 1. In particular, balanced binary classifiers are created for each mood granularity by randomly selecting 1,000 positive training instances from each class; 1000 negative training instances are also randomly selected from other classes [12] .

Classification was performed using the Weka machine learning framework [33] with the LIBLinear L2-SVM classification algorithm [34], which is known to perform efficiently on large sparse datasets. Moreover, SVM classifiers have been used in the past in many binary classification scenarios with success [21, 31, 35].

Classification performance is evaluated using a standard 5-fold cross validation approach using the accuracy metric:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \qquad (6)$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

A Kruskal-Wallis test [36] at the 0.05 level is performed to determine whether statistically significant differences in results exist between the various term weighting schemes. Finally, the *delta tf-idf* weights are computed over training set instances only and these weights are then applied to test set instances.

6.2 Results

Tables 4, 5 and 6 show the results for the three different mood granularities and term-weighting schemes considered. Overall, it can be seen that the performance of the different term weighting approaches is very similar in terms of classification accuracy; no statistically significant differences in results were found. These results are expected given the analysis presented in the previous section. For example, most (98%) of the term frequency values seen in the dataset are equal to one. Thus, classification accuracy using the *tf* weighting scheme is close to the binary representation. Further, the distribution of terms

[12] When less than 1,000 positive instances are available, the maximum number of positive instances are selected, together with an equal number of negative instances. In the group and tags dataset, the mean number of positive (and negative) training instances is 744 and 426 per class, respectively.

across documents and classes in the dataset also limited the effect of the *idf*, *BM25* and *delta tf-idf* schemes.

Table 4 shows the results for mood quadrant granularity. Here, we can see that the classifier performs best for the v^-a^+ mood, which is interesting given that this class contains the least number (25) of unique terms (see Table 2). This result may be due to songs in this class containing the greatest numbers of total and distinct terms, although further analysis is required to test this hypothesis.

mood	size	accuracy				
		binary	tf	tf-idf	BM25	δ tf-idf
v^+a^+	1000	0.561	0.572	0.586	0.569	0.581
v^+a^-	1000	0.525	0.537	0.536	0.532	0.520
v^-a^+	1000	0.638	0.656	0.626	0.653	0.636
v^-a^-	1000	0.554	0.562	0.549	0.560	0.541

Table 4. Classification results for each mood quadrant.

Table 5 shows the classification results for mood groups. The best results are obtained for groups G29 (*aggression*, *aggressive*) and G28 (*anger*, *angry* etc.), where classification accuracies of 0.695 and 0.671 using binary term weighting are achieved, respectively. The remaining mood groups all have accuracies less than 0.6 (binary term weighting). Since both G29 and G28 belong to the v^-a^+ quadrant, these results confirm that the lyrics-based classification approach works well for songs in this quadrant.

mood	size	accuracy				
		binary	tf	tf-idf	bm25	δ-tf
G5	1000	0.566	0.573	0.559	0.570	0.542
G12	1000	0.549	0.548	0.525	0.562	0.517
G2	1000	0.555	0.571	0.571	0.570	0.558
G29	619	0.695	0.720	0.690	0.718	0.670
G28	1000	0.671	0.651	0.648	0.672	0.628
G1	196	0.574	0.571	0.543	0.567	0.543
G8	561	0.536	0.516	0.515	0.516	0.521
G15	1000	0.552	0.534	0.522	0.538	0.515
G6	530	0.555	0.558	0.555	0.558	0.531
G25	267	0.586	0.545	0.526	0.586	0.520
G17	749	0.592	0.599	0.571	0.592	0.570
G16	1000	0.599	0.600	0.575	0.598	0.574

Table 5. Classification results for each mood group.

Finally, the results obtained for the individual mood tags (Table 6) also align with the above findings, where high classification accuracies are seen for songs with tags belonging to the G29 group. Although. the best performance (0.767) is achieved for the tag "cool down" (which belongs to group G12 and quadrant v^+a^-), this particular tag appears infrequently in the dataset, thereby limiting its effectiveness.

mood	size	accuracy				
		binary	tf	tf-idf	bm25	δ-tf
mellow	1000	0.519	0.516	0.518	0.523	0.514
chillout	1000	0.562	0.558	0.551	0.553	0.542
happy	1000	0.562	0.568	0.554	0.561	0.558
aggressive	589	0.699	0.698	0.666	0.705	0.654
angry	821	0.649	0.668	0.633	0.665	0.624
soothing	271	0.505	0.494	0.494	0.514	0.522
melancholic	1000	0.557	0.577	0.569	0.570	0.554
calm	535	0.480	0.479	0.498	0.485	0.491
sad	1000	0.557	0.559	0.563	0.553	0.543
reflective	216	0.518	0.502	0.486	0.530	0.500
cheer up	112	0.544	0.563	0.535	0.536	0.544
depressing	267	0.524	0.597	0.585	0.584	0.554
depressive	126	0.703	0.644	0.620	0.651	0.616
dark	1000	0.582	0.615	0.597	0.603	0.577
depression	127	0.511	0.566	0.578	0.554	0.598
happiness	169	0.524	0.497	0.535	0.529	0.517
heartache	125	0.544	0.536	0.544	0.516	0.508
calming	131	0.505	0.550	0.588	0.531	0.554
wistful	209	0.510	0.493	0.505	0.493	0.486
sunny	156	0.519	0.536	0.516	0.513	0.510
cheerful	150	0.557	0.557	0.560	0.590	0.527
heartbreaking	173	0.552	0.532	0.518	0.523	0.468
rage	115	0.683	0.696	0.635	0.696	0.626
angst	179	0.547	0.565	0.556	0.579	0.541
cool down	172	0.767	0.796	0.770	0.779	0.776

Table 6. Classification results for each mood tag.

7. DISCUSSION

In this paper, we have presented a comprehensive evaluation of music mood classification, relying solely on lyrics as a source of information. We have studied three different granularities for mood representation (quadrants, groups and mood tags) and evaluated four term-weighting schemes (*tf, tf-idf, BM25* and *delta tf-idf*). In particular, we have used a large publicly available dataset in our analysis to enable reproducibility of experiments. This approach contrasts with previous work in this area, where much of the work has relied on small-scale, private datasets, and where contradictory results were reported in some instances.

The results obtained show that lyrics alone can be used for the mood classification task, performing particularly well for some moods (e.g. the v^-a^+ mood quadrant, where classification accuracies up to 70% were reached). However, in contrast to findings reported in [37], in this work no statistically significant differences in classification performance were found when using the various term-weighting schemes considered. These results align with [18], where the use of a smaller subset of term-weighting approaches (*binary*, *tf* and *tf-idf*) evaluated on a different dataset led to similar performance. The *delta tf-idf* term-weighting scheme also did not outperform other approaches in our analysis, which is somewhat surprising given this scheme takes term distribution across classes into account. Moreover, the results presented in [37], where a term-weighting scheme which also considers class distribution of terms is proposed, do not align with our findings, as they show a substantial improvement in classification performance over the *tf* approach.

Given the discrepancies in findings between the various works discussed above, clearly there is a need for a benchmark dataset to assess the performance of lyrics-based classification approaches. While the The *MusixMatch Dataset* and *Million Songs Dataset* represent significant steps in this direction, the analysis presented in Section 5 of this pa-

per highlights some important limitations in them. For example, only counts for the top 5,000 terms per song across the collection are made available, which precludes the application of more sophisticated natural language processing techniques to the classification task. Moreover, the approach used to selected the top 5,000 terms leads to a high degree of common terms across moods; as shown in Table 2, only 365 of the 4,481 terms are unique to one mood quadrant, which severely limits the discriminating power of these terms. In this regard, we conducted a small scale study involving 800 songs (200 for each mood quadrant) for which full lyrics are available. The term statistics in this dataset are very different: in total, there are 9,276 distinct terms (across all quadrants), with between 4,000 and 4,600 distinct terms per class, of which between 1,200 and 1,600 (on average, 32% per class) of these terms are *unique* to each class — which is clearly very different to the very low percentage of unique terms (on average, 2% per class) in the publicly available *MusixMatch* dataset used in this work.

In conclusion, while acknowledging that lyrics are copyrighted material and the legal considerations involved in making (full) song lyrics publicly available, the analysis presented in this paper highlights the need for a new benchmark dataset to progress work in this area. The provision of such a dataset would facilitate a true comparison of the different approaches to music classification, the reproducibility of experiments, and allow the true potential for lyrics-based classification approaches to be established.

8. ACKNOWLEDGEMENTS

This work is supported by Science Foundation Ireland through the Insight Centre for Data Analytics under grant number SFI/12/RC/2289.

9. REFERENCES

[1] A. Schindler and A. Rauber, *Capturing the Temporal Domain in Echonest Features for Improved Classification Effectiveness*. Springer International Publishing, 2012.

[2] M. F. McKinney and J. Breebaart, "Features for Audio and Music Classification." *ISMIR*, vol. 3, pp. 151–158, 2003.

[3] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *Speech and Audio Processing, IEEE transactions*, vol. 10, no. 5, pp. 293–302, 2002.

[4] M. Besson, F. Faita, and I. Peretz, "Singing in the Brain: Independence of Lyrics and Tunes," *Psychological Science*, vol. 9, no. 6, pp. 494–498, 1998.

[5] J. S. Downie, "When Lyrics Outperform Audio for Music Mood Classification: A Feature Analysis," *In Proceedings of the 11th Internation Society for Music Information Retrieval Conference (ISMIR 2010)*, pp. 619–624, 2010.

[6] R. Mayer, R. Neumayer, and A. Rauber, "Combination of Audio and Lyrics Features for Genre Classification in Digital Audio Collections Categories and Subject Descriptors," *In Proceedings of the 16th ACM International Conference on Multimedia*, pp. 159–168, 2008.

[7] C. McKay and I. Fujinaga, "Improving Automatic Music Classification Performance by Extracting Features from Different Types of Data," *In Proceedings of the international Conference on Multimedia Information Retrieval - MIR'10*, pp. 257–266, 2010.

[8] Y. Hu, X. Chen, and D. Yang, "Lyric-based Song Emotion Detection with Affective Lexicon and Fuzzy Clustering Method," *In Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, pp. 123–128, 2009.

[9] R. Mihalcea and C. Strapparava, "Lyrics, Music, and Emotions," *In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 590–599, 2012.

[10] T. Bertin-mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, "The Million Song Dataset," *In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pp. 591–596, 2011.

[11] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multi-Label Classification of Music into Emotions." *ISMIR*, vol. 8, pp. 325–330, 2008.

[12] R. Foucard and S. Essid, "Exploring New Features for Music Classification," *In Proccedings of the 14th International IEEE Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pp. 1–4, 2013.

[13] R. Mayer and A. Rauber, "Musical Genre Classification by Ensembles of Audio and Lyrics Features," *In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pp. 675–680, 2011.

[14] Y. Song, S. Dixon, M. Pearce, and A. Halpern, "Do Online Social Tags Predict Percieved or Induced Emotial Responses to Music?" *In Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, pp. 89–94, 2013.

[15] K. Kosta, Y. Song, G. Fazekas, and M. B. Sandler, "A Study of Cultural Dependence of Perceived Mood in Greek Music," *In Proceedings of the 14th International Society for Music Information Retrieval (ISMIR 2013)*, pp. 317–322, 2013.

[16] J. A. Russell, "A Circumplex Model of Affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[17] X. Hu, "Music and mood: Where Theory and Reality Meet," *In Proceedings of iConference*, pp. 1–8, 2010.

[18] H. Xiao, J. S. Downie, and A. F. Ehmann, "Lyric Text Mining in Music Mood Classification," *American Music*, vol. 183, no. 5040, pp. 411–416, 2009.

[19] M. Bradley and P. Lang, "Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings," The Center for Research in Psychophysiology, University of Florida, Tech. Rep., 1999.

[20] X. Hu, J. S. Downie, C. Laurier, M. Bay, and A. F. Ehmann, "The 2007 MIREX Audio Mood Classification Task: Lessons Learned." *In Proceedings of the 9th International Conference of Music Information Retrieval (ISMIR 2008)*, pp. 462 – 467, 2008.

[21] X. Hu and J. S. Downie, "Improving Mood Classification in Music Digital Libraries by Combining Lyrics and Audio," *In Proceedings of the 10th Annual Joint Conference on Digital Libraries*, pp. 159–168, 2010.

[22] M. Kim and H.-C. Kwon, "Lyrics-Based Emotion Classification Using Feature Selection by Partial Syntactic Analysis," *23rd IEEE International Conference on Tools with Artificial Intelligence*, pp. 960–964, Nov. 2011.

[23] V. Kumar and S. Minz, "Mood Classifiaction of Lyrics using SentiWordNet," *2013 International Conference on Computer Communication and Informatics (ICCCI-2013)*, pp. 1–5, Jan. 2013.

[24] P. S. Dodds and C. M. Danforth, "Measuring the Happiness of Large-Scale Written Expression: Songs, Blogs, and Presidents," *Journal of Happiness Studies*, vol. 11, no. 4, pp. 441–456, Jul. 2009.

[25] J. P. G. Mahedero, A. Martinez, P. Cano, M. Koppenberger, and F. Gouyon, "Natural Language Processing of Lyrics," *In Proceedings of the 13th Annual ACM International Conference on Multimedia - MULTIMEDIA '05*, pp. 475–478, 2005.

[26] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. ACM Press, New York, 1999, vol. 9.

[27] S. Ceri, A. Bozzon, and M. Brambilla, *An Introduction to Information Retrieval*. Springer Berlin Heidelberg, 2013.

[28] K. S. Jones, "A Statistical Interpretation of Term Specificity and its Application in Retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[29] A. Aizawa, "An Information-theoretic Perspective of tf-idf Measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, Jan. 2003.

[30] K. Sparck Jones, S. Walker, and S. Robertson, "A Probabilistic Model of Onformation Retrieval: development and comparative experiments," *Information Processing & Management*, vol. 36, pp. 809–840, 2000.

[31] J. Martineau and T. Finin, "Delta TFIDF: An Improved Feature Space for Sentiment Analysis," *In Proceedings of the Third International ICWSM Conference*, pp. 258–261, 2009.

[32] M. a. Hall, "Correlation-based Feature Selection for Machine Learning," Ph.D. dissertation, 1999.

[33] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: an Update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[34] R. Fan, K. Chang, and C. Hsieh, "LIBLINEAR: A Library for Large Linear Classification," *The Journal of Machine Learning*, vol. 9, pp. 1871–1874, 2008.

[35] Y. Song, S. Dixon, and M. Pearce, "A Survey of Music Recommendation Systems and Future Perspectives," *9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)*, pp. 19–22, 2012.

[36] E. Theodorsson-Norheim, "Kruskal-Wallis test: BASIC computer program to perform nonparametric one-way analysis of variance and multiple comparisons on ranks of several independent samples." *Computer Methods and Programs in Biomedicine*, vol. 23, no. 1, pp. 57–62, 1986.

[37] M. V. Zaanen and P. Kanters, "Automatic Mood Classification Using TF*IDF Based on Lyrics." *In Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, pp. 75–80, 2010.

Analyzing the influence of pitch quantization and note segmentation on singing voice alignment in the context of audio-based Query-by-Humming

Jose J. Valero-Mas
Pattern Recognition and Artificial Intelligence Group, University of Alicante
jjvalero@dlsi.ua.es

Justin Salamon
Music and Audio Research Laboratory, New York University
justin.salamon@nyu.edu

Emilia Gómez
Music Technology Group, Universitat Pompeu Fabra
emilia.gomez@upf.edu

ABSTRACT

Query-by-Humming (QBH) systems base their operation on aligning the melody sung/hummed by a user with a set of candidate melodies retrieved from polyphonic songs. While MIDI-based QBH builds on the premise of existing annotated transcriptions for any candidate song, audio-based research makes use of melody estimation algorithms for the songs. In both cases, a melody abstraction process is required for solving issues commonly found in queries such as key transpositions or tempo deviations. Full automatic music processes are commonly used for this, but due to the reported limitations in state-of-the-art methods for real-world queries, other possibilities should be considered. In this work we explore three different melody representations, ranging from a general time-series one to more musical abstractions, which avoid full automatic transcription, in the context of an audio-based QBH system. Results show that this abstraction process plays a key role in the overall accuracy of the system, obtaining the best scores when temporal segmentation is dynamically performed in terms of pitch change events in the melodic contour.

1. INTRODUCTION

Query-by-Humming systems constitute a particular case of content-based music similarity search schemes in which the input query is a sung, hummed or whistled section of a song, usually its main melody [1, 2], and the output is the target song. Such a music retrieval paradigm stands as an interesting alternative to classic text-based retrieval frameworks (for instance, tag-based search) for its simple usage complemented by the fact that no musical knowledge from the user is required [3].

Research in QBH mainly focuses on addressing the inaccuracies found when producing the queries: on the one hand, *tuning issues* have to be considered as users may sing out of tune and/or in a different key [4]; on the other hand, *tempo deviations* among queries and candidates may also occur [4, 5]. For overcoming them, a *melody abstraction* process, which may range from general time-series codifications to more sophisticated music-based ones, followed by a *melody comparison* stage are performed for estimating the dissimilarity between the query and the candidates [6].

The process for obtaining the set of candidate melodies is not trivial [2, 5, 7]: main fundamental frequency (f0) estimation for queries and candidates cannot be assumed as an accurate process, especially when dealing with polyphonic songs [8]. While this estimation process is inevitable for the queries as they constitute the user audio input to the system, this issue has been typically avoided for the candidate songs by assuming the existence and availability of high-level annotated representations (for instance, MIDI files) of these melodies.

Due to the limitations the previous assumption implies, mostly in terms of practical systems, some QBH schemes try to estimate this melody algorithmically from audio. Although more realistic, this adds more complexity to the system since no melody estimation algorithm is error-free.

As aforementioned, melodic contours require of an abstraction process. For taking advantage of the large amount of research carried in the symbolic melodic similarity field, melodies estimated from audio sources are coded into high-level music representations [9], usually with full automatic music transcription systems. However, given the limitations current state-of-the-art transcription algorithms exhibit [10], it seems interesting to study alternative abstractions to such high-level representations.

In this paper we present a study of the influence of different *melody abstraction* processes which avoid the complexity of full automatic music transcription in the context of QBH. Particularly, we assess the influence of pitch quantization and note segmentation in singing voice alignment for QBH. For that, we take as starting point the scheme in Figure 1 and we evaluate three different melodic contour representations: the first one makes use of the time-series encoding algorithm Symbolic Aggregate Approximation (SAX) [11], which is based on a fixed-duration temporal segmentation and statistical encoding; the second one modifies the original SAX algorithm so that the encoding is performed using a semitone-band representation; finally, as a third method we propose to segment the melody using the pitch change events in the melodic contour.

To ensure the scalability of the system we use the melody estimation algorithm MELODIA [12]. This method estimates the predominant pitch from both monophonic and

Copyright: ©2015 Jose J. Valero-Mas 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.

polyphonic music signals. In terms of the contour comparison, we apply two sequence alignment algorithms: Smith-Waterman [13], originally meant for DNA sequences but with large application in the time series field, and Subsequence Dynamic Time Warping [14].

The rest of the paper is structured as it follows: Section 2 briefly reviews similar research proposals; Section 3 and Section 4 present the melody extraction algorithm MELODIA and the local alignment algorithms considered respectively; Section 5 introduces the assessed contour representations; Section 6 presents the evaluation methodology; Section 7 presents and discusses the results obtained; finally, Section 8 outlines the conclusions obtained and proposes possible future work.

2. RELATED WORK

One of the first proposed QBH systems was the one by Ghias et al. [15] in which queries were transcribed using autocorrelation for pitch tracking, the candidate elements were MIDI files and the search was performed using a fuzzy string matching algorithm. Although many similar systems based on some kind of full automatic music transcription have been proposed since then, the work by Dannenberg et al. [3] with the MUSART Testbed, a framework for the assessment of this type of QBH systems, stands as a relevant example.

In terms of systems not based on full automatic music transcription, a relevant example is the one by Duda et al. [1] in which a series of audio descriptors (Mel-Frequency Cepstrum Coefficients, Power, Fundamental frequency contour, Voice Formants and Chroma) are extracted from the audio files and are then encoded using SAX [11]; similarity is performed using Edit distance [17].

Another example can be found in the system by Ito et al. [5]. In this case, instead of obtaining a single melodic contour for the candidate elements, multiple fundamental frequency candidates are retrieved, using a variation of the PreFEst algorithm [18], for comparison to the query contour using a basic scoring function. Salamon et al. [2] proposed a system in which melodies are quantized into semitones and mapped into one octave. Similarity is performed using the Q_{max} algorithm [19].

In terms of the automatic extraction of melodies, some explored techniques use fundamental frequency extraction algorithms [5, 16], main singing voice extraction [1, 7] or the use of predominant melody estimation algorithms [2].

All approaches are summarized in Table 1.

3. MELODY ESTIMATION

Melodies from both queries and candidate songs are obtained using the predominant melody estimation algorithm MELODIA [12] [1] . For a given music piece, the algorithm estimates the fundamental frequency of the predominant melodic line in the song. This particular algorithm outperformed all other state-of-the-art methods in the 2011 Music Information Retrieval Evaluation eXchange (MIREX) campaign [2] in the *Audio Melody Extraction* task.

In a more detailed analysis, results in [12] report its robustness in terms of octave errors (properly tracking pitch values in the correct octave) and voiced frame detection (frames belonging to the predominant melody). However, it must be also pointed out that the algorithm tends to confuse unvoiced elements as voiced, thus lowering the overall performance.

Finally, we provide a brief explanation to the four stages MELODIA comprises: an initial *Sinusoid extraction* step estimates the predominant frequency values at each instant in the signal; then, a *Salience function* based on a harmonic series is derived; after that, a series of *Pitch contours* are created using a set of rules based on Auditory Scene Analysis (ASA) for finally selecting the predominant melody in the *Melody selection* stage. In this experimentation, MELODIA has been configured to its default analysis rate ($\Delta t_{\text{MEL}} = 2.9$ ms).

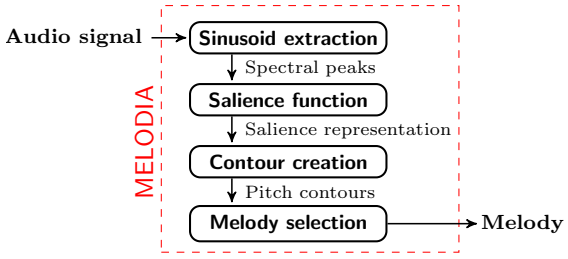


Figure 2. Block diagram of the MELODIA algorithm.

4. MELODY ALIGNMENT

In this work, similarity between the query and the candidate melodies is estimated by means of sequence alignment methods. This premise suits the QBH task as queries may contain tempo deviations with respect to the corresponding melodies of the actual song to be retrieved. The two algorithms considered are now introduced.

4.1 Smith-Waterman

The Smith-Waterman (SW) method [13] is an alignment algorithm formerly proposed for DNA sequences. This algorithm performs a search for the most similar regions between a pair of sequences, coded as strings, in a time-warped scenario. Smith-Waterman requires a series of costs to be defined: a reward for symbol matches (C_{MATCH}), a penalty for mismatches (C_{MISMATCH}) and two costs for time warps ($C_{\text{INSERTION}}$ and C_{DELETION}). Table 2 shows the different configurations considered.

4.2 Subsequence Dynamic Time Warping

Subsequence Dynamic Time Warping (S-DTW) constitutes a modification on Dynamic Time Warping (DTW) proposed by Müller in [14]. While DTW forces a global alignment between two sequences, S-DTW eliminates that restriction for allowing local matches between the sequences. The modification makes it suitable for query-by-example

[1] http://mtg.upf.edu/technologies/melodia

[2] http://www.music-ir.org/mirex/wiki/MIREX_HOME

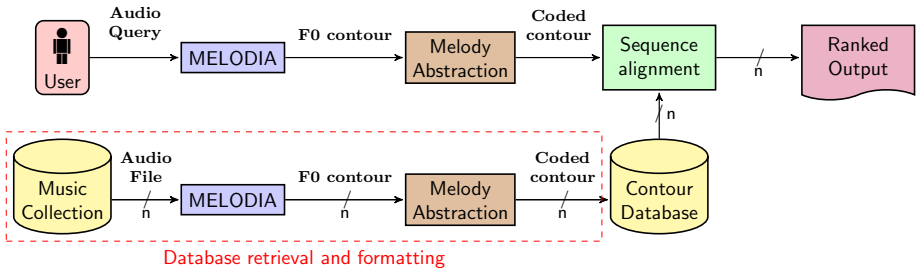


Figure 1. Scheme of the QBH system proposed. Main melodies are estimated from the audio files (query and candidate songs) using the melody estimation algorithm MELODIA, being then encoded using a certain contour representation; local alignment between the query and each element in the database is then performed and the results are eventually ranked.

First Author	Feature(s)	Feature extraction		Abstraction	Similarity
		Query	Music collection		
Ghias [15]	Main F0 contour	Pitch tracking (autocorrelation)	MIDI files	Strings representing changes in contour: U (up), D (down) and S (same)	Fuzzy string matching
Dannenberg [3]	Main F0 contour	Pitch tracking (autocorrelation)	MIDI files	IOI + Relative pitch, Fixed-Time Segmentation + Relative pitch	Note Interval, N-gram, Contour Matching, HMM Matching, CubyHum Matcher
Duda [1]	MFCC, Audio Power, F0, Voice Formants, Chroma + derivatives (1st and 2nd order)	No extraction	Stereo pan removal to retrieve lead singing voice	SAX coefficients	Edit distance
Jeon [16]	Main F0 contour	Constant-Q Transform + heuristics	Constant-Q Transform + heuristics	Wavelet coefficients	Coefficient's comparison
Ito [5]	Multiple F0 contours	PreFEst variation	PreFEst variation	Tempo normalization + logarithm of frequencies values	Scoring function (absorbs key differences)
Salamon [2]	Main F0 contour	MELODIA	MELODIA	Semitone-band based chromagrams with fixed-time segmentation	Q_{max}
Rocamora [7]	Lead singing voice	YIN + energy-based segmentation and extraction	Singing voice detection and (+ query process)	Pitch and duration ratios (relative encoding)	Edit distance

Table 1. Summary of related QBH approaches.

	C_{MATCH}	$C_{MISMATCH}$	$C_{INSERTION}$	$C_{DELETION}$
T1	1	-0.5	-0.5	-0.5
T2	1	-1	-0.5	-0.5
T3	1	-1	-1	-1
T4	1	-0.5	-1	-1

Table 2. Weights of the four tested configurations for the Smith-Waterman alignment algorithm.

applications [20] as queries usually constitute an excerpt of the element to be retrieved. The cost function used in this paper has been the Edit distance (ED) [17].

5. MELODY ABSTRACTIONS

We now describe the three considered melody abstractions for encoding the estimated melodic pitch contours.

5.1 Symbolic Aggregate Approximation (SAX)

SAX, introduced by Lin et al. [11] in 2007, is a symbolic representation for time series (sequences encoded as strings) able to cope with two major drawbacks usually found in other methods: the need for both a *dimensionality reduction* and a *lower bound* in the distance computations. Although reported as a fast and competitive algorithm for similarity search, SAX has not been widely used in Music Information Retrieval (MIR). Some of the few examples in

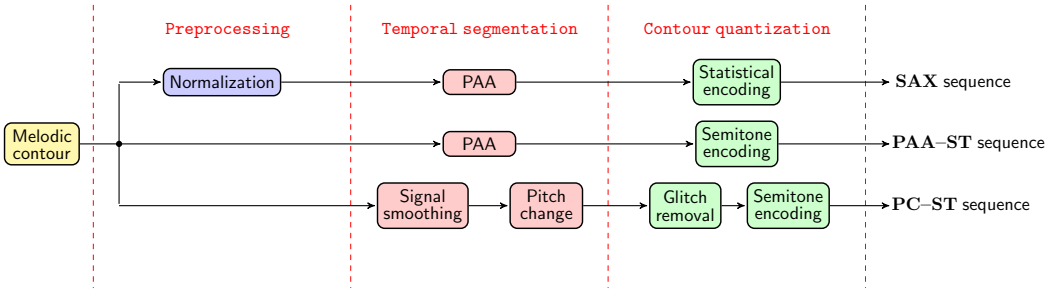


Figure 3. Diagram depicting the different stages the three proposed abstractions comprise.

this field can be found in the study of guitar articulations [21], Beijing opera singing similarity [22] or in QBH [1].

SAX comprises three steps for coding any sequence:

5.1.1 Time-series normalization

Given a time series $C = \{c_1, c_2, ..., c_n\}$ of length **n**, this abstraction performs an initial normalization process:

$$c'_i = \frac{c_i - \mu}{\sigma} \qquad 1 \leq i \leq n \tag{1}$$

were c_i represents each element of the initial time series (the f0 contour in cents [3] retrieved by MELODIA) and μ and σ the mean and the standard deviation respectively.

5.1.2 Piecewise Aggregate Approximation (PAA)

This second stage takes the normalized time series C' of length **n** and maps it in an **M**-dimensional (modifiable parameter) vector $\bar{C} = \{\bar{c}_1, \bar{c}_2, ..., \bar{c}_M\}$ of equally-sized segments:

$$\bar{c}_i = \frac{M}{n} \cdot \sum_{j=\lfloor \frac{n}{M} \rfloor (i-1)+1}^{\lfloor \frac{n}{M} \rfloor i} c'_j \qquad 1 \leq i \leq M \tag{2}$$

Given the different length of the f0 sequences to encode, fixing a global **M** value would produce each segment to represent a different temporal duration in each sequence. Instead, we fix a frame temporal duration τ_t for all sequences. Since each c'_i represents Δt_{MEL}, the frame size in samples can be obtained as $\tau_s = \tau_t / \Delta t_{\text{MEL}}$. Thus, **M** is given by $M = n/\tau_s$. As an initial experiment, τ_t values considered are 0.3, 0.5, 0.8, 1 and 2 seconds.

5.1.3 Symbolic representation

The last stage maps $\bar{C}$ to a series of **a** (adjustable parameter) discrete symbols. To assure equiprobability of appearance for all symbols, **a** regions are defined based on a statistical distribution, typically Gaussian [11]. The group of breakpoints $B = (\beta_1, \beta_2, ..., \beta_{a-1})$ for delimiting such regions accomplish that the area under a $\mathcal{N}(0,1)$ Gaussian curve from β_j to β_{j+1} equals $1/\mathbf{a}$. In addition, $\beta_0 = -\infty$ and $\beta_a = +\infty$.

[3] The reference frequency is 55 Hz as it represents the minimum frequency value retrieved by MELODIA.

Each interval $[\beta_{j-1}, \beta_j)$ represents a certain symbol α_j. Therefore, **M**-length vector $\bar{C} = \{\bar{c}_1, \bar{c}_2, ..., \bar{c}_M\}$ is mapped into the **M**-length vector $\hat{C} = (\hat{c}_1, \hat{c}_2, ..., \hat{c}_M)$:

$$\hat{c}_i = \alpha_j \quad \text{if} \quad \bar{c}_i \in [\beta_{j-1}, \beta_j) \qquad \begin{matrix} 1 \leq i \leq M \\ 1 \leq j \leq a \end{matrix} \tag{3}$$

As an exploratory study, the **a** tested values have been 3, 4, 6, 8, 12, 16 and 20.

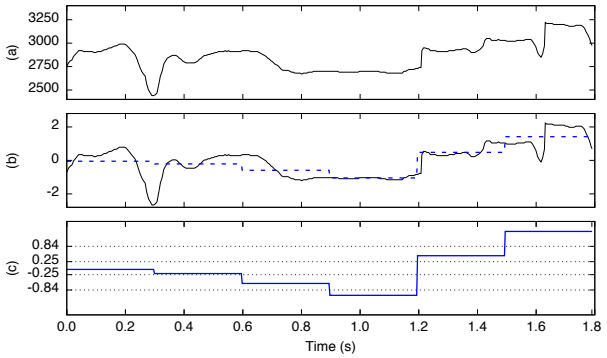


Figure 4. Example of the SAX abstraction process with $\mathbf{a} = 5$ and $\tau_s = 0.3$ *s*: (a) Initial time series in cents; (b) Normalized time series (solid) and PAA codification (dashed); (c) PAA codification (solid) and SAX encoding breakpoints (dotted).

5.2 PAA temporal segmentation with semitone quantization (PAA–ST)

The first proposed SAX modification revises the *Symbolic representation* stage: instead of using a statistical distribution approach for the vertical quantization, a fixed grid with semitone divisions is established. The minimum considered frequency value is 55 Hz given it is the minimum f0 retrieved by MELODIA. The normalization stage is omitted as it modifies the pitch range. Folding the contour to a single octave as in [2] was discarded as preliminary non-exhaustive experimentation did not report improvements.

Finally, relative pitch encoding is applied (storing intervals between segments) to provide transposition invariance. In this abstraction, the assessed time durations for the PAA segments have been the same as in the SAX abstraction.

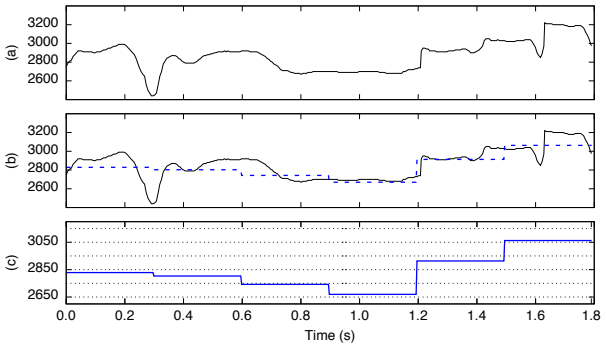


Figure 5. Example of the PAA–ST abstraction process with $\tau_s = 0.3\ s$: (a) Initial time series in cents (solid); (b) Initial time series in cents (solid) and PAA codification (dashed); (c) PAA codification (solid) and semitone grid breakpoints (dotted).

5.3 Pitch change segmentation with semitone quantization (PC–ST)

This second modification builds on the previous one but avoids PAA and dynamically segments the melodic contour when there is a pitch change event. Vertical quantization using a semitone grid is maintained. In order to avoid *false* segments due to artifacts and fast pitch changes the pitch contour may contain, a softening process is applied.

The softening process comprises two steps: (a) an initial *signal smoothing* using an average filter of τ_{SM} duration with sliding window (applied before the semitone quantization process) and (b) a *glitch removal* step by applying a median filter of τ_{GR} with sliding window for removing segments shorter than a certain duration (applied after the semitone quantization step).

We have studied four different filter durations: 25, 50, 75 and 100 pitch samples. Given the MELODIA analysis rate, these values correspond to filter durations τ_{SM} and τ_{GR} of 70, 140, 218 and 290 milliseconds respectively.

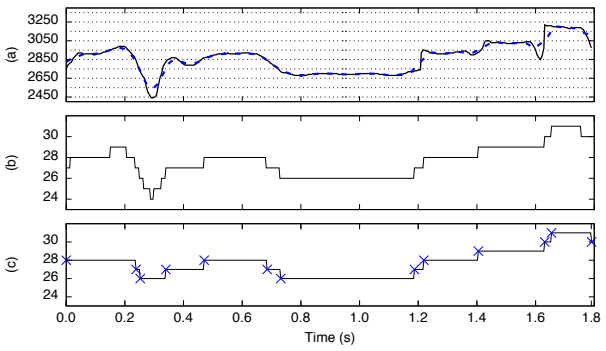


Figure 6. Example of the PC–ST abstraction process with $\tau_{SM} = 70\ ms$ and $\tau_{GR} = 140\ ms$: (a) Initial time series in cents (solid), smoothed contour after the first filter (dashed) and semitone grid (dotted); (b) absolute semitone encoding; (c) absolute semitone encoding after the second filter, the cross symbol ($\times$) points out each new temporal segment.

6. EVALUATION METHODOLOGY

6.1 Dataset

The evaluation data is the same as in [2] and it comprises a query corpus and a music collection.

The music collection, or candidate songs, contains 2125 commercial songs [19] distributed in 523 groups (each one being a group of covers of the same song). Song lengths range from 0.5 to 8 minutes with an average duration of 3.6 minutes. Following the evaluation strategy in [2], the collection is divided into two subsets: a first one containing only canonical songs [4] from the corpus (481 elements) and a second one comprising the entire music collection (2125 elements).

The freely-available query corpus set [5] comprises a total of 118 queries recorded by 17 users (9 female and 8 male) whose musical knowledge ranged from none to amateur musician, with an average of 6.8 queries per user (1 as a minimum and 11 as a maximum). As reference songs, users chose among the 481 canonical subset of the music collection. Queries range from 11 to 98 seconds, with an average length of 28.6 seconds.

6.2 Measures

Generally, a QBH system is assessed using rank metrics as its output is a sorted list of the similarity scores between the query and each candidate melody. In these terms, the two most common evaluation measures are the Mean Reciprocal Rank (MRR) and the Top-X Hit Rate.

6.2.1 Mean Reciprocal Rank (MRR)

For a given user query **Q**, corresponding to a target song **A**, the system returns sorted list in which song **A** is located at position (or rank) **r**. The Reciprocal Rank (RR) for **A** is defined as 1/**r**. Generalizing for a series of **n** queries, the Mean Reciprocal Rank (MRR) is defined as:

$$\mathrm{MRR} = \frac{1}{\mathrm{N}} \cdot \sum_{i=1}^{\mathrm{N}} \frac{1}{\mathrm{r}\left(\mathrm{Q}_i\right)} \tag{4}$$

Scores obtained fall in the range $0 \leq \mathrm{MRR} \leq 1$, where 0 stands for the worst case and 1 for the best.

For any of the evaluation sets considered, **r** is assumed to be highest-ranked version matching query **Q**.

6.2.2 Top-X Hit Rate

Given the resulting rank, this measure checks whether the position **r** of the matching element of **Q** is among the first **X** positions of the list, *i.e.* $\mathbf{r}\left(\mathbf{Q}_i\right) \leq \mathbf{X}$. This estimates the frequency of retrieving the correct result among the first **X** positions [2].

As in the previous case, the highest-ranked version which matches query **Q** is considered as **r**.

[4] The songs as published by the band who composed/played it.
[5] http://mtg.upf.edu/download/datasets/MTG-QBH.

7. RESULTS AND DISCUSSION

7.1 Results

Results obtained for the abstractions and alignment algorithms considered are presented in Table 3. Due to space requirements, only best result obtained for each configuration is reported. In order to consistently assess these results, a baseline configuration has been added: for each query, the candidates' rank is randomly sorted (without performing any similarity measure) and the evaluation figures are then obtained; the results shown for this configuration constitute the average of 10,000 repetitions. Results from [2] are also included for a comparative assessment.

We note that all the proposed configurations significantly outperform the MRR figure of 0.014 obtained with the considered baseline. However, the results are still considerably lower than the ones obtained in [2]. Nevertheless, the differences in performance among the different configurations allow us to make some interesting observations.

We see that the combination of SAX with the Smith-Waterman alignment obtains an MRR of 0.05 when evaluated against the canonical (481 songs) test set. The semitone quantization step, which constitutes the only difference with the SAX abstraction process, does not significantly affect the results with respect to the SAX ones (MRR score is now around 0.04). This is a point to be remarked since, although the abstraction is more related to an actual music representation, the accuracy scores obtained are similar to the ones obtained with SAX.

PC–ST assesses the influence of note segmentation in the process. Focusing on the canonical set and the Smith-Waterman alignment, this particular encoding methodology achieves an MRR score around 0.09, thus outperforming the two other abstractions. This suggests that musically-informed temporal segmentation of pitch sequences may benefit the performance of the system.

As expected from [2], the inclusion of cover songs among the candidates set enhances retrieval accuracy for our configurations, except for the PAA–ST: while for both SAX and the PC–ST there is an improvement of 0.05 in the MRR measure, results in the PAA–ST do not significantly vary in comparison with the canonical set.

Results obtained for the Top-X Hit Rate measure also support our observation that a proper temporal segmentation in the process is beneficial for the system. When only considering the canonical set, the correct candidate is retrieved on the first position around 3 % and 1 % of the time for the SAX and the PAA–ST respectively while, when considering the PC–ST, this figure goes close to 6 %. This same conclusion can be observed with the rest of the Hit Rates (3, 5 and 10) as well as with the inclusion of covers among the candidates.

Focusing on the alignment algorithms, although the different proposed Smith-Waterman configurations show some influence on the overall accuracy, there is no clear outperforming configuration for all the cases. Results obtained with Subsequence Dynamic Time Warping show lower performance than the other considered alignment algorithm. This may be improved with the use of more complex cost functions rather than the considered Edit distance.

7.2 Discussion

While the proposed SAX abstraction has been shown to perform successfully for a variety of time-series tasks [11], results in the experiments proposed suggest that this is not the case for musical time-series data in the context of QBH. The most likely reason for this to happen is the fact that SAX does not consider any particularities the origin domain of the time series may have. Thus, in the case of QBH, SAX may be abstracting away musically-related information from the melodic contours required for properly performing the alignment. This idea is further supported by the improvement in the results when using the PC–ST abstraction as, although in a very naïve way, it tries to segment the different musical notes present in the contour.

The results obtained in the two modifications proposed support the relevance of using musically-informed temporal segmentation of the contour. In this study, the use of a basic temporal segmentation based on pitch change events leads to accuracy improvements when compared to the use of the PAA dimensionality reduction algorithm. The most likely reason for this is again the fact that the use of the PAA algorithm does not take into account the musical nature of the data to encode, thus abstracting away relevant information necessary for the alignment. In these terms, the use of more sophisticated temporal segmentation techniques for music data, as for instance onset detection, could improve these results.

Although the abstractions studied in this paper are not competitive in terms of a practical QBH system, evidence from previous work (cf. [2]) shows non-transcription abstractions may lead to successful results. These results encourage the exploration of other abstractions to provide competent alternatives to transcription-based QBH systems.

8. CONCLUSIONS

Query-by-Humming (QBH) systems constitute a particular type of music search engine in which the query is a sung or hummed excerpt of the main melody of a song. Most often, these schemes rely on both existing music annotations and fully-automated music transcription algorithms for performing the melodic similarity. Although many examples of QBH systems have been proposed under this premise, its limited scalability together with the fact that no full automatic transcription algorithm is error-free clearly limits their performance in practical situations.

In this work we assessed the influence of this particular step in such systems by using of three melody encoding alternatives which avoid full music transcription. More precisely, starting from the general time-series encoding method Symbolic Aggregate Approximation (SAX), we modify this algorithm by incorporating music-based pitch quantization and segmentation for evaluating their influence in the context of a QBH system. Results obtained suggest that the time-series representation algorithm SAX does not seem to be suitable for melody alignment in the context of Query by Humming. In this sense, the main out-

Approach	Evaluation subset	Alignment algorithm	Algorithm configuration	MRR	Top-X Hit Rate (%)			
					1	3	5	10
SAX	Canonical	SW	T1	0.0500	2.54	5.93	7.63	9.32
			T2	0.0566	2.54	5.93	5.93	11.02
			T3	0.0632	4.24	5.93	5.93	9.32
			T4	0.0472	3.39	4.24	5.08	6.78
		S-DTW	ED	0.0333	1.69	3.39	3.39	8.47
	Complete	SW	T1	0.1117	7.63	11.86	12.71	17.80
			T2	0.1155	7.63	11.86	12.71	17.80
			T3	0.0962	5.08	10.17	11.86	14.41
			T4	0.0849	5.08	8.47	11.02	12.71
		S-DTW	ED	0.0443	2.54	4.24	5.08	8.47
PAA–ST	Canonical	SW	T1	0.0515	2.54	4.24	6.78	11.02
			T2	0.0421	1.69	3.39	4.24	9.32
			T3	0.0391	1.69	2.54	4.24	6.78
			T4	0.0424	1.69	4.24	4.24	5.93
		S-DTW	ED	0.0346	1.69	2.54	3.39	5.93
	Complete	SW	T1	0.0396	1.69	2.54	5.93	9.32
			T2	0.0424	1.69	3.39	4.24	8.47
			T3	0.0406	1.69	3.39	5.08	8.47
			T4	0.0558	3.39	5.08	5.93	9.32
		S-DTW	ED	0.0334	1.69	2.54	6.78	9.32
PC–ST	Canonical	SW	T1	0.0894	5.93	9.32	10.17	12.71
			T2	0.0967	6.78	11.86	12.71	15.25
			T3	0.0957	6.78	8.47	12.71	14.41
			T4	0.0772	5.08	6.78	8.47	12.71
		S-DTW	ED	0.0165	0.00	0.85	1.69	4.24
	Complete	SW	T1	0.1447	10.17	14.41	17.80	24.58
			T2	0.1460	10.17	16.95	19.49	22.88
			T3	0.1563	11.02	16.95	17.80	22.88
			T4	0.1447	10.17	14.41	17.80	24.58
		S-DTW	ED	0.0181	0.00	0.85	0.85	3.39
Baseline	Canonical	Random		0.0140	0.21	0.62	1.03	2.06
	Complete	Random		0.0039	0.05	0.15	0.25	0.50
Salamon [2]	Canonical	Q_{max}		0.45	40.68	47.46	49.15	51.69
	Complete	Q_{max}		0.56	50.85	58.47	61.02	66.10

Table 3. MRR and Top-X Hit Rate results obtained for the proposed experimentation. Figures represent the best score achieved in each particular abstraction configuration.

come of this study is that, given the complexity of Query by Humming, musically-related abstractions should be considered for encoding the contours.

Future work will consider the incorporation of the conclusions obtained in this work to the abstraction proposed in [2]: as the abstraction in the cited work performs a chromagram representation with a fixed-time temporal segmentation, the incorporation of dynamically-based segmentation could improve the results obtained. Moreover, given the relevance of the user in this particular task, interactive pattern recognition paradigms for addressing the similarity step could be considered: when a query is incorrectly answered, the system could modify the dissimilarity measure (metric learning) to incorporate the user's feedback.

Acknowledgments

This research work has been partially supported by Consejería de Educación de la Comunitat Valenciana through project PROMETEO/2012/017, Vicerrectorado de Investigación, Desarrollo e Innovación de la Universidad de Alicante through FPU programme (UAFPU2014–5883), the Spanish Ministerio de Economía y Competitividad through project TIMuL (No. TIN2013–48152–C2–1–R, supported by EU FEDER funds) and the Spanish entity Fundació

Obra Social 'laCaixa'. Authors would also like to thank José M. Iñesta for kindly proofreading this paper.

9. REFERENCES

[1] A. Duda, A. Nürnberger, and S. Stober, "Towards Query by Singing/Humming on Audio Databases," in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, Austria, 2007, pp. 331–334.

[2] J. Salamon, J. Serrà, and E. Gómez, "Tonal Representations for Music Retrieval: From Version Identification to Query-by-Humming," *International Journal of Multimedia Information Retrieval, special issue on Hybrid Music Information Retrieval*, vol. 2, no. 1, pp. 45–58, 2013.

[3] R. B. Dannenberg, W. P. Birmingham, B. Pardo, N. Hu, C. Meek, and G. Tzanetakis, "A Comparative Evaluation of Search Techniques for Query-by-humming Using the MUSART Testbed," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 5, pp. 687–701, 2007.

[4] D. Little, D. Raffensperger, and B. Pardo, "A Query by Humming System that Learns from Experience," in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, Austria, 2007, pp. 335–338.

[5] A. Ito, Y. Kosugi, S. Makino, and M. Ito, "A query-by-humming music information retrieval from audio signals based on multiple F0 candidates," in *Proceedings of the International Conference on Audio Language and Image Processing (ICALIP)*, China, 2010, pp. 1–5.

[6] M. Ryynänen and A. Klapuri, "Query by humming of midi and audio using locality sensitive hashing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, USA, 2008, pp. 2249–2252.

[7] M. Rocamora, P. Cancela, and A. Pardo, "Query by humming: Automatically building the database from music recordings," *Pattern Recognition Letters*, vol. 36, no. 1, pp. 272–280, 2014.

[8] J. Salamon, E. Gómez, D. P. Ellis, and G. Richard, "Melody Extraction from Polyphonic Music Signals: Approaches, applications, and challenges," *IEEE Signal Processing Magazine*, vol. 31, no. 2, pp. 118–134, 2014.

[9] R. Typke, "Music retrieval based on melodic similarity," Ph.D. dissertation, Utrecht University, Netherlands, February 2007.

[10] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions." *J. Intell. Inf. Syst.*, vol. 41, no. 3, pp. 407–434, 2013.

[11] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: A Novel Symbolic Representation of Time Series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[12] J. Salamon and E. Gómez, "Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 6, pp. 1759–1770, 2012.

[13] T. Smith and M. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195–197, 1981.

[14] M. Müller, *Information retrieval for music and motion*. Springer, 2007.

[15] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query by Humming: Musical Information Retrieval in an Audio Database," in *Proceedings of the 3rd ACM International Conference on Multimedia*, USA, 1995, pp. 213–236.

[16] W. Jeon, C. Ma, and Y. M. Chen, "An Efficient Signal-Matching Approach to Melody Indexing and Search Using Continuous Pitch Contours and Wavelets," in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, Japan, 2009, pp. 681–686.

[17] R. A. Wagner and M. J. Fischer, "The String-to-String Correction Problem," *Journal of the Association for Computing Machinery*, vol. 21, no. 1, pp. 168–173, 1974.

[18] M. Goto, "A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.

[19] J. Serrà, H. Kantz, X. Serra, and R. G. Andrzejak, "Predictability of Music Descriptor Time Series and its Application to Cover Song Detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 514–525, 2012.

[20] X. Anguera and M. Ferrarons, "Memory efficient subsequence DTW for Query-by-Example Spoken Term Detection," in *IEEE International Conference on Multimedia and Expo (ICME)*, USA, 2013, pp. 1–6.

[21] T. H. Özaslan and J. L. Arcos, "Legato and Glissando identification in Classical Guitar," in *Proceedings of the 7th Sound and Music Computing Conference (SMC)*, Spain, 2010, pp. 457–463.

[22] S. Zhang, R. C. Repetto, and X. Serra, "Study of the Similarity Between Linguistic Tones and Melodic Pitch Contours in Beijing Opera Singing," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taiwan, 2014, pp. 343–348.

TRAP: TRAnsient Presence detection exploiting Continuous Brightness Estimation (CoBE)

G. Presti[1], D.A. Mauro[2], and G. Haus[1]
[1]Laboratorio di Informatica Musicale (LIM), Dipartimento di Informatica (DI), Universitá degli Studi di Milano
Via Comelico 39, 20135 Milan, Italy
giorgio.presti@unimi.it
[2]Iuav University of Venice, Department of Architercture and Arts
Dorsoduro 2196, 30123 Venice, Italy
dmauro@iuav.it

ABSTRACT

A descriptor of features' modulation, useful in classification tasks and real time analysis, is proposed. This descriptor is computed in the time domain, ensuring fast computation speed and optimal temporal resolution.

In this work we take into account amplitude envelope as inspected feature, so the outcome of this process can be useful to gain information about the input' energy modulation and can be exploited to detect transients presence in audio segments.

The proposed algorithm relays on an adaptation of *Continuous Brightness Estimation* (CoBE).

1. INTRODUCTION

In the context of Music Information Retrieval (MIR) a naive approach for tracking the amount and nature of modulations can be achieved measuring the standard deviation inside a window of the underlying modulated feature, but this method can only quantify the amount of the modulation, with no information about the shape or frequency. For example, the envelope of a rhythmic pattern may have the same standard deviation of a sustained signal with lot of amplitude modulation, while the pitch of an arpeggio may have the same standard deviation of the pitch of a frequency modulated tone.

A possible alternative to this tracking technique might be to estimate the high frequency content of the time series of the feature under consideration. In such a way the outcome is a measure that only depends on the shape, frequency, and amount of the modulation (i.e. how *rich*, *crispy*, or *jagged* is the feature).

Techniques which can promptly respond to this needs providing good approximations with fast computing time are welcome in real-time applications or when analysing very large datasets. This approach is then motivated by its implementation in the temporal domain, low computational cost and parametrizable temporal resolution.

Copyright: ©2015 G. Presti[1], D.A. Mauro[2], and G. Haus[1] 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.

In this paper we present a case study where we approach this issues using CoBE [1] as main algorithm to measure the presence of transients in audio segments. The rationale for this technique came from trying to automatically classify sonification examples (to appear in [2]), where a feature useful to distinguish between continuous sounds and discrete events can be exploited.

2. THE COBE BEHAVIOUR

CoBE can be interpreted as the ratio of high frequencies in a signal. It is computed comparing the energy of a filtered version of the input with the original one. This approach matches in some way the definitions of Brightness given by [3], [4], [5] and [6] but instead of being computed in frequency domain, it is computed in the time domain, enabling some interesting properties, besides performance improvements. For example, exploiting the inverse transfer function of the magnitude of the filter used, it is possible to infer the frequency of a sine wave having the same CoBE value of the input signal, namely the *Equivalent Brightness Frequency* (EBF), shown in Eqn. 1 (for further details see [1]).

$$f = (\frac{f_s}{\pi}) \arcsin(\frac{B}{2}) \tag{1}$$

Where f_s is the sampling frequency and B is the CoBE Brightness value.

2.1 Implementation

With respect to the implementation previously described in [1], a slightly different implementation is proposed here and detailed in Fig. 1. It presents some advantages in terms of stability and it is more readable while preserving the same output. The source code written in Matlab language is the following:

```
function [B,EBF] = CoBE(X,fs,EnvFun,
    varargin)
        % Amplitude envelope E
        E   = EnvFun(X, varargin{:});
        % Filtered version dX
        dX  = diff( [0; X] );
        % dX amplitude envelope Ed
        Ed  = EnvFun(dX, varargin{:});
```

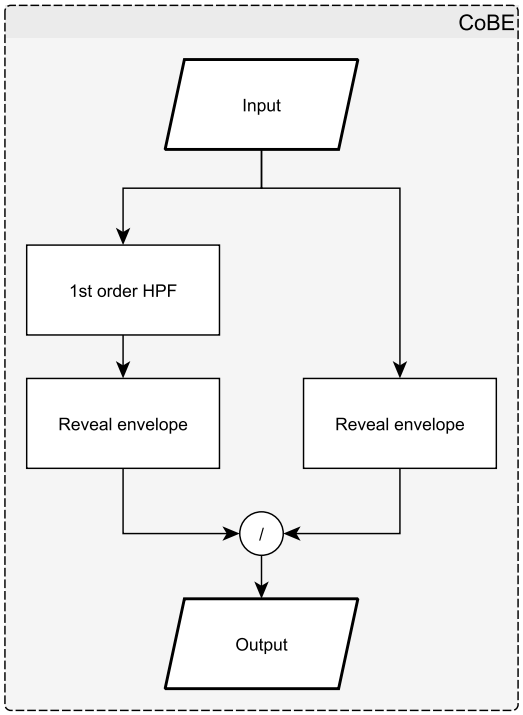


Figure 1. Diagram of the CoBE algorithm, intended as the ratio of high frequencies that constitutes the signal.

```
        % Brightness as Ed / E
        B    = Ed./E;
        % Equivalent Brightness Freq.
    EBF = (fs/pi).*asin(B./2);
end;
```

In contrast to the *High Frequency Content* feature described in [7], the behaviour of CoBE is independent from the signal level, and for monophonic sine waves it is also independent from signal filtering [1]. However, as can be clearly pointed out from both Fig. 1 and source code, it strongly depends on the envelope follower algorithm.

2.2 Behaviour with different envelope followers

Four different envelope followers has been analysed:

- VU-meter style follower, with zero attack and slow release (*Vu*);
- RMS of a moving window, lowpass filtered in order to remove residual ripples (*RMS*);
- Local maxima inside a moving window (*Max*);
- Classic Rectify and Filter approach (*RF*) [2].

As can be seen in Fig. 2, the very-low frequency band may cause issues with the first three algorithms, while high frequencies may be tracked incorrectly by *Vu* and, less significantly, by *Max* and *RF*. In Fig. 3 it is possible to notice

[1] In such a case filtering can be considered as a simple *delay and scale* function

[2] Used in MIRToolbox and implemented according to [8].

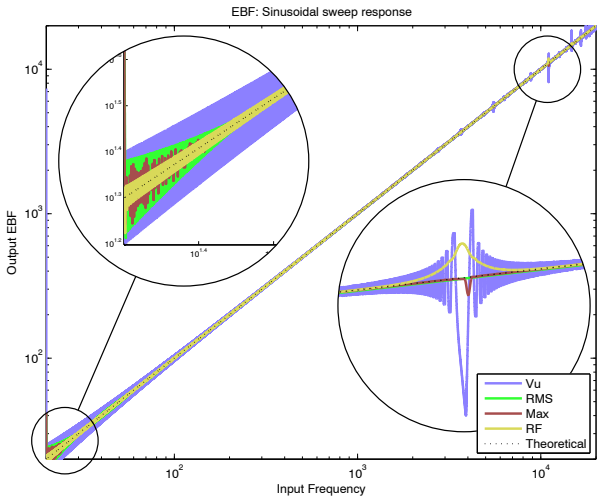


Figure 2. EBF Sweep response using different envelope followers. Differences in the top and bottom ends of the spectrum are magnified.

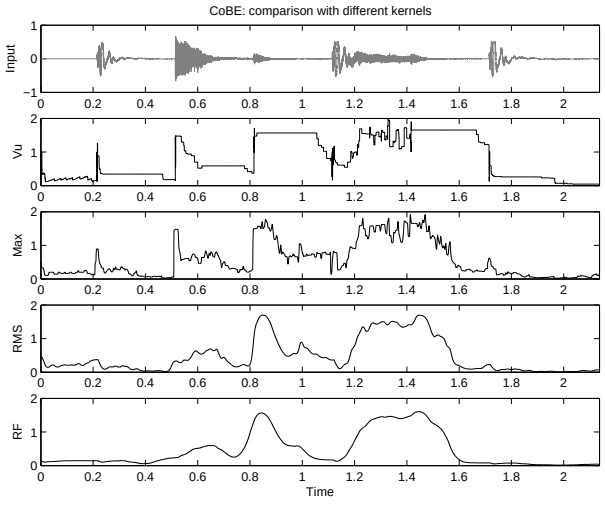


Figure 3. CoBE of a drum sample estimated with different envelope followers.

that the different followers are characterized by an increasing level of smoothness, with *Vu* which behaves in a peculiar way, holding previous CoBE values during release phase. This behaviour may be useful for percussive sounds analysis or transient detection, since it holds the brightness of the attack phase and ignores the release phase. Nevertheless, for all other purposes, the use of *Vu* as envelope follower for CoBE is discouraged. As regards *Max*, it shows a very sharp function, which may be ideal for some application, but non for general purpose. The same can be stated for *RF* for its extreme smoothness.

In conclusion, *RMS* and *RF* seems to be the best choices for general purpose CoBE. In this context *RMS* is used, since it is easy to implement both in analogue and digital domain and, most important, it is related to a physical property (the *effective value* [3]) which applies to any signal.

3. TRANSIENT PRESENCE DETECTION BY ENERGY-ENVELOPE BRIGHTNESS (TRAP)

The main idea is to measure the brightness of the amplitude envelope using the CoBE algorithm. Applying CoBE to the signal envelope, instead of the signal itself, should reveal that continuous amplitude envelopes (where sound is likely to be a smooth modulation of features) will produce a low CoBE value, while crispy amplitude envelopes (corresponding to strong amplitude modulations or numerous transients) will present a high CoBE value. Two examples of TRAP signal behaviour are shown in Fig. 4 and Fig. 5.

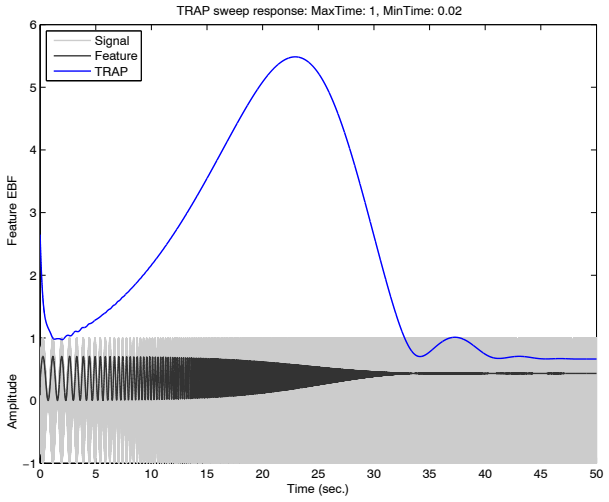


Figure 4. TRAP signal for an input created using a low frequency sweep as modulation for a 1kHz sine wave.

3.1 Implementation and tuning

We choose *RMS* also as the follower that will produce the main envelope signal, basically for the same reason we choose *RMS* as follower inside CoBE. To avoid confusion we will refer to the signal envelope as the *feature*, and to

[3] RMS. The value of the direct current that would produce the same power dissipation in a resistive load.

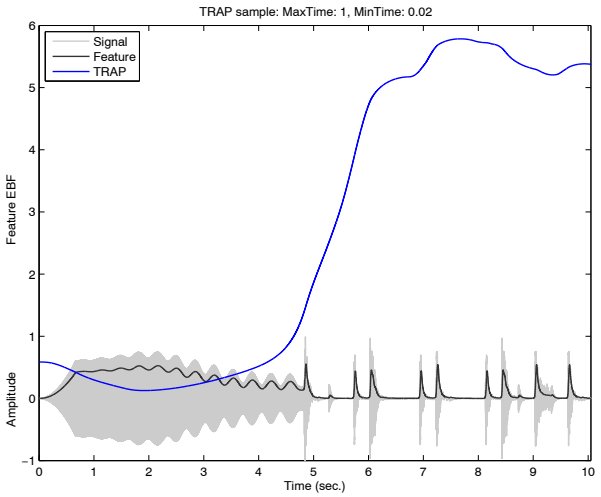


Figure 5. Output of the TRAP algorithm. The input file has been created to show different behaviours.

the algorithm used inside CoBE as the *kernel*. The window size of the feature follower (called *minTime* in the code) basically defines what is the minimum distance between sound events or, in other words, how smooth has to be the envelope that will be fed to the CoBE algorithm. This feature is then down-sampled to spare computing power and then fed into CoBE. The window size of the kernel (called *maxTime*) defines the overall smoothness of the output: smaller windows accentuates short therm variations of the envelope, while larger windows will generate smooth outputs. Finally, to make the algorithm independent from sampling rate, we consider EBF instead of the mere CoBE value.

The following code is an example of how this can be implemented in Matlab, the algorithm is also represented in Fig. 6.

```
function G =TRAP(X,maxTime,minTime,fs)
        % Kernel and Feature functions
        Feature = @RMSEnvelope;
        Kernel  = @RMSEnvelope;
        % Time to samples conversion
        minTime = floor(minTime*sr);
        k = 100; sre = fs/k;
        maxTime = floor(maxTime*sre);
        % Feature extraction
        E = Feature(X,minTime);
        E = downsample(E,k);
        % Feature EBF extraction
        [~,G] = CoBE(E,fs,Kernel,
            maxTime);
end
```

Lowering *minTime* too much makes the algorithm fitting the waveform instead of the envelope, thus introducing noise. This noise increase considerably the envelope brightness and EBF. On the other hand, higher values of *minTime* may ends in ignoring transients or short burst of signal. We empirically found that a value between 0.0125 and 0.0250 seconds may be suitable for most situation. Best results were

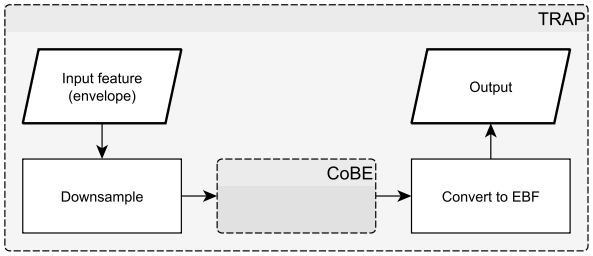


Figure 6. Diagram of the TRAP algorithm.

obtained with $minTime = 0.02$. This window size can detect variations up to 50 Hz, while higher frequencies will be smoothed out and considered as a continue envelope.

Please note that only those results obtained with the same *minTime* are fully comparable, for this reason we suggest to set $minTime = 0.02$ as conventional starting point [4]. For comparison different values of *minTime* are shown in Fig. 7.

For what concerns *maxTime*, high values average out the whole signal, while low values (*maxTime*<0.5) fit the signal more precisely, magnifying the sharp amplitude modulations of the signal. In this case, a default value of 1 second may fit most of the scenarios. The behaviour obtained with different *maxTime* values is shown in Fig. 8.

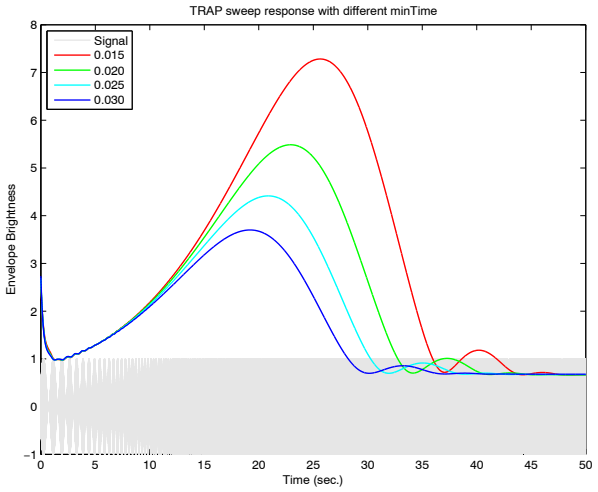


Figure 7. Same signal of Fig. 4 analysed with different *minTime* values.

4. EVALUATION AND TESTING

To inspect the information redundancy carried by the TRAP signal, correlation analysis with other features is performed.

Since the richness of the envelope may depends on the presence of transients, we took into account spectral descriptors normally used in onset detection tasks [5] ([9], [10], [11]), besides common time domain energy descriptors (listed below).

[4] This value is the same default value provided by MIRToolbox as time constant for *mirenvelope*.

[5] Please note that even if correlated onsets and transients are not exactly the same.

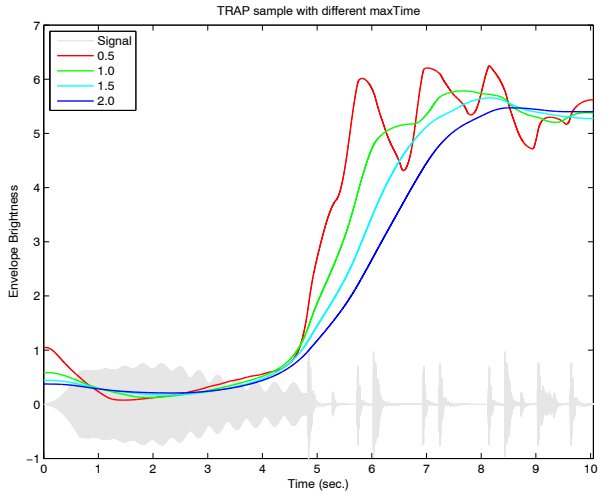


Figure 8. Same signal of Fig. 5 analysed with different *maxTime* values.

Chosen features can be grouped in two main categories: *Monodimensional time-varying* features, each represented by a single time series, and *general descriptors*, where each feature is represented with a scalar value. Time series are then collapsed to scalar values by taking the median value and interquartile range (IQR); as pointed out by [12]. Those measures are more stable and resilient to silence segments and outliers than mean and standard deviation.

Chosen features are shown in Table. 1.

Type	Name	Reference
General	Pulse clarity	[13]
	Event density	[3]
	Low energy	[3]
	Modulation frequency	[12]
	Modulation amount	[12]
Time varying	TRAP	
	CoBE EBF	[1]
	RMS	
	Peak	
	Crest factor	
	Attack leap	[3]
	Spectral Flux	[3]
	Centroid	[3]
	Flatness	[3]
	Hi-Frequency Content (HFC)	[7], [15]

Table 1. Extracted features

The sound samples are divided into 5 groups:

- 10 monophonic instruments taken from the MUMS database [16], characterized by a pizzicato or percussive excitation;
- 10 monophonic bowed or wind instruments taken from the MUMS database;

- 10 segments of orchestral music;
- 10 segments of POP music taken randomly within POP sub-genres;
- 10 voice recordings containing various examples (singing and spoken, males and females).

Samples from the MUMS database are made of single notes interleaved with silence. This files were manually edited to make silence between notes constant to 100 ms. To obtain a robust correlation analysis we decided to use Spearman rank correlation, instead of the typical linear Pearson correlation as proposed in [12]. For the feature extraction we used MIRToolbox [3] and TimbreToolbox [12]. Results are shown in Table 2 and Table 3. Fig. 9 shows a dendrogram built using *1 - ABS(correlation)* as distance to try to reveal a hierarchy of the extracted features.

Feature	**Correlation**	p**-value**
Modulation amount	0,76	<0,05
Event density	0,60	<0,05
Centroid (IQR)	0,55	<0,05
CoBE EBF (IQR)	0,54	<0,05
TRAP (IQR)	0,52	<0,05
Flatness (IQR)	0,50	<0,05
Attack leap (med)	0,45	<0,05
Flatness (med)	0,45	<0,05
HFC (med)	-0,43	<0,05
Spectral flux (med)	0,41	<0,05
Centroid (med)	0,37	<0,05
Peak (IQR)	0,34	<0,05
Crest factor (IQR)	0,34	<0,05
RMS (IQR)	0,33	<0,05
Low energy	0,31	<0,05
Decay	-0,31	<0,05

Table 2. TRAP median, correlation with other features and p-value (sorted by decreasing absolute correlation, only significative values are reported.)

Feature	**Correlation**	p**-value**
Modulation amount	0,55	<0,05
TRAP (med)	0,52	<0,05
Low energy	0,40	<0,05
Flatness (IQR)	0,40	<0,05
HFC (med)	-0,38	<0,05
HFC (IQR)	-0,34	<0,05
Centroid (IQR)	0,32	<0,05
RMS (med)	-0,29	<0,05

Table 3. TRAP IQR correlation with other features and p-value (sorted by decreasing absolute correlation, only significative values are reported.)

As shown by the dendrogram in Fig. 9, the "distance" between TRAP and other time varying features is low thus implying that it provides different information. Table 2 and Table 3 show that correlation, when present, is significant, in particular the features that seems to be more related to

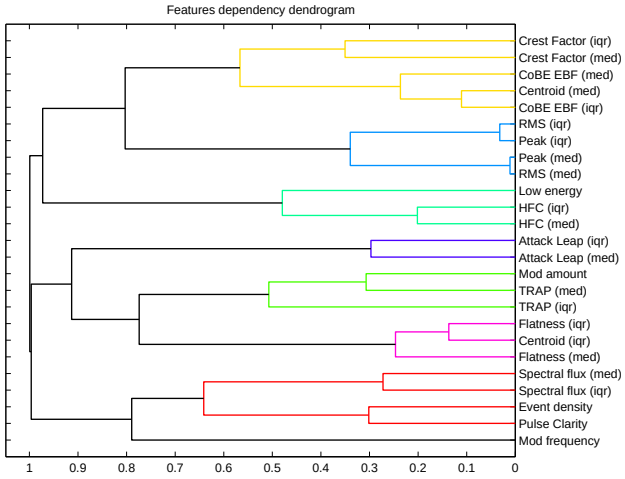


Figure 9. The dendrogram extracted from the correlation data shows the hierarchy of the investigated features.

TRAP are: *Energy Modulation Amount*, *Event Density* and *Centroid IQR*.

Energy Modulation Amount and *Event Density* are exactly the features we expected to see as the most correlated, since they affect the energy envelope (the former more explicitly than the latter). Also the *Spectral Centroid interquartile range*, with other spectrum-dependent interquartiles, are correlated with TRAP median. This can be explained by the fact that changes in timbre may correspond to different sound events and variations in the energy envelope, and during transients variations of spectral features are commonly found.

Time consumption analysis has been made comparing TRAP computing time with some of the most correlated features: *Energy Modulation Amount*, *Event Density*, *Flatness*, *Centroid* and *Low Energy*.

The results are shown in Fig. 10 and Table 4 and prove the implementation to be useful in terms of computational time, especially in the case of *Event Density* and *Energy Modulation Amount*.

Feature	**Processing time**	**Ratio**
Event density	48,89 ms	5,88
Energy Modulation	22,05 ms	2,65
Flatness	15,74 ms	1,89
Centroid	14,26 ms	1,72
TRAP	08,31 ms	1,00 (ref)

Table 4. Median time necessary to compute one second of audio and ratio with TRAP time. Data computed from those in Fig. 10

Finally, in Fig. 11, we scattered the sound samples to show the distribution of TRAP.

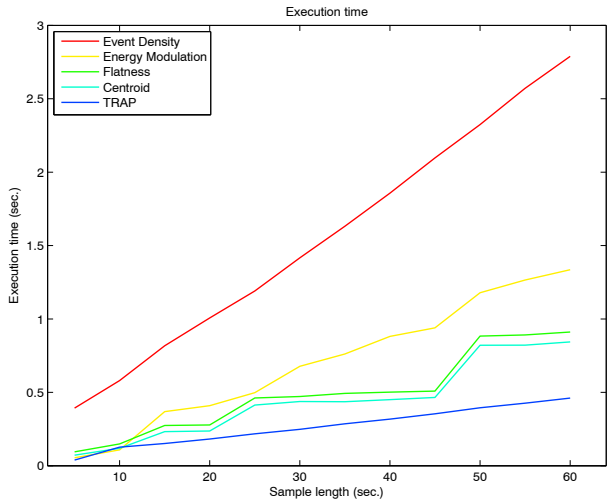


Figure 10. Average execution time for audio samples of different length. The test run on a common laptop computer with an Intel I5 processor with a clock frequency of 1.7 GHz

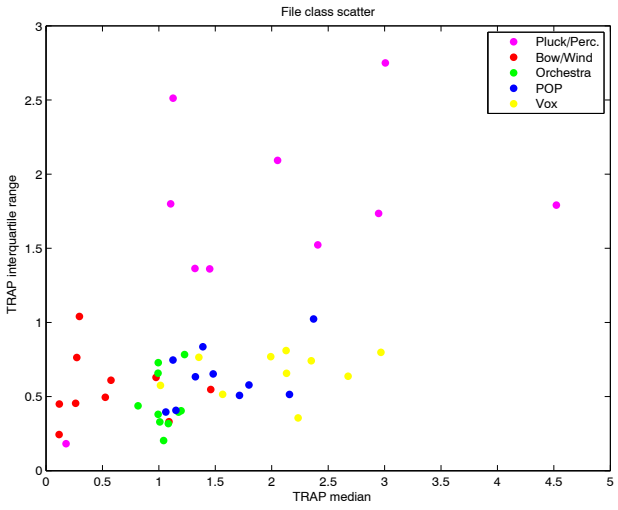


Figure 11. TRAP median and IQR used to plot the dataset.

5. CONCLUSIONS & FUTURE WORKS

A descriptor for features' shape has been proposed. In particular this method has been applied to energy envelope and has been proved as an indicator for transient presence and energy modulations.

TRAP has been used to distinguish between continuous signals and discrete acoustic events in [2]. With appropriate thresholding, it is useful to describe the presence of transient in segments of sounds.

It might also serve to create automatic dynamics processors that change their behaviour according to the content of the signal. Another possibility is to apply this very same method not to energy envelope but to other features (e.g. the pitch contour).

In order to better explain results an experimental set-up for testing perceptual correlations is advised: simple signals (amplitude modulated noise/sine waves) clustered by this feature and by humans can be compared. Finally, to overcome the possible limitation of the envelope follower method as presented in Section 2.2 a comparison of different approach can be taken into consideration.

6. REFERENCES

[1] G. Presti and D. Mauro, "Continuous brightness estimation (cobe): Implementation and its possible applications," in *10th International Symposium on Computer Music Multidisciplinary Research (CMMR)*. Laboratoire de Mécanique et d'Acoustique, 2013, pp. 967–974.

[2] L. A. Ludovico and G. Presti, "The sonification space: a reference system for sonification tasks," *Accepted for Journal on Human Computer Studies: special issue on Data sonification and sound design in interactive systems*, 2015.

[3] O. Lartillot and P. Toiviainen, "Mir in matlab (ii): A toolbox for musical feature extraction from audio," in *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, Austria, September 23-27 2007, pp. 127–130.

[4] P. N. Juslin, "Cue utilization in communication of emotion in music performance: relating performance to perception." *Journal of Experimental Psychology: Human perception and performance*, vol. 26, no. 6, p. 1797, 2000.

[5] 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, North Western University, Illinois*, 2004, pp. 112–116.

[6] P. Laukka, P. Juslin, and R. Bresin, "A dimensional approach to vocal expression of emotion," *Cognition & Emotion*, vol. 19, no. 5, pp. 633–653, 2005.

[7] P. Masri and A. Bateman, "Improved modelling of attack transients in music analysis-resynthesis," in *Pro-*

ceedings of the International Computer Music Conference. Citeseer, 1996, pp. 100–103.

[8] A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the meter of acoustic musical signals," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 342–355, 2006.

[9] S. Dixon, "Onset detection revisited," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*. Citeseer, 2006, pp. 133–137.

[10] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 1035–1047, 2005.

[11] N. Collins, "A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions," in *Audio Engineering Society Convention 118*. Audio Engineering Society, 2005.

[12] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams, "The timbre toolbox: Extracting audio descriptors from musical signals," *The Journal of the Acoustical Society of America*, vol. 130, no. 5, pp. 2902–2916, 2011.

[13] O. Lartillot, T. Eerola, P. Toiviainen, and J. Fornari, "Multi-feature modeling of pulse clarity: Design, validation and optimization." in *ISMIR*. Citeseer, 2008, pp. 521–526.

[14] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," 2004.

[15] K. Jensen and T. H. Andersen, "Real-time beat estimationusing feature extraction," in *Computer Music Modeling and Retrieval*. Springer, 2004, pp. 13–22.

[16] F. J. Opolko and J. Wapnick, *MUMS: McGill University Master Samples*. McGill University, Faculty of Music, 1989.

HOW WELL CAN A MUSIC EMOTION RECOGNITION SYSTEM PREDICT THE EMOTIONAL RESPONSES OF PARTICIPANTS?

Yading Song and Simon Dixon
Centre for Digital Music
Queen Mary University of London
{y.song, s.e.dixon}@qmul.ac.uk

ABSTRACT

Music emotion recognition systems have been shown to perform well for musical genres such as film soundtracks and classical music. It seems difficult, however, to reach a satisfactory level of classification accuracy for popular music. Unlike genre, music emotion involves complex interactions between the listener, the music and the situation. Research on MER systems is handicapped due to the lack of empirical studies on emotional responses. In this paper, we present a study of music and emotion using two models of emotion. Participants' responses on 80 music stimuli for the categorical and dimensional model, are compared. In addition, we collect 207 musical excerpts provided by participants for four basic emotion categories (happy, sad, relaxed, and angry). Given that these examples represent intense emotions, we use them to train musical features using support vector machines with different kernels and with random forests. The most accurate classifier, using random forests, is then applied to the 80 stimuli, and the results are compared with participants' responses. The analysis shows similar emotional responses for both models of emotion. Moreover, if the majority of participants agree on the same emotion category, the emotion of the song is also likely to be recognised by our MER system. This indicates that subjectivity in music experience limits the performance of MER systems, and only strongly consistent emotional responses can be predicted.

1. INTRODUCTION

With technological and social changes in our daily lives, the experience of music has changed at a fundamental level. Music can be heard at far more diverse places, and people report that the primary reason for listening to music lies in its emotional effects, the induction and expression of emotions [1]. Because of the emotional function of music, over the past decade, the study of music and emotion has become increasingly important, and has attracted research from different fields, for instance, computer science [2], musicology, and psychology [3]. Previous studies on emotion provide us with a better understanding of music and emotion, which can also help improve the design of subjective music recommendation systems [4].

For music information retrieval (MIR) researchers, music emotion recognition (MER) systems have been widely discussed [2, 5]. On the one hand, previous studies using musical features have applied various machine learning approaches (e.g., support vector machines [6], k-nearest neighbours [7], random forests [8], regression models [9], and deep belief networks [10, 11]). Although these techniques perform well for genres such as classical music and film soundtracks, the recognition accuracy for popular music fails to reach a satisfactory level [7, 8]. On the other hand, psychological studies in music have focussed on emotional responses to music [12], emotion models [13], emotion experience and recognition [14], and cross-cultural emotion perception in music [15, 16]. The comparison between listeners' responses using the categorical and dimensional model are often neglected [13]. Other research also suggests that differences in individuals may affect how emotional meaning is elicited [17, 18]. In this study, however, we compare participants' responses in general, rather than individual factors such as one's personality, current mood, or culture.

Emerging from research in both computer science and psychology, we study the differences between music emotion recognition systems and participants' responses using two models of emotion, the categorical and dimensional model. Therefore, the goals of this paper are, (1) To compare participants' responses for two models of emotion; (2) To provide a user-suggested dataset of musical excerpts for four basic emotions (happy, sad, relaxed and angry); (3) To study the differences between machine learning approaches (e.g., support vector machines and random forest) and participants' responses.

2. MUSIC AND EMOTION

2.1 Emotional Responses

One important distinction in music is between *perceived emotion* (or expressed emotion) which is an emotion expressed by music, and *induced emotion* (or felt emotion) which is an emotion felt in response to music. In general, music evokes emotions similar to the emotions perceived in music [18, 19]. However, some research suggests that responses for induced emotion are generally positive [1], and responses for perceived emotion are more consistent [14].

Copyright: ©2015 Yading Song 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.

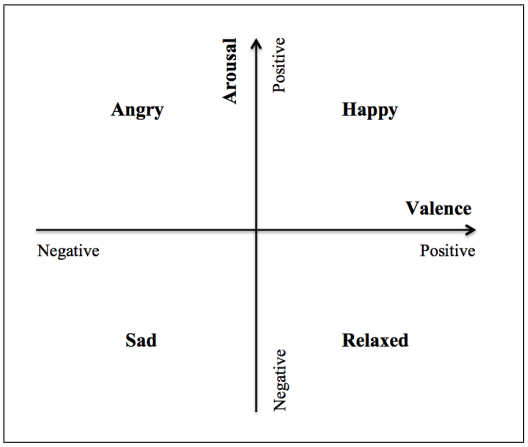


Figure 1. A mapping between a categorical (happy, sad, relaxed and angry) and dimensional model (valence and arousal) of emotion.

2.2 Emotion Models

Although different emotion models such as miscellaneous [20] and domain-specific [21] models have been proposed in the past, the most popular ones are the categorical and dimensional models. The typical dimensional models of emotion represent emotions in an affective space with two dimensions: one related to valence (a pleasure-displeasure continuum), and the other to arousal (activation-deactivation) [22]. Previous studies using the dimensional model have suggested that prediction for arousal is more consistent than for valence [23]. In contrast, the categorical model represents all emotions as being derived from a limited number of universal and innate basic emotions such as happiness, sadness, fear, and anger [24]; and is often used in the study of perceived emotion [3]. In this study, both categorical and dimensional models are used, and to compare the results between these two models of emotion, a mapping is provided in Figure 1. We used four basic emotion classes: happy, angry, sad, and relaxed, considering these four emotions are widely accepted across different cultures and cover the four quadrants of the two-dimensional model of emotion [25].

3. DATA COLLECTION

The majority of studies on music and emotion have used film soundtracks and classical music [17, 26]. Compared with other musical genres, there has been a lack of MER research on popular music [25, 27–29]. Although social tags provide us with highly relevant metadata such as genre, mood, and instrument [30], participants' agreement with emotion tags such as "relaxed" is still very low [31, 32].

3.1 Musical Excerpt Collection

In a previous listening experiment on music emotion using the categorical model [32], forty participants were asked to provide examples of songs (song title and artist's name) that represent each of the four basic emotions (happy, sad, relaxed, and angry) in perceived and induced emotion. Given that music evokes emotions similar to the emotions perceived in music, the examples for perceived and induced emotion were aggregated for this study. If the same excerpt was mentioned in both perceived and induced emotion, the song is only counted once. However, some participants mentioned only the artist name (e.g., Death Cab for Cutie, Mayday Parade, and Bandari), or the album name (e.g., The Dark Side of the Moon), so this information was not considered for further analysis. Musical excerpts were then fetched via the 7Digital API [1] or Amazon mp3 [2]. A total of 207 songs were collected in this way, with the distribution over emotion categories as shown in Table 1.

In contrast to songs retrieved using emotion tags, these examples are considered more likely to represent intense emotions. A music example from each emotion category is shown in Table 2. The dataset (song title, artist's name, 7digital ID, and musical features) is made available to encourage other researchers to reproduce the results for research and evaluation [3].

Emotion category	No. of examples
Happy	59
Sad	58
Relaxed	48
Angry	42
Total	207

Table 1. The distribution of musical examples provided by participants.

Emotion category	Song title	Artist name
Happy	Wannabe	Spice Girls
Sad	Fix You	Coldplay
Relaxed	Eggplant	Michael Franks
Angry	Fighter	Christina Aguilera

Table 2. User-provided examples for each emotion category.

3.2 Emotion Ratings

A separate eighty (n = 20 for each emotion category) popular musical excerpts were randomly selected from a data set of 2904 songs that had been tagged with one of the four words "happy", "sad", "relaxed", and "angry" [6]. These 80 musical excerpts were given in random order to forty participants using the categorical model [31], and fifty-four participants using the dimensional model [32]. Previous research showed a higher consistency in participants' perceived emotional responses. Therefore, only perceived emotional responses are considered in this study. For the categorical model, participants were asked to choose from one of the following options: happy, sad, relaxed, angry, and "cannot tell"/"none of the above". For the dimensional model, participants were asked to rate on an 11-point scale for the two core dimensions: valence (sad-happy) and arousal (calm-excited). Their ratings were aggregated, and a summary of the responses, participants' profiles, and musical excerpts is made publicly available [19] [3].

[1] http://developer.7digital.com/

[2] http://www.amazon.co.uk/Digital-Music/b?ie=UTF8&node=77197031

[3] https://code.soundsoftware.ac.uk/projects/emotion-recognition/repository

Dimension	Description
Dynamics	RMS energy, slope, attack, low energy
Rhythm	tempo, fluctuation peak (pos, mag)
Spectral	spectrum centroid, brightness, spread, skewness, kurtosis, rolloff95, rolloff85, spectral energy, spectral entropy, flatness, roughness, irregularity, zero crossing rate, spectral flux, MFCC, DMFCC, DDMFCC
Harmony	chromagram peak, chromagram centroid, key clarity, key mode, HCDF

Note. The mean and standard deviation values were extracted, except for the feature "low energy", for which only the mean was calculated.

Table 3. Features extracted from the audio data.

3.3 Musical Feature Extraction

Two different emotion datasets, training and testing, are used in our experiment. The training dataset, which is provided by participants, contains 207 songs. The testing dataset contains 80 musical excerpts (*n* = 20 for each emotion category). These musical excerpts for testing range from recent releases back to 1960s, and cover a range of Western popular music styles such as pop, rock, country, metal, and instrumental. Each excerpt was either 30 seconds or 60 seconds long (as provided by 7Digital [1]). Previous studies have suggested that emotion can be recognised within a second [17,33]. To expand both the training and testing datasets, each excerpt was split into 5-second clips with 2.5-second overlap. Musical features were then extracted using MIRtoolbox 1.5 [34] [4] for both the full 30/60-second excerpts and the 5-second clips. The musical features extracted are shown in Table 3.

4. RESULTS

4.1 Participants' Responses for the Two Models of Emotion

To compare participants' responses for the categorical and dimensional models, their ratings were aggregated by label (for the categorical model: happy, sad, relaxed, and angry; for the dimensional model: valence and arousal). The ratings of valence and arousal in the dimensional model were mapped to the four basic emotions in the categorical model (see Figure 1). We calculated the inter-rater reliability (Fleiss's Kappa) for participants' ratings using the categorical ($\kappa = 0.31$) and dimensional model ($\kappa = 0.25$). In addition, for each stimulus, we took the label with the greatest number of votes to be the dominant emotion in each model. If the same dominant emotion was found in both categorical and dimensional models, the song was marked as a "match" (53 cases), otherwise "no match" (27 cases). However, three responses using the categorical model and one response using the dimensional model received equal number of votes (e.g., angry with happy, happy with sad, and sad with relaxed), and they were considered as "no match".

Although over the half of the excerpts received the same emotion in both models of emotion, the participants' consistency (the greatest number of votes on the four emotions) between "match" and "no match" cases is still unclear. Therefore, a Kruskal-Wallis one-way analysis of variance test was conducted on participants' consistency between "match" and "no match" cases for two models of emotion. As expected, a significant higher consistency can be found for "match" cases in both categorical ($Median = 0.70$, $Std = 0.19$, $\chi^2(1, N = 80) = 8.79$, and $p < .05$) and dimensional models ($Median = 0.74$, $Std = 0.09$, $\chi^2(1, N = 80) = 4.91$, and $p < .05$) than "no match" cases (for categorical model: $Median = 0.50$, and $Std = 0.13$; for dimensional model:$Median = 0.68$, and $Std = 0.10$). However, no significant differences were found for the two core dimensions, valence ($\chi^2(1, N = 80) = 3.79$, and $p > .05$) and arousal ($\chi^2(1, N = 80) = 1.05$, and $p > 0.05$).

Among the 27 "no match" cases, 10 were collectively confused between the emotions "sad" and "relaxed". When participants' responses were not consistent, it is important to know how machine learning approaches perform. Therefore, we built emotion classifiers using support vector machines and random forest approaches.

4.2 Emotion Recognition Using Machine Learning Approaches

207 excerpts provided by participants were used for training (see Section 3.3). However, a smaller training size may influence classification performance. To expand the data, each audio file was split into 5-second clips with 2.5-second overlap. Therefore, 207 (30/60 seconds) and 2990 (5 seconds) musical clips were collected, and trained separately.

4.2.1 Training

We adopted a 10 fold cross-validation approach, where for each song, all clips were placed in a single fold to avoid overfitting, and chose support vector machines (SVM) with different kernels (e.g., linear, radial basis function, and polynomial) and random forests (RF) as classifiers for training. We used the implementation of the sequential minimal optimisation algorithm in the Weka 3-7-11 data mining toolkit [5] . 55 musical features extracted from MIRtoolbox for both the 30/60-second (*N* = 207), and 5-second (*N* = 2990) datasets were used, with the recognition results shown in Table 4.

The RF approach and SVM with linear kernel both performed well, and recognition accuracy using 5-second clips was 1% higher (but not significantly) than for the full excerpts. Although RF using 5-second clips performed best, it still did not reach a satisfactory level. From the confusion matrix, we noticed that classification for the emotion *relaxed* was also collectively confused with *sad*.

4.2.2 Testing

In training, RF gave the best classification accuracy using 5-second clips, and performed time efficiently. Therefore,

[4] https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox/MIRtoolbox1.5Guide

[5] http://www.cs.waikato.ac.nz/ml/weka/

	Recognition Accuracy	
Approaches	30-sec clips	5-sec clips
SVM w/ linear kernel	**39.04%**	40.35%
SVM w/ RBF kernel	28.57%	26.89%
SVM w/ poly kernel	37.62%	29.16%
Random forests	38.57%	**40.75%**

Note. For the training of 5-second clips, the clips from the same song if used in training, were not used for testing. Due to the unbalanced ground truth data for training, the results might be biased.

Table 4. Comparison of classification performance using support vector machines and random forest approaches.

this approach (i.e., RF with 5-second clips) was also applied on the 80 popular musical excerpts. Similar to the data expansion for the training dataset, each audio clip in the testing dataset was also split into 5-second clips (N = 1292). Section 4.3 shows the recognition results in comparison to participants' emotional responses.

4.3 Responses from Participants and the Recognition System

As each excerpt was split into 5-second chunks, each clip was recognised as expressing one emotion. The label with the greatest number of votes from the four emotions was chosen, and the greatest number of votes (consistency) for each excerpt was calculated as well. To compare the responses between outputs from the recognition system and participants for two models of emotion, Pearson's correlation analysis was conducted on the consistency for the 80 musical excerpts.

Table 5 shows that recognition consistency for each excerpt using RF approach is positively correlated with participants' consistency in the categorical ($r(78) = .23$, and $p < .05$) and dimensional models ($r(78) = .36$, and $p < .01$). It tentatively suggests that regardless of the emotion, the consistency of the recognition system is very similar to the consistency of participants' responses.

	Recognition	**Categorical**
Categorical	.23*	
Dimensional	.36**	.32**

Note. $^*p < .05$, $^{**}p < .01$.

Table 5. Correlation between the consistency from recognition system using the RF approach and participants' responses.

To explore participants' responses for each emotion, correlation analyses were further conducted on the emotion vote distribution from each excerpt for the recognition system and participants' responses. Table 6 and Table 7 show that no matter which emotion model is used, the emotion vote distribution from the recognition system and participants' responses is highly correlated (i.e., happy, relaxed, and angry). Interestingly, responses for relaxed from the categorical model are also correlated with sad from the recognition system. It suggests that both system and people find it difficult to distinguish between *sadness* and *relaxedness*. The same results could be found in the results for the dimensional model, where significant correlations were shown in the ratings of arousal, whereas only weak correlations were found in the responses of valence (e.g., happy with angry, and relaxed with sad).

		Categorical model			
		Happy	Sad	Relaxed	Angry
Pred.	Happy	.42***	-.33**	-.32**	.13
	Sad	-.16	.18	.33**	-.30**
	Relaxed	-.07	.00	.46***	-.22
	Angry	.07	-.31**	-.39***	.52***

Note. $^*p < .05$, $^{**}p < .01$, and $^{***}p < .001$.

Table 6. Correlation between the responses from recognition system and participants using the categorical model.

		Dimensional model			
		Pos V	Neg V	Pos A	Neg A
Pred.	Pos V	.33**	-.34**	.10	-.12
	Neg V	-.10	-.01	.15	-.16
	Pos A	.22	-.28*	.59***	-.64***
	Neg A	-.02	-.01	-.42***	.44***

Note. $^*p < .05$, $^{**}p < .01$, and $^{***}p < .001$.

Table 7. Correlation between the responses from recognition system and participants using the dimensional model.

Finally, the dominant label(s) from each experiment (recognition system and responses from categorical and dimensional models of emotion) were compared. Considering the same dominant emotion label from both dimensional and categorical model as the ground truth, 32 responses out of 53 (accuracy = 60%) from recognition system were classified correctly. However, if we consider the dominant emotion labels from both categorical and dimensional models, 51 responses out of 80 (accuracy = 64%) are classified correctly.

In spite of the recognition accuracy given by the random forest approach, the incorrect classification results were compared with participants' responses. We found that majority of songs given the incorrect classifications had opposite signs for valence, confusing sad with relaxed and angry with happy. It suggests that compared with arousal, valence is more difficult to recognise. This also agrees with previous studies using regression models [9].

We noticed that if the recognition results were incorrect, it was likely that the emotion of a song itself was ambiguous. Examples for each emotion are provided in Table 8. For example, for the song "Josephine" by *Wu-Tang Clan*, the dominant emotion was chosen as relaxedness in both the categorical and dimensional models, whereas it was recognised as sad by the machine. The distribution, however, shows that 8 clips from the same excerpt were classified as relaxed, and another 10 clips were classified as angry. In addition, participants' responses for both models of emotion were also distributed across four emotions. Similarly, for the song "Blood On The Ground" by *Incubus*, participants all agreed on arousal level, but the responses for valence were ambivalent. Likewise, the recognition result was also influenced by this uncertainty.

Interestingly, we found the song "Anger" by *Skinny Puppy* was recognised as angry by all participants, whereas the recognition system classified it as happy. Possible explanations could be the selection of clips, that some parts are

	Recognition					Categorical					Dimensional				
Title	H	S	R	A	Label	H	S	R	A	Label	PoV	NeV	PoA	NeA	Label
If the Creeks Don't Rise	1	13	9	0	Sad	7	3	5	0	Happy	17	8	16	8	Happy
Loves Requiem	0	2	9	0	Relaxed	0	16	3	0	Sad	1	24	4	23	Sad
Josephine	5	0	8	10	Angry	2	5	6	5	Relaxed	13	10	7	14	Relaxed
Blood On The Ground	3	1	6	1	Relaxed	1	1	3	13	Angry	9	13	24	0	Angry

Note. H - Happy, S - Sad, R - Relaxed and A - Angry.

Table 8. Examples of vote distribution on emotion for the recognition system, categorical and dimensional models.

expressing happiness. It is also reasonable to guess that the emotion perceived is genre-specific (e.g., metal as anger, and pop music as happy) and cultural-dependent. Participants may also be influenced by titles or lyrics.

5. DISCUSSION AND FUTURE STUDIES

In this paper, we presented an empirical study of music and emotion, comparing the results between a music emotion recognition system and participants' responses for two models of emotion. Firstly, we studied the emotional responses for 80 popular musical excerpts in the categorical and dimensional models of emotion. The analysis showed similar responses for both emotion models. A positive correlation between categorical and dimensional models was also found on the rating consistency for each musical excerpt.

A separate 207 musical excerpts were collected from participants for four basic emotion categories (i.e., happy, sad, relaxed, and angry). Our emotion recognition model was trained using support vector machines and random forest classifiers. Two different training datasets were compared, one using the entire 30/60-second audio files and the one using multiple 5-second segments with 2.5-second overlap from the same excerpt. Audio features were extracted using MIRtoolbox. The results showed that the support vector machine with linear kernel and random forest approaches performed best, and the use of 5-second clips increased the classification accuracy by only 1%. In addition, the recognition system did not classify emotions well for the emotions sadness and relaxedness. One of the possible reasons for the low accuracy of music emotion recognition systems could be the user-suggested dataset, which was mixed with both perceived and induced emotion. Another explanation could be the subjective nature of music emotion perception.

Finally, the time-efficient random forest with 5-second clips approach was applied on the 80 musical excerpts for testing. The analysis showed that responses from the recognition system were highly correlated with participants' responses for the categorical and dimensional models. Moreover, the distribution of responses for each emotion was also highly correlated. However, significant correlations between sadness and relaxedness in the categorical model suggest that listeners and emotion recognition systems have difficulty distinguishing valence (positive and negative emotions). Similarly, strong correlations were found for responses of arousal, whereas only weak correlations were shown in responses for valence. The comparison of emotion distribution also indicates that the performance of music emotion recognition systems is similar to participants' emotional responses for the two models of emotion. Additionally, the prediction accuracy is higher for songs where participants agreed more. This suggests that only strongly consistent emotional responses can be predicted by the music emotion recognition systems.

Due to the dynamic nature of music, emotions may vary over time and the emotion recognition accuracy may also be affected by the selection of clips. More importantly, music emotion involves complex interactions between the listener, the music, and the situation. The perception of music is most likely influenced by individual differences such as age, music skills, culture, and music preference [35–38]. The experience of emotions may also vary according to various situational contexts. Therefore, our future work is to incorporate emotion into the design of a subjective music recommendation system, and also to study the influence of situational contexts and individual differences such as culture in the emotion perception of music.

Acknowledgments

We are very grateful to all the participants, and reviewers for their valuable suggestions. We would like to thank the China Scholarship Council (CSC) for financial support and the Centre for Digital Music (C4DM).

6. REFERENCES

[1] P. N. Juslin and P. Laukka, "Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening," *Journal of New Music Research*, vol. 33, no. 3, pp. 217–238, 2004.

[2] Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbull, "Music emotion recognition: A state of the art review," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 255–266, Utrecht, Netherlands, 2010.

[3] T. Eerola and J. K. Vuoskoski, "A review of music and emotion studies: Approaches, emotion models, and stimuli," *Music Perception: An Interdisciplinary Journal*, vol. 30, no. 3, pp. 307–340, 2013.

[4] Y. Song, S. Dixon, and M. T. Pearce, "A survey of music recommendation systems and future perspectives," in *Proceedings of the 9th International Symposium on Computer Music Modeling and Retrieval (CMMR)*, pp. 395–410, London, UK, 2012.

[5] Y.-H. Yang and H. H. Chen, "Machine recognition of music emotion," *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 3, pp. 1–30, 2012.

[6] Y. Song, S. Dixon, and M. T. Pearce, "Evaluation of musical features for emotion classification," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 523–528, Porto, Portugal, 2012.

[7] P. Saari, T. Eerola, and O. Lartillot, "Generalizability and simplicity as criteria in feature selection: Application to mood classification in music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1802–1812, 2011.

[8] T. Eerola, "Are the emotions expressed in music genre-specific? An audio-based evaluation of datasets spanning classical, film, pop and mixed genres," *Journal of New Music Research*, vol. 40, no. 4, pp. 349–366, 2011.

[9] Y. Yang, Y. Lin, Y. Su, and H. H. Chen, "A regression approach to music emotion recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 2, pp. 448–457, 2008.

[10] E. M. Schmidt and Y. E. Kim, "Learning emotion-based acoustic features with deep belief networks," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 65–68, New Paltz, New York, USA, 2011.

[11] E. M. Schmidt, J. Scott, and Y. E. Kim, "Feature learning in dynamic environments: Modeling the acoustic structure of musical emotion," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 325–330, Porto, Portugal, 2012.

[12] P. N. Juslin and D. Västfjäll, "Emotional responses to music: The need to consider underlying mechanisms," *The Behavioral and Brain Sciences*, vol. 31, no. 5, pp. 559–621, 2008.

[13] T. Eerola and J. K. Vuoskoski, "A comparison of the discrete and dimensional models of emotion in music," *Psychology of Music*, vol. 39, no. 1, pp. 18–49, 2010.

[14] A. Gabrielsson, "Emotion perceived and emotion felt: Same or different?" *Musicae Scientiae*, vol. 5, no. 1, pp. 123–147, 2002.

[15] K. Kosta, Y. Song, G. Fazekas, and M. B. Sandler, "A study of cultural dependence of perceived mood in Greek music," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 317–322, Curitiba, Brazil, 2013.

[16] X. Hu and J. H. Lee, "A cross-cultural study of music mood perception between American and Chinese listeners," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 535–540, Porto, Portugal, 2012.

[17] 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 & Emotion*, vol. 19, no. 8, pp. 1113–1139, 2005.

[18] K. Kallinen and N. Ravaja, "Emotion perceived and emotion felt: Same and different," *Musicae Scientiae*, vol. 10, no. 2, pp. 191–213, 2006.

[19] Y. Song, S. Dixon, M. T. Pearce, and A. R. Halpern, "Perceived and induced emotion responses to popular music: Categorical and dimensional models," *to appear in Music Perception*, pp. 1–46, 2015.

[20] S. McAdams, B. W. Vines, S. Vieillard, B. K. Smith, and R. Reynolds, "Influences of large-scale form on continuous ratings in response to a contemporary piece in a live concert setting," *Music Perception: An Interdisciplinary Journal*, vol. 22, no. 2, pp. 297–350, 2004.

[21] 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, 2008.

[22] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[23] A. Huq, J. P. Bello, and R. Rowe, "Automated music emotion recognition: A systematic evaluation," *Journal of New Music Research*, vol. 39, no. 3, pp. 227–244, 2010.

[24] P. Ekman, "An argument for basic emotions," *Cognition & Emotion*, vol. 6, no. 3/4, pp. 169–200, 1992.

[25] C. Laurier, J. Grivolla, and P. Herrera, "Multimodal music mood classification using audio and lyrics," in *International Conference on Machine Learning and Applications (ICMPLA)*, pp. 1–6, San Diego, California, USA, 2008.

[26] T. Eerola, O. Lartillot, and P. Toiviainen, "Prediction of multidimensional emotional ratings in music from audio using multivariate regression models," in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 621–626, Kobe, Japan, 2009.

[27] C. Mak, T. Lee, S. Senapati, Y.-T. Yeung, and W.-K. Lam, "Similarity measures for Chinese pop music based on low-level audio signal attributes," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 513–518, Utrecht, Netherlands, 2010.

[28] E. M. Schmidt and Y. E. Kim, "Prediction of time-varying musical mood distributions from audio," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 465–470, Utrecht, Netherlands, 2010.

[29] Y.-H. Yang and H. H. Chen, "Prediction of the distribution of perceived music emotions using discrete samples," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2184–2196, 2011.

[30] P. Lamere, "Social tagging and music information retrieval," *Journal of New Music Research*, vol. 37, no. 2, pp. 101–114, 2008.

[31] Y. Song, S. Dixon, M. T. Pearce, and G. Fazekas, "Using tags to select stimuli in the study of music and emotion," in *Proceedings of the 3rd International Conference on Music & Emotion (ICME)*, Jyväskylä, Finland, 2013.

[32] Y. Song, S. Dixon, M. T. Pearce, and A. R. Halpern, "Do online social tags predict perceived or induced emotional responses to music?" in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 89–94, Curitiba, Brazil, 2013.

[33] I. Peretz, "Listen to the brain: A biological perspective on musical emotions," in *Music and Emotion: Theory and Research*, P. N. Juslin and J. A. Sloboda, Eds. New York, NY, USA: Oxford University Press, pp. 105–134, 2001.

[34] O. Lartillot and P. Toiviainen, "MIR in Matlab (II): A toolbox for musical feature extraction from audio," in *Proceedings of the 8th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 237–244, Vienna, Austria, 2007.

[35] C. Z. Malatesta and M. Kalnok, "Emotional experience in younger and older adults," *Journal of Gerontology*, vol. 39, no. 3, pp. 301–308, 1984.

[36] P. J. Rentfrow and S. D. Gosling, "The Do Re Mi's of everyday life: The structure and personality correlates of music preferences," *Journal of Personality and Social Psychology*, vol. 84, no. 6, pp. 1236–1256, 2003.

[37] M. Shiota, D. Keltner, and O. John, "Positive emotion dispositions differentially associated with Big Five personality and attachment style," *The Journal of Positive Psychology*, vol. 1, no. 2, pp. 61–71, 2006.

[38] D. L. Novak and M. Mather, "Aging and variety seeking," *Psychology and Aging*, vol. 22, no. 4, pp. 728–737, 2007.

Exploring the General Melodic Characteristics of XinTianYou Folk Songs

Juan Li[+] **Lu Dong*** **Jianhang Ding*** **Xinyu Yang***

{ [+]Centre for Music Education, *Department of Computer Science & Technology}, Xi'an Jiaotong University

Emails: lijuan@mail.xjtu.edu.cn, {donglu666, jh.ding}@stu.xjtu.edu.cn, yxyphd@mail.xjtu.edu.cn

ABSTRACT

This paper aims to analyze one style of Chinese traditional folk song named Shaanxi XinTianYou. By analyzing the melody of this folksong genre, we make a clear, vivid, and thus easily approachable presentation of the cultural characteristics and significance of XinTianYou. Comparing to previous researches which mainly focus on mathematics and statistics, we further consider the musical continuity. Our insight is that, the combination of intervals reflects the characteristics of the music style. The significant pattern of the combinations can be used as representations of XinTianYou. We build a MIDI database, based on which the most representative combination of intervals are extracted. We propose to use N-Apriori algorithm which counts the frequent patterns of melody. Considering both the significance and similarity between music pieces, we provide a multi-layer melody perception clustering algorithm which uses both the melodic direction and the melodic value. The experiment results are analyzed based on both pattern mining techniques and music theories. For evaluation, we asked experts in this field to mark our results and proved that our results are consistent with the expert's intuition.

1. INTRODUCTION

XinTianYou is a style of improvised folk song in mountain area of Shaanxi province in China. It is the most important component of Shaanxi folk songs, and quite unique and attractive. Explortion on XinTianYou is beneficial to the understanding of this music style and also the applications such as music information retrieval. Moreover, it facilitates musical education and composition in Xintianyou as well as contributes to its inheritance, protection, and development. With the development of music digitalization, machine learning techniques have been widely used for music information analysis and greatly promoted the music research. Naoko kosugi.et.al [1] built a music dataset and developed a retrieval system. They compared the similarity between music pieces based on rhythm information. Li.et.al [2] improved this system by taking the pitch and rhythm into consideration. They proposed to do matching based on the geometrical similarity of melodic contours. Yang.et.al [3] investigated mood categories with audio features, and used it for comparison between Chinese music and western music. These research studies have proven to be effective for music retrial and music emotion analysis, but mainly focus on the differences between music genres rather than the general characteristics within a genre. In this paper we focus on research of general characteristics for one genre because it is potentially useful for the music composition and understanding of the development route of music as well.

In addition, most previous researches focus on single feature and the statistical analysis of the feature. For example, the Alicante set [4] contains 28 global features based on the statistics of pitch, duration and the mean or standard deviation. The Mckay set [5] contains 109 global features, based on device, texture, rhythm, dynamic, pitch statistics, melody and chords. These studies only focus on mathematics and statistics, and ignore the musical continuity.

Some patterns mining algorithms have been introduced in [6]. In this paper, a new perspective based on the combination of interval is proposed to consider the integrity and continuity of music. An interval reflects the relative tendency of the two pitches. The combination of intervals reflects the trend of a melody. We use the N-Apriori algorithm to mine for frequent interval combinations. Additionally, based on the redundancy-aware top-*k* idea, we further use a multi-layer melody perceptive cluster algorithm to cluster the frequent interval combinations. Then the significant patterns are selected as the general characteristics of XinTianYou.

The organization of the paper is as follows. We first explain the preprocessing of data in Section 2. Section 3 contains the description of the N-Apriori patterns mining algorithm. Next we present how to use the multi-layer melodic perception to do the clustering in Section 4. The experimental results are shown in Section 5. Conclusions and future research are in Section 6.

2. PREPROCESSING

We build a XinTianYou MIDI database for all the XinTianYou songs collected by [7]. We choose this format because features can be easily computed. Each MIDI file only contains one repeat of melody to avoid redun-

Copyright: © 2015 J. Li, L. Dong, J. Ding, X. Yang. 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.

dancy, and no ornament is added to the main melody. After these initial preparations, we first divide each piece from the database into segments, and then map the difference between adjacent pitches to the interval.

2.1 Melody Segmentation

Traditional Chinese music is normally the combination of short units. Some Chinese researchers have investigated the short melody units. Liu [8] reported a tricolor-theory about restricting the traditional style of music. In his article, several sequences of three notes within a 4th were found and named musical chromosomes. Wang [9, 10] observed that Chinese folk music includes five phonological systems and each of them is a specific interval structure composed of three or four notes. Based on these studies, we believe it is important to extract general characteristics from the segments of XinTianYou melodies.

To extract the short melody units, we need to divide the melody into segments based on the semantics of the lyrics, because lyrics and tunes are depend on each other in Chinese folk songs. Most XinTianYou songs contain eight measures which correspond to two sentences. Generally, two measures express as a semantic element. Also we find that normally the singer needs a pause to breathe after singing two or three measures. Based on these observations, we first divide the music piece into segments, each of which contains two measures. Then we manually mark the sentences in the database to make sure the division is within the range of each sentence. The last measure is combined with the one before it when there are an odd number of measures in one sentence. Our method can divide the music piece to segments reasonably. In Figure 1 we show the division of an example song.

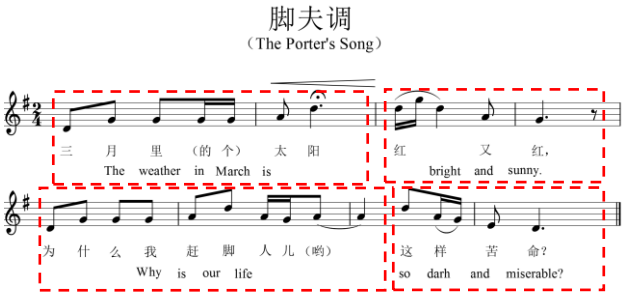


Figure 1. Segmentation of an example song. Each dashed box contains a segment. The lyrics reflect the laboring people's complaining about their miserable life and yearning for happiness. The song can be sung repeatedly with this fix tune structure.

2.2 Mapping

The MIDI data in our database is extracted by using the open source MIDI Toolbox [11]. Notes are represented by hexadecimal numbers from the 00 to 7F. The difference in numerical values between notes corresponds to the number of semitones. Under the naturally symmetric intervals, there is a certain relationship between the number of semitones and the interval distance (without considering the interval property). This relationship appears in all octaves. Next, we summarize the mapping in two cases.

Case 1: Mapping within an octave.

We show the results in Table 1. Here we use "*Diff*" to represent the difference between the two adjacent notes in numerical values and "*Interval*" to represent the corresponding musical distance.

As we can see from Table 1, when the *Diff* is six, there are two options. This is because the augmented 4th and diminished 5th are naturally symmetric intervals. If two notes are from the same register, for example f^1 and b^1, the *Interval* is 4th. But if the two notes are from adjacent registers, for example $b^1$and f^2, the *Interval* is 5th.

Diff	**0**	**1, 2**	**3, 4**	**5, 6**	**6, 7**	**8, 9**	**10,11**	**12**
Interval	unison	2nd	3rd	4th	5th	6th	7th	octave

Table 1. Relationship between *Diff* and *Interval*.

Case 2: Mapping beyond one octave (*Diff>12*).

The interval can be computed by the following Equation (1):

$$\begin{cases} Interval = \mathrm{F}(X_1) + X_2 \times 7 \\ X_1 = \mathrm{Mod}(Diff, 12) \\ X_2 = \mathrm{Fix}(Diff\ /12) \end{cases} \quad (1)$$

where the function Mod() is modulo arithmetic, and the Function Fix() is quotient operation. F() is a discrete function which maps the semitones to interval within one octave, as shown in Table 1. In this way, we can turn MIDI data into musical intervals.

3. N-APRIORI PATTERN MINING

In this section, we introduce an improved Neighbor-Apriori (N-Apriori) algorithm to explore the frequent patterns. We mainly improve the "join step" and the "search step" of the traditional Apriori algorithm. Algorithm 1 shows pseudo-code for the N-Apriori algorithm.

Algorithm 1. N-Apriori algorithm.

Input *D*: Database.
Output {*L*}: Frequent items.

1. Find frequent_1_items L_1 ;
2. **The join step** : candidate *k*-items generation C_k,
3. $C_k = L_{k-1} \times L_1$;
4. **The search step**: Candidate *k*-items adjacent statistics;
5. **The prune step** : frequent *k*-items generation L_k, $C_k \rightarrow L_k$;
6. Repeat 2~5, until there are no more frequent items remaining;
7. Return frequent items {*L*};

Here we explain the process of N-Appriori in detail. Firstly, same as Apriori algorithm in [12], the N-Appriori algorithm scans each transaction in the database, and counts the frequency of each item. Here, a transaction refers to a semantic fragment, and item refers to the interval calculated by the Equation (1). A pattern could be one item or combination of several items. The frequency of each pattern is referred as support. Any pattern with support lower than a minimum support threshold is removed. N-Apriori also employs an iterative approach

known as a level-wise search, where *k*-items are used to explore (*k+1*)-items. However, in the join step, the set of candidate k-item C_k is generated by joining L_{k-1} to L_1, and the identical items are not merged. In the search step, each pattern in C_k is regarded as a whole. In our method, only the patterns with specific context and order are counted, while in the original Apriori algorithm all patterns are considered. These processes are repeated until no more frequent items can be found.

4. MULTI-LAYER MELODY CLUSTERING

Although by using the N-Apriori algorithm we can find correct and reasonable intervals, it is not clear how to find an ideal support threshold. If the threshold is too low, it may lead to a large number of output patterns, while a high threshold may fail to find representative modes. In order to compress the number of frequent patterns and find high-quality melodic framework, Dong et al [13] propose to extract redundancy-aware top-*k* patterns from a large collection of frequent patterns. Their method uses significance and redundancy as criterions.

Inspired by [13], we propose to use a multi-layer melody perception clustering method for extracting top-*k* representative patterns. We use the Cosine Similarity to determine the similarity between melodic contours, and Edit Distance to determine the similarity of melodic amplitude. In order to extract high significance and low redundancy melodic structures from the combinations of intervals, we use Hierarchical Agglomerative Clustering (HAC) method, which has been commonly used for document similarity clustering [19-21]. We apply HAC to the similarity or cost matrix computed by Cosine Similarity and Edit Distance. To evaluate the clustering results, we use the average Silhouette Coefficient (SC) [14] and find the best number of clusters. At last the most significant patterns are selected as the general characteristics of XinTianYou.

4.1 Melodic Direction based Clustering

The tendency of a melodic line includes ascending and descending stages, which can be represented as its melody contour. So the problem of clustering the melodic direction can be converted to the comparison between contours. Cosine Similarity (CS) has been commonly used to solve the problem of matching feature vectors [15-17]. It can be used to solve the problem of matching melodic contours. The CS inherently requires the length of compared strings should be equal. However, it is impossible for the frequent melody contour to be equal length invariably. It can be seen that the framework of N-Apriori algorithm form patterns of different sizes, between which the CS method cannot be directly applied. For the vectors with same length, the CS can be directly computed. For the vectors with different sizes, we calculate the CS between the shorter vector and each sub-vectors of the longer vector, and get a list of corresponding similarity values. We then compute the mean of all the similarity values as the final similarity. After computing the the simlarity matrix, hierarchical clustering method can be applied. The pseudo-code is given in Algorithm 2.

Algorithm 2. Direction based clustering algorithm.

Input L: Frequent pattern sets $L= \{L_1, L_2 \ldots L_n\}$.
Output DCluster: Clustering Result

```
1. CosValue = 0;      //Cosine value
2. MDirection = ∅;  //Similarity matrix
3. For  i = 1: Length(L)
4.    For  j = i : Length(L)
5.      If  length(L_i) == length(L_j)  CosValue = Cos(L_i, L_j);
6.      Else  CosValue = mean(Cos);
                  //Take the average of all cosine values
7.      EndIf
8     MDirection [i, j] = CosValue;
9.    EndFor
10. EndFor
11.  DCluster = HAC_Cluster(MDirection);
                  //Clustering based on the similarity matrix
12. Return DCluster;
```

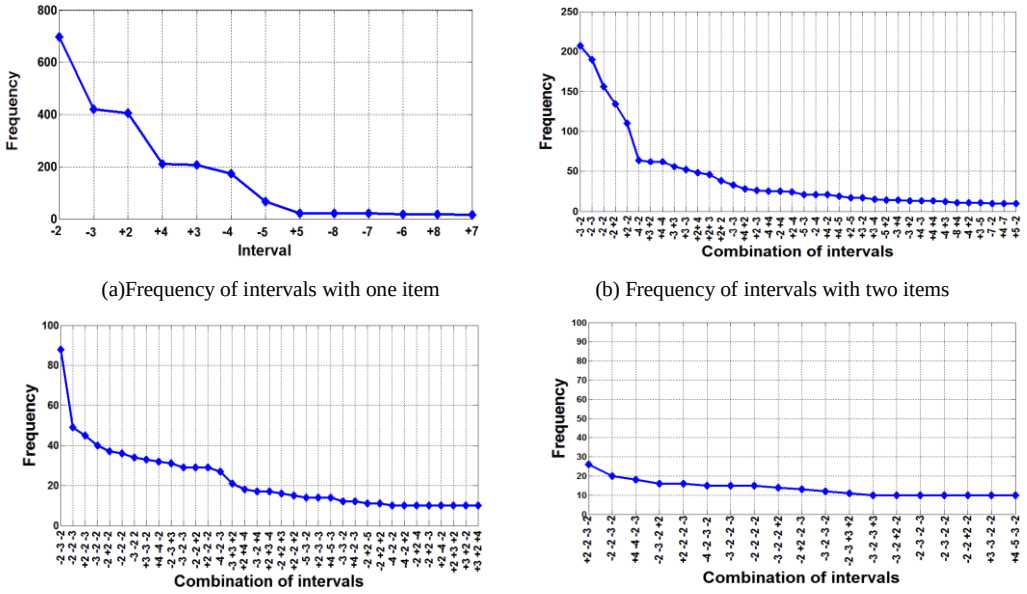
4.2 Melodic Value based Clustering

The values within the combination of intervals represent the amplitude of the melody in a piece of music. In this section we apply Edit Distance (ED) [18] to compare the amplitude values of different melodic lines. The basic idea of ED is to measure the cost of transforming one string to another through a dynamic programming process. The typical transformations include substitution, insertion and deletion. We propose an improved ED method named self-adaptation cost Edit Distance (SAC ED) to compute the distance between two vectors.

Algorithm 3. Computing self-adaptation cost Edit Distance (SAC ED).

Input String *A* and *B*: frequent patterns in each *Dcluster*.
Output *DistValue*: The minimum cost between *A* and *B*.

```
1.  DistMatrix = ∅;
2.  La = Length(A);
3.  Lb = Length(B);
4.  For  p = 0 : La-1
5.     For  q = 0 : Lb-1
6.        If p == 0 && q==0        DistMatrix (p, q) = 0;
7.        Elseif  p == 0 && q > 0    DistMatrit(p, q) = InsCost;
8.        Elseif  p > 0 && q == 0    DistMatrix (p, q) = DelCost;
9.        Elseif  P > 1 && q > 1
10.                 Ins = DistMatrix(p-1, q) + InsCos;
11.                 Rep = DistMatrix (p-1, q-1) + RepCost;
12.                 Del = DistMatrix (p, q-1) + DelCost;
13.              DistMatrix (p,q) = Min{ Ins , Rep , Del };
14.       EndIf
15.    EndFor
16. EndFor
17. DistValue = DistMatrix (La-1, Lb-1);
18. Return  DistValue;
```



(a)Frequency of intervals with one item

(b) Frequency of intervals with two items

(c)Frequency of intervals with three items

(d) Frequency of intervals with four items

Figure 2. Frequency sets computed by N-Apriori

When an interval ***a*** is inserted or deleted, the cost of the transformation is $|\boldsymbol{a}|$, which represents the norm of ***a***. If ***a*** is replaced by ***b***, the cost is the norm of the difference between the two vectors $|\boldsymbol{a}-\boldsymbol{b}|$. Therefore the distance function can be summarized by equation (2).

$$Cost = \begin{cases} |\boldsymbol{a}| \text{ if } \boldsymbol{a} \text{ is insersted} & (InsCost) \\ |\boldsymbol{a}-\boldsymbol{b}| \text{ if } \boldsymbol{a} \text{ is replaced by } \boldsymbol{b} & (RepCost) \\ |\boldsymbol{a}| \text{ if } \boldsymbol{a} \text{ is deleted} & (DelCost) \end{cases} \quad (2)$$

The transformation between two interval vectors is not unique, so there are different distance values. Among these values we choose the minimum one to measure the similarity between two interval vectors. The interval vectors are more similar if the distance value is lower. Computing the minimum SAC ED is a dynamic programming process, which is shown by pseudo-code in Algorithm 3. With SAC ED method, we can compute a cost matrix, based on which we can do the HAC clustering similar to Algorithm 2.

4.3 Evaluation of Clustering Results and Significance Analysis

To evaluate the clustering results, we compute the average Silhouette Coefficient (SC) [14] value of all objects in the cluster. The range of average SC values is [-1, 1]. The larger the average SC is, the higher the clustering quality will be.

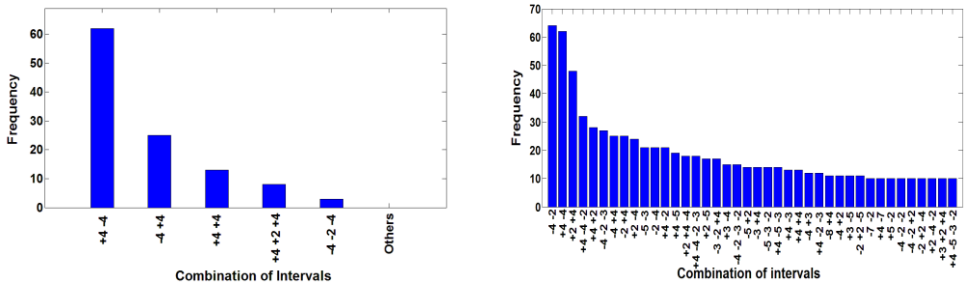
After the clustering, each cluster is relatively independent, but the coherence is high within each cluster. So the most significant pattern is selected from each cluster as the general characteristic of melodies of XinTianYou. In our experiment, we use the "support" introduced in Section 3 to measure the significance of pattern. More specifically, we choose the one with highest support value as the most significant pattern.

5. EXPERIMENTS

5.1 Results of N-Apriori

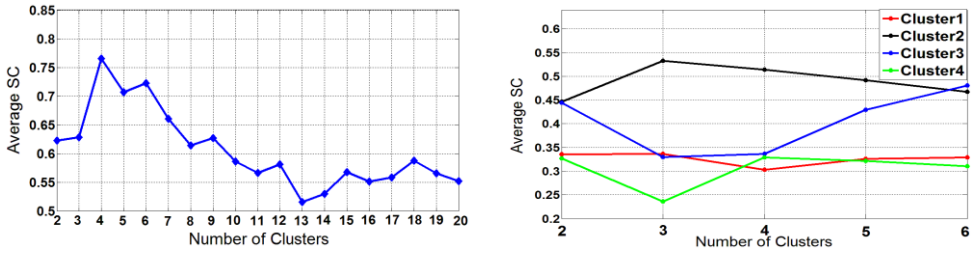
In total we extracted 453 semantic fragments from the 109 songs in our database. The N-Apriori method is applied to these fragments. The support threshold is set to 10 in our experiment. The resulting frequency is shown in Figure 2. The four graphs represent the frequency pattern of intervals with different lengths. The horizontal axis indicates different interval patterns. The sign '+' and '-' in the axis notes denote the melody ascending and descending respectively. The vertical axis indicates the corresponding frequency. Interval of unison is not shown in the results. This is because unison represents two adjacent notes with the same pitch, which cause no changing in the melody trend. The adjacent intervals are merged in our experiment.

From Figure 2-(a) we can see that frequency sets within the interval from -2 to +2 take a large proportion, and the trend slows down from +4 to -4. In Chinese folk songs, intervals no more than 3rd are regarded as narrow intervals [22], and intervals above 3rd are regarded as wide intervals [22]. Usually the narrow intervals appear frequently, however, 4th is dominant in our experiment, which appears around 400 times and takes about 88% of the semantic fragments. Figure 2-(b) shows that for combination of a 2nd and a 3rd, the descending trend appears to be more prominent than the ascending trend. We can also see that symmetric structure appears a lot in the high proportion combinations, such as +2 -2,-2 +2, +3 -3, 3 +3, +4 -4. Figure 2-(c) shows that the structure -2-3-2, is much higher than the second frequent structure -2-2-3. They tend to be evenly distributed after adding 4th. Figure 2-(d) presents an obvious decline in frequency compared to the three previous results. The distribution is relatively stable at the low frequency. Next we analyze



(a)Analysis of double-4th distance (b) Frequency of wide intervals

Figure 3. Deep analysis of wide intervals



(a)Average SC results based on CS (b)Average SC results based on ED

Figure 4. Average SC results

the result from the aspects of the direction trend, pitch selection, and musical structure.

Regarding the direction trend, a narrow structure with a continuous descending is dominant in XinTianYou, while continuous ascending structures rarely happen. The frequency of -3-2 and -2-3 structures verify the "three notes within a 4th" structure theory proposed by Liu [8]. In this paper they regarded the combination of 2nd and 3rd from the same direction as being the chromosome of Chinese folk songs. Moreover, the results show that the chromo some - the top frequent patterns - also includes other combinations of "2nd and 3rd" and "2nd and 2nd".

In the aspect of pitch selection, earlier researchers considered that a notable feature of XinTianYou is the "double-4th" structure which usually in two forms of +4+2+4 and +4+4 [7]. But according to the results in Figure 3, the main 'double-4th' structures are +4-4 and -4+4, rather than +4+2+4 and +4+4 (as shown in Figure 3-(a)). We also found the wide interval frequently appears in various combinations as shown in Figure 3-(b). This makes XinTianYou full of strength and vitality.

Regarding the structure, we found that the symmetric structures are extensively used. This introduces balance into the melody, and adds flexibility without losing harmony. This enables the music to present a sense of destination seeking and provoke deep feelings in the heart. Long melodic lines have various patterns, as they are a flexible mixture of small sections, for which the general characteristics is not very obvious.

5.2 Results of Multi-layer Clustering

In this section, based on the frequency analysis by using the N-Apriori algorithm, we first use the Algorithm 2, and then use Algorithm 3 to do the clustering. Then the average Silhouette Coefficient (SC) is used to evaluate the clustering results.

Figure 4-(a) shows the average SC results under different cluster numbers. It can be seen that using 4 clusters leads to the highest average SC value. So we use 4 clusters in this stage. In Figure 4-(b), we show the average SC for the further clustering of each cluster from last step. The best selected number from cluster 1 to cluster 4 is 3, 3, 6 and 4. So we obtain 16 clusters in total. For each of these 16 clusters, the combination of intervals with the highest support is selected by using the top-*k* method. All these representative combinations form the final general characteristics of XinTianYou. The results show in Table 2. The corresponding melodies are shown in the Appendix.

Cluster	Interval Combination	Cluster	Interval Combination
k=1	-3 -2	k=9	+4 +4
k=2	-2 -3 +3	k=10	+3 +2 +4
k=3	-7 -2	k=11	+3 +2 -2
k=4	-2 +2 -2	k=12	+2 +4 -4
k=5	-3 +3	k=13	+2 -2 -3
k=6	-8 +4	k=14	+4 -4
k=7	+2 +3 +2	k=15	+4 -7
k=8	+2 +4	k=16	+4 -4 -2 -3

Table 2. XinTianYou's general characteristics in melody

To evaluate our results, we asked ten experts to manually mark the resulting musical data. All the participants are professionals in music and familiar with XinTianYou. More specifically, we first choose the most frequent melody segments for each of the 16 clusters and built the first set of data "Group One", and then randomly choose 16 other melodies appear in our database named it "Group Two". The Group One and Group Two are put together. Table 3 shows the melody scores for the 32 melody seg-

ments. All these melody segment scores are shown to the participants in random order. The participants were asked to mark each melody to one of five levels which represent how frequent the melody appears in XinTianYou. Level 1 means not frequent at all, while the level 5 means the most frequent. In Figure 5, we visualize the marks of both the 16 representative melody segments found by our method (in blue) and the randomly chosen noisy data (in red). The horizontal axis lists the segment id and the vertical axis indicates the corresponding mark. The item id in Table 3 is in the same order as the horizontal axis of chart in Figure 5. We also show the average mark for each group by the black dashed lines. From the chart we can see that obviously the 16 melody segments found by our method obtains a much higher mark (4 on average) while the noisy data (Group Two) only get mark 1.75 on average. This means the experts think that the 16 melody segments are much more representative than others, and implies the result of our method is consistent with the experts' perception and professional knowledge.

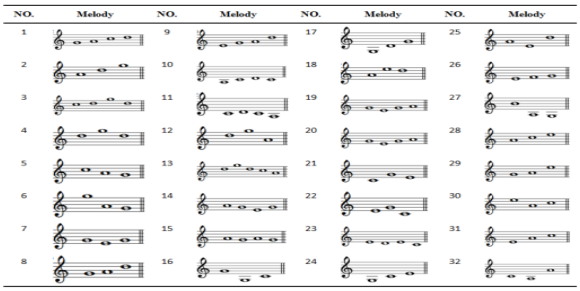


Table 3. Melody segments in the evaluation set.

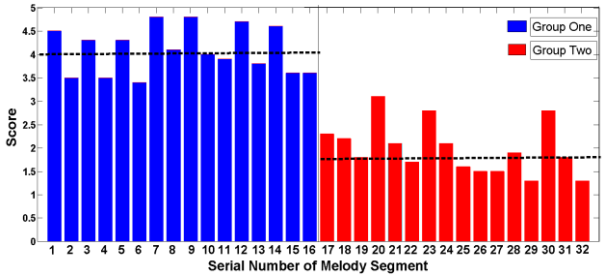


Figure 5. Evaluation results

6. CONCLUSIONS

XinTianYou, an important style of folk songs, has had a profound impact on the musical history of China. The exploration of XinTianYou's general characteristics is important for both automated musical analysis research and obtaining greater insights into Chinese folk music. In this paper, we explore the general characteristics of XinTianYou. Firstly, we built a XinTianYou MIDI database to facilitate future study. Secondly, we propose a new direction of XinTianYou research which examines the music data through its interval combinations (the small melody segments). Thirdly, we propose to use the redundancy-aware top-*k* patterns method to do the clustering. Based on the similarity measure and the support, we finally find the most dominant patterns as the general melodic characteristics.

Each culture has its own evolving heritage. We wonder if these general characteristics are gene of Chinese music. In the future, we can investigate more thoroughly if the general characteristics discovered in this paper are representative across all Chinese folk music.

7. REFERENCES

[1] K. Naoko, et al., "A practical query-by-humming system for a large music database," in Proceedings of the eighth ACM international conference on Multimedia, 2000, pp.333-342.

[2] Y. Li, Y. Wu, and B. Liu, "A new method for approximate matching melody humming retrieval system and application," Computer Research and Development, 2004, pp. 1554-1560.

[3] Y. Yang, and X. H, "Cross-cultural music mood classification: a comparison on English and Chinese Songs," in ISMIR, 2012, pp.19-24.

[4] P.D. León, et al., "Statistical description models for melody analysis and characterization," In International Computer Music Conference, 2004, pp.149--156.

[5] C. McKay and F. Ichiro, "Automatic genre classification using large high-level musical feature sets," in ISMIR, 2004, pp.525-530.

[6] Collins, T., "Improved methods for pattern discovery in music, with applications in automated stylistic composition," Open University, 2011.

[7] National Editorial Board, Chinese folk songs·Shaanxi volume, China ISBN Center, 1994.

[8] Z. Liu, "Constraints of traditional music style of the -- Tricolor theory," Chinese Music, 2014, pp. 59-71.

[9] Y. Wang, "Analysis of the tonal structure in the folk song's melody--take the folk songs of She minority as example (I)," Music Research, 2007(1), pp. 68-74.

[10] Y. Wang, "Analysis of the tonal structure in the folk song's melody--take the folk songs of She minority as example (II)," Music Research, 2007(2), pp. 15-21.

[11] T. Eerola and P. Toiviainen, "MIR in MATLAB: The MIDI Toolbox," in ISMIR, 2004, pp.22-27.

[12] C. Wu, J. Wang, and C. Chen, "Mining condensed rules for associative classification," in Machine Learning and Cybernetics, International Conference on, 2012, pp.1565-1570.

[13] X. Dong, et al., "Extracting redundancy-aware Top-K patterns," in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006, pp.444-453.

[14] Zhao, M., S.J. Turner and W. Cai., "A Data-Driven Crowd Simulation Model Based on Clustering and

Classification, ” In Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications. 2013, pp.125-134.

[15] S. Zhu, J. Wu, and G. Xia, “Top-K cosine similarity interesting pairs search,” Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on, 2010, pp. 1479 - 1483.

[16] S. Zhu, et al., “Top-MATA: A Max-First traversal method for Top-K cosine similarity search,” in Service Systems and Service Management, 2010 7th International Conference on, 2010, pp.994-998.

[17] A. Madylova and S.G. Oguducu, “A taxonomy based semantic similarity of documents using the cosine measure,” 24th Information Symposium on Computer and Information Sciences, 2009, pp. 129-134.

[18] D. Deng, et al., “Top-K string similarity search with edit-distance constraints，” Data Engineering (ICDE), 2013 IEEE 29th International Conference on, 2013, pp. 925 - 936.

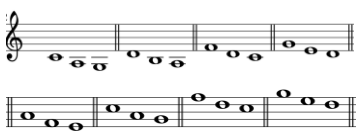
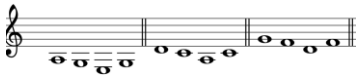
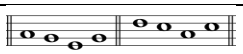
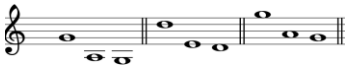
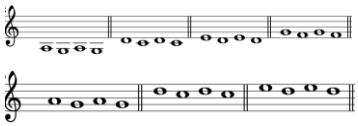
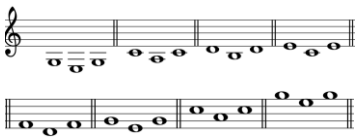
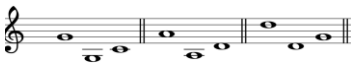
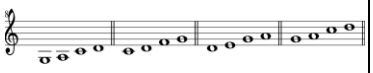
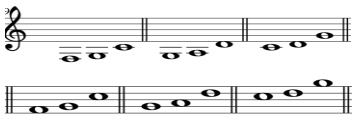
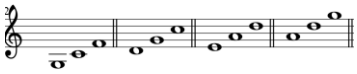
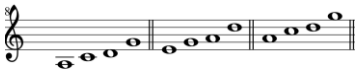
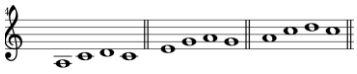
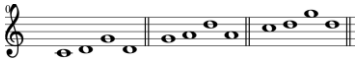
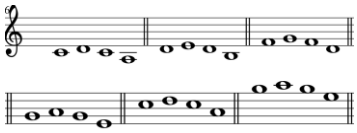
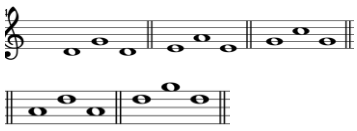
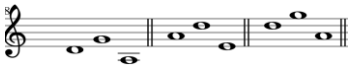
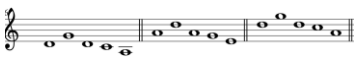
[19] M. Akbar and R.A. Angryk, “Frequent pattern-growth approach for document organization,” in Proceedings of the 2nd international workshop on Ontologies and information systems for the semantic web, 2008, pp.77-82.

[20] N. Gao, et al., “Topic detection based on group average hierarchical clustering,” in Advanced Cloud and Big Data, 2013 International Conference on, 2013, pp.88-92.

[21] S. Gilpin, B. Qian and I. Davidson, “Efficient hierarchical clustering of large high dimensional datasets,” in Proceedings of the 22nd ACM International Conference on information and knowledge management, 2013, pp.1371-1380.

[22] C. Li, Basic Music Theory Teaching Materials，BeiJing: Higher Education Press. 2004

APPENDIX: 16 combinations of intervals and corresponding melodies. First column is the cluster id. Second column is the combination of intervals. Third column is the musical score of corresponding melody segments.

Cluster	Combination	Melodies
k=1	-3 -2	
k=2	-2 -3 +3	
		
k=3	-7 -2	
k=4	-2 +2 -2	
k=5	-3 +3	
k=6	-8 +4	
k=7	+2 +3 +2	
k=8	+2 +4	
k=9	+4 +4	
k=10	+3 +2 +4	
k=11	+3 +2 -2	
k=12	+2 +4 -4	
k=13	+2 -2 -3	
k=14	+4 -4	
k=15	+4 -7	
k=16	+4 -4 -2 -3	

Non-negative Sparse Decomposition of Musical Signal using Pre-trained Dictionary of Feature Vectors of Possible Tones from Different Instruments

Ryo Nomura
Graduate School of Integrated Arts and Sciences
Hiroshima University
m140951@hiroshima-u.ac.jp

Takio Kurita
Department of Information Engineering
Hiroshima University
tkurita@hiroshima-u.ac.jp

ABSTRACT

Decomposition of the musical signal into the signals of the individual instruments is a fundamental task for musical signal processing. This paper proposes a decomposition algorithm of the musical signal based on non-negative sparse estimation. We estimate the coefficients of the linear combination by assuming the feature vector of the given musical signal can be approximated as the linear combination of the elements in the pre-trained dictionary. Since the musical signal is considered as a mixture of tones from several instruments and only a few tones appear at the same time, the coefficients must be non-negative and sparse if the musical signals are represented by non-negative vectors. In this paper we used the feature vector based on the auto correlation functions. The experimental results show that the proposed decomposition method can accurately estimate the tone sequence from the musical signal played using two instruments.

1. INTRODUCTION

Decomposition of the musical signals into the signals of the individual instruments is a fundamental task for musical signal processing. This function is necessary for accurate source identification. Once the sounds of the individual instruments are accurately estimated, we can develop music search engine in which the users can search their favorite songs based on the individual instruments sounds.

Non-negative matrix Factorization (NMF) has attracted the attention of many researchers as a method to decomposing of the musical signals into the signals of individual instruments. NMF decomposes a given non-negative matrix into two non-negative matrices, respectively the dictionary matrix and the activation matrix. The dictionary matrix includes non-negative bases which characterize individual common patterns in the given data set. The activation matrix consists of non-negative coefficients which are used to approximate the non-negative input vectors represented by the given matrix by the linear combinations of the bases represented by the dictionary matrix.

We can apply NMF to various fields if the matrix is non-negative. NMF has already been applied to document analysis [1] and face analysis [2]. Also some applications of NMF to musical signal processing have been already investigated. For example, Kawamoto et al. proposed a method to estimate single sounds from chord sounds by using NMF [3].

Several extensions of NMF have been proposed to apply NMF to musical signal analysis. Kameoka et al. [4] extended the original NMF to "Complex NMF" in which the given matrix was factorized into two complex matrices. Another factorization technique is proposed by Yoshii et al. [5] that is calling Positive Semi-Definite Tensor Factorization (PSDTF). PSDTF can decompose not only non-negative spectrograms but also monaural musical signal directly into instrument sounds. It is worth mentioning that the computational complexity of the PSDTF increases as the cube of the number of the estimating parameters.

Since the musical signals usually include only a few signals of the individual instruments within the short time period, the corresponding coefficients of NMF should be sparse, namely only a few are active and all the other coefficients should be close to zero. The extension of NMF to this direction also have been investigated. P. O. Hoyer [6] introduced the concept of sparseness in NMF. This method is called non-negative sparse coding (NNSC). NNSC uses L1 regularization to make both of the coefficients and bases sparse. By this regularization, it is expected that the two non-negative matrices of NNSC become more sparse than those of the original. S. A. Abdallah at el. [7] applied NNSC to spectral basis decomposition to identify the individual spectrum from the given sequence of short-term Fourier spectra. P. Smaragdis and J. Brown [8]applied NMF with regularization for polyphonic music transcription. Tuomas Virtanen [9]extended NNSC to adopt with the temporal continuity nature of musical signal. Masahiro Nakano at el. [10] proposed nonnegative matrix factorization with markov-chained for time-varying.

NMF simultaneously decomposes a given non-negative matrix into the dictionary matrix and the activate matrix. However it is not difficult to gather the signals of each instruments in advance. It is known that human beings can distinguish the sound of instruments within a music by using prior knowledge on the signals of each instruments. If we can use such prior knowledge for the problem of the decomposition of the musical signals into the signals for the individual instruments, we can construct the dictionary of the signals of the individual instruments in advance. This means that we do not need to estimate the dictionary ma-

Copyright: ©2015 Ryo Nomura 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.

trix and we can simplify the algorithm.

In this paper we propose musical signal decomposition algorithm based on non-negative sparse coding using pre-trained dictionary. For simplicity we use the auto correlation function of the musical signals. It is known that auto correlation function is additive and the auto correlation function of the mixing signals are the sum of the auto correlation functions of each component signals. Since the auto coefficient function has a probability negative value, we transform the value to take only non-negative value. We call the features the non-negative normalized auto correlation vectors (nnACV). The nnACV of the music sound is used as the input of the proposed algorithm. The average of the nnACVs of the signals per instrument is used as the elements of the dictionary. The nnACV of the music sound was approximated as the linear combinations of the vectors in the dictionary. The estimated sparse non-negative coefficients of the linear combinations gives the evidence of the existence of the tone corresponding to the vector in the dictionary.

2. DICTIONARY LEARNING

At first we introduce how to make the dictionary for music decomposition. Since we can gather the sounds of the individual instruments in advance, it is easy to make the dictionary. In this paper, we use the auto correlation functions as features which characterize the sounds of each instrument. Since the normalized auto correlation functions takes the values in the range from -1 to 1, we transform them to non-negative values.

If two signals are uncorrelated to each other, then the auto correlation function of these signals is given by the weighted sum of the auto correlation functions of the two signals, namely the linear combinations of the auto correlation functions of the two signals. Since it is possible to consider that the signals of the individual instruments are roughly independent from each other, we can decompose the auto correlation function of the mixed signals into the sum of the auto correlation functions of the individual signals by estimating the non-negative coefficients of the linear combinations.

2.1 Non-negative Normalized Auto Correlation Functions

Auto correlation is defined as the cross-correlation of a signal with itself. It computes the similarity between observations as a function of the time lag between them. It is one of the fundamental tools for signal processing.

Let $\{x(t)|t = 0, \ldots, T-1\}$ be a musical signal with in a short period from $t = 0$ to $T - 1$. The mean value and the variance of $x(t)$ for all times t are defined as μ_x and σ_x^2. Then the auto correlation function $c(\tau)$ for this signal with time lag τ is given by

$$c(\tau) = \frac{E\big[\big(x(t+\tau) - \mu_x\big)\big(x(t) - \mu_x\big)\big]}{{\sigma_x}^2}. \qquad (1)$$

Consider a mixed signal $\{z(t)|t = 0, \ldots, T-1\}$ which is defined as the weighted sum of two signals $x_1(t)$ and $x_2(t)$ as $z(t) = w_1 x_1(t) + w_2 x_2(t)$. The mean of value and variance of signal $z(t)$ for all times t are defined as μ_z and σ_z^2. Then the auto correlation function of $z(t)$ is given as

$$\begin{aligned} c_z(\tau) &= \frac{E\big[\big(z(t+\tau) - \mu_z\big)\big(z(t) - \mu_z\big)\big]}{{\sigma_z}^2} \\ &= \frac{1}{{\sigma_z}^2} E\big[w_1^2 x_1(t+\tau)x_1(t) \\ &\qquad + w_1 w_2 x_1(t+\tau)x_2(t) \\ &\qquad + w_2 w_1 x_2(t+\tau)x_1(t) \\ &\qquad + w_2^2 x_2(t+\tau)x_2(t)\big] \qquad (2) \end{aligned}$$

If the cross correlation between $x_1(t)$ and $x_2(t)$ is 0, the auto correlation function of $z(t)$ becomes

$$\begin{aligned} c_z(\tau) &\simeq \frac{1}{{\sigma_z}^2} E\big[w_1^2 x_1(t+\tau)x_1(t) \\ &\qquad + w_2^2 x_2(t+\tau)x_2(t)\big] \\ &= \frac{E\big[w_1^2 c_1(\tau) + w_2^2 c_2(\tau)\big]}{{\sigma_z}^2}, \qquad (3) \end{aligned}$$

where $c_1(\tau)$ and $c_2(\tau)$ are the auto correlation functions of the signals $x_1(t)$ and $x_2(t)$ respectively. This means that the auto correlation function of the weighted sum of two signals is approximately equal to the weighted sum of the auto correlation functions of each signal. If the cross correlation between two signals is not 0, it is impossible to estimate the weight of these two signals. Because of this, it is desirable that the two signals is independently from each other for estimating the weight these two signals from the mixed signal.

The auto correlation function takes both negative or positive values. If we normalize the auto correlation functions by $c(0)$ as

$$nc(\tau) = \frac{c(\tau)}{c(0)}, \qquad (4)$$

the normalized auto correlation functions take the values in the range $[-1, 1]$. From this normalized auto correlation functions we can define the non-negative features as

$$y(\tau) = \frac{nc(\tau) + 1}{2}. \qquad (5)$$

Then the non-negative normalized auto correlation features take the values in the range $[0, 1]$. We use a set of these features as the feature vector of the non-negative sparse coding which is given as $\boldsymbol{y} = (y(1), \ldots, y(L))^T$. We call this vector non-negative normalized auto correlation vector (nnACV).

2.2 Dictionary Learning

Usually it is not difficult to gather the signals of the individual instruments. For example, it is easy to record the sounds of the target instrument. From the recorded signals of each tone of the target instruments, we calculate the nnACVs to make a dictionary for decomposing musical signals.

Consider the case where we have N different tones from several instruments and we record the sounds of each tone

M times. We denote the nnACVs calculated from the recorded sounds by $\{\boldsymbol{y}_i^{(j)} | i = 1, \ldots, M; j = 1, \ldots, N\}$, where i denotes the i-th recorded sound of the tone of the target instruments and j is the index of the tones from different instruments.

For simplicity, we use the average of the nnACVs of each tone

$$\boldsymbol{d}^{(j)} = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{y}_i^{(j)} \tag{6}$$

as the representative base vector of each tone. Then the dictionary matrix D is given by

$$D = \left(\boldsymbol{d}^{(1)}, \ldots, \boldsymbol{d}^{(N)} \right). \tag{7}$$

Since the dimension of the nnACV is L, the dimension of the dictionary matrix D is $L \times N$.

3. DECOMPOSITION OF MUSICAL SIGNAL

In this section we introduce how to estimate the non-negative sparse coefficients of each element in the pre-trained dictionary from a given music sound.

The music sound has melodies, harmony, and rhythm. These factors are composed as the combination of the instruments. The music sounds can be observed as a mixture of sounds from all instruments. Since we are considering to use the nnACV as the feature vector of the given music sound, the estimated coefficients must be also non-negative.

A lot of tones from different instruments appear in music, but only a few tones are played simultaneously. For examples, Piano has eighty eight keys but we use less than 10 keyboards at the same time. This means that almost all coefficients should be zero and only a few have non-zero values, namely the estimated coefficients must be sparse.

Thus we propose the decomposition method in which the non-negative sparse coefficients are estimated while the nnACV of the given music sound is approximated as the linear combinations of the vectors in the pre-trained dictionary.

3.1 Decomposition by Non-negative Sparse Coding

Let $\{x(t) | t = 0, ..., T-1\}$ be the musical signal recorded from the given music sound. From this signal we calculate the nnACV $\boldsymbol{y}$. We want to decompose this nnACV vector as

$$\boldsymbol{y} \approx \sum_{j=1}^{N} w_j \boldsymbol{d}^{(j)} = D\boldsymbol{w}, \tag{8}$$

where $\boldsymbol{w} = (w_1, \ldots, w_N)^T$ is the vector of the non-negative coefficients and each element of this vector indicates the degree of the presence of the corresponding tone in the given music sound.

To evaluate the goodness of the approximation we use the squared errors between the nnACV of the musical signal and the vector estimated by the linear combination of the pre-trained dictionary. The squared errors is defined as

$$E(\boldsymbol{w}) = \|\boldsymbol{y} - D\boldsymbol{w}\|^2. \tag{9}$$

Data: $\boldsymbol{y}$ and D
Result: $\boldsymbol{w}$
Initialize $\boldsymbol{w}^0$ to a random positive vector. Set $h = 0$.;
repeat
  $\boldsymbol{w}^{h+1} = \boldsymbol{w}^h . * (D^T \boldsymbol{y} ./ (D^T D \boldsymbol{w}^h + \lambda))$;
  Increment h;
until $\boldsymbol{w}^h$ *converges*;

Algorithm 1: Algorithm to estimate the non-negative sparse coefficients $\boldsymbol{w}$.

To make the estimated coefficients sparse, we introduce the L1 regularization and consider the cost function of the optimization as

$$Q(\boldsymbol{w}) = E(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_{L1}. \tag{10}$$

Then the optimization problem for decomposition of the music sound is given as

$$\min_{\boldsymbol{w}} Q(\boldsymbol{w}) \quad \text{with constraints} \quad \boldsymbol{y}, \boldsymbol{w}, D \geq 0. \tag{11}$$

This optimization problem is similar to the one that appeared in a single column non-negative sparse coding [6]. An efficient iterative algorithm to solve this optimization is shown by P.O.Hoyer as the algorithm to estimate the activation matrix W for the non-negative sparse coding. We are only estimating the coefficients vector $\boldsymbol{w}$ because the dictionary matrix D is trained in advance. On the other hand, both the basis matrix D and the activation matrix W are estimated in the original non-negative sparse coding.

The update rule to estimate the non-negative sparse coefficients vector $\boldsymbol{w}$ is given as

$$\boldsymbol{w}^{h+1} = \boldsymbol{w}^h . * (D^T \boldsymbol{y} ./ (D^T D \boldsymbol{w}^h + \lambda)), \tag{12}$$

where h is the counter of the iterations. The operators $.*$ and $./$ are the element-wise product and division respectively.

Thus we can summarize the algorithm to estimate the non-negative sparse coefficients vector in Algorithm 1.

4. EXPERIMENT

To confirm the effectiveness of the proposed decomposition algorithm, we have performed some experiments using the sounds generated using a MIDI sound module.

4.1 Data set

We created the music data by using MIDI sound source "KONTAKT5" released from Native Instrument. We used the sounds generated from two instruments. One is Contra-bass and the other is Alto-saxophone. The sound source type of Contra-bass in "KONTAKT5" was set to "Jazz Upright" in "Upright Bass". "Band" was used as the type of Alto-saxophone. The other parameters were set to the default values. To play these sounds, MIDI sequencer "Domino" released from TAKABO Soft was used.

We connected the MIDI sequencer and the MIDI sound source by the virtual loop-back MIDI port "loopMIDI" released by Tobias Erichse. The sounds generated by the

MIDI sound source were recorded with linear pulse code modulation (LPCM) by "Audacity" and they were stored in the files with WAV format. Windows Audio Sound API (WASAPI) was used to record these sounds. The sampling-rate is set to 44100 Hz and the bit rate is 16 bits. The details of these data are shown in table1 and 2.

4.1.1 Training Data Sets for Dictionary Learning

We prepared a set of sound signals $\{\boldsymbol{s}_l^{(j)}|j=1,\ldots,24; l=1,\ldots,3\}$ from 24 different tones from two instruments (alto-saxophone and contrabass). The sound signals were generated from 12 different notes for both alto-saxophone and contrabass. The compass of each instrument was set from $C4$ to $B4$ for alto-saxophone and from $G1$ to $G\flat 2$ for contrabass. To introduce the variations of the sounds, we generated the sound with 3 different values of the velocity parameter, which means the loudness parameters in MIDI $(30, 60, 120)$. Then the value of velocity varies from 0 to 127. The length of the generated sounds in real time is about 3 seconds. We put the silent periods at the beginning and at the end for 0.5 second. So the total length of these sound signals becomes about 4 seconds. The length of each signal U was determined by the sampling rate Fs and the length of the signals in real time Ts as $U = Fs \times Ts$.

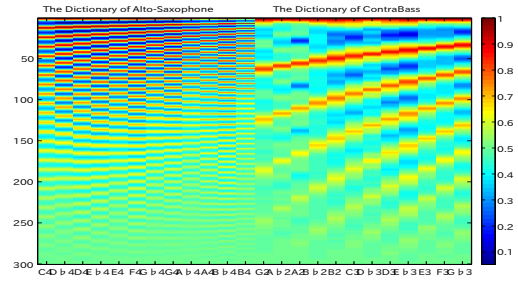
To prepare the training samples for dictionary, We cut these signals by using the rectangular window with the length T. Then the obtained signals are given as $\{\boldsymbol{x}_i^{(j)} = (x_i^{(j)}(0),\ldots,x_i^{(j)}(T-1))^T|i=1,\ldots,M; j=1,\ldots,24\}$. The number of samples of each tone M was determined depending on the sampling rate Fs, the window size T, the window shift size K and the length of the original signals as $M = 3 \times (\frac{U-T-1}{K} - 1)$. Then the nnACV $\boldsymbol{y}_i^{(j)}$ was calculated from each signal $\boldsymbol{x}_i^{(j)}$. Finally the set of the dictionary vectors $\{\boldsymbol{d}^{(j)}|j=1,\ldots,24\}$ were obtained by using the equation 6 and the dictionary matrix D was created.

Figure 1 visualizes the dictionary matrices which are calculated from the sound signals with the sampling rates 3 kHz and 16 kHz. The first 12 columns are the feature vectors calculated from the alto-saxophone and the last 12 columns are from the contrabass.

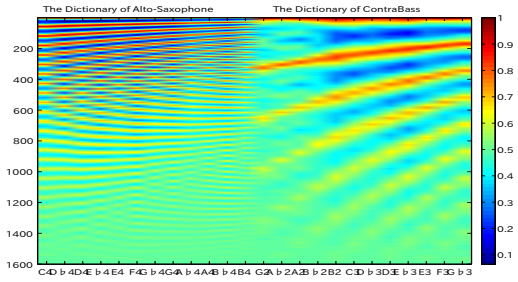
4.1.2 Music Data for Test

We composed a jazz themed music to be used as a test sound signal which is defined as $\{\boldsymbol{s}_t\}$. The tempo of the music is 120 bpm. The note value of the music is a triplet to a double half-note. Only a double note is used in contrabass part. The key of this music is C major. The chord progression is Cmaj7, A7th, Dm7 and G7th. Figure 2 shows the MIDI sequences of the alto-saxophone part and the contrabass part.

From this test signal $\boldsymbol{s}_t$, we calculated the sequence of the non-negative normalized auto-correlation feature vectors $\{\boldsymbol{y}_i|i=1,\ldots,I\}$. Here we used the same parameters (the sampling rate, auto-correlation lags, window size and window shift size) with the training samples.



(a) Dictionary created with the sampling rate 3 kHz



(b) Dictionary created with the sampling rate 16 kHz

Figure 1. Visualization of Dictionary matrix with pseudo color.These figures show that there are few differences between the two different sampling rates that we used (3 kHz, 16 kHz) under the same real time conditions.

	Alto-Saxophone	ContraBass
MIDI sound source	KONTAKT5	
MIDI sequencer	Domino	
Recording Soft	Audacity	
Sampling-rate	44100[Hz]	
pitch range	C4 - B4	G2 - G♭3
Bit	16[bits]	
Velocity	30, 60, 120	

Table 1. The basic information of the training data.

4.2 Selection of the Sampling Rate

As shown in Section 3, the approximation of the input nnACV by the linear combinations of the basis vectors in the dictionary is possible when the cross-correlations between the basis vectors become zero.

We prepared the dictionary $D = \{\boldsymbol{d}^{(j)}|j=1,\ldots,24\}$ using the sound signal with the 3 kHz and 16 kHz sampling rates. Then the cross-correlations between all possible pairs of the elements in D were calculated as

$$\rho_{kl} = \frac{\boldsymbol{d}^{(k)T}\boldsymbol{d}^{(l)}}{||\boldsymbol{d}^{(k)}||||\boldsymbol{d}^{(l)}||} \quad (k,l=1,\ldots,24). \tag{13}$$

The figure 3 visualizes the pairwise cross-correlations $R = [\rho_{kl}]$ with pseudo colors.

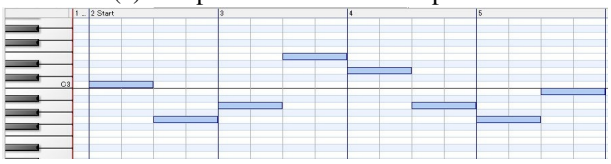
From these figures, we can notice that the cross-correlations between the nnACVs calculated from the alto-saxophone are almost zero and the cross-correlations between the alto-saxophone and the contrabass are also almost zero too. While they have some values between the neighboring notes

	Alto-Saxophone	ContraBass
MIDI sound source	KONTAKT5	
MIDI sequencer	Domino	
Recording Soft	Audacity	
Sampling-rate	44100[Hz]	
pitch range	C4 - B4	G2 - G♭3
Bit	16[bits]	
Velocity	61-97	50
Beat Per Minutes	120[bpm]	
Rhythm	2 beats	
The number of bars	4 bars	

Table 2. The basic information of music data.



(a) The piano roll of alto-saxophone



(b) The piano roll of contrabass

Figure 2. MIDI sequences.

of the contrabass. We also calculated the ratio of the sum of diagonal elements to the sum of all auto-correlations for two instruments by equation 14.They are summarized in the table 3.

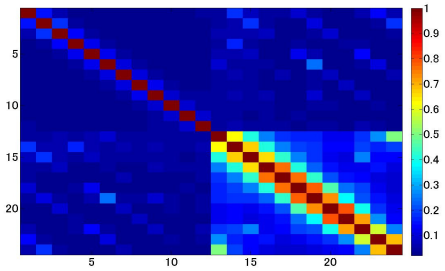
$$Idp = \frac{\sum_i^N |\rho_{ii}|}{\sum_{k=1}^N \sum_{l=1}^N |\rho_{kl}|} \leq 1. \quad (14)$$

Where Idp is the ratio of independence of dictionary. When Idp of the dictionary is 1, it is said that this dictionary is ideal independence.

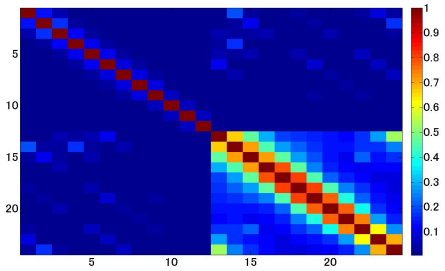
From this table, the sampling rate has minor effect on the ratio of the sum of diagonal elements to all auto-correlations in the alto-saxophone. On the other hand, the sampling rate does not effect that ratio in the contrabass case at all. This difference is probably caused by the harmonic structure in high frequency. Alto-saxophone has some harmonic structure in high frequency area, thus higher sampling rate can capture these structure correctly and give a better score. There is no improvement for contrabass because the harmonic structure of the contrabass is in the low frequency

	3[kHz], $L = 300$	16[kHz], $L = 1600$
Alto-sax	0.6717	0.7619
Contrabass	0.2363	0.2304
Mixed	0.2901	0.3235

Table 3. The ratio of the sum of diagonal elements to the total sum of all cross-correlations.



(a) Calculated using the vectors in the dictionary with the sampling rate 3 kHz



(b) Calculated using the vectors in the dictionary with the sampling rate 16 kHz

Figure 3. Visualization of the pairwise cross-correlations between the vectors in the learned dictionary.

area only.

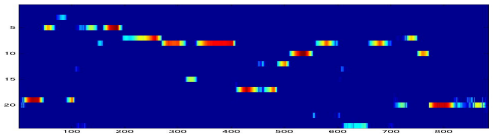
We use 3 kHz down sampled signals for the following experiments because the differences with the sampling rates are not so large.

4.3 Decomposition of Music Sound

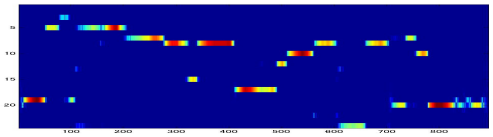
To confirm the effectiveness of the proposed decomposition algorithm, we applied the proposed non-negative sparse decomposition algorithm to the prepared music sound. We performed the decomposition experiments with three different lengths ($T = 600, 900, 1200$). If T is set to a small value, the window size becomes short and we can get more samples form the prepared music sound. We got 1133, 1123, and 903 samples from the cases of $T = 600$, $T = 900$, and $T = 1200$ respectively.

The sparseness parameter λ was changed from 0 to 400 with an incremental step size of 10. If the number of iterations of the proposed decomposition algorithm exceeds 500 times without convergence, the algorithm stops.

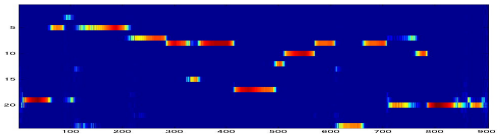
Figure 4 shows some results of the decomposition by the proposed algorithm. The alto-saxophone part is shown in the upper half of each figure and the contrabass part is shown in the lower half. The x-axis is the sequence number of the extracted signals marked by the rectangle window. For each extracted signal, we applied the proposed algorithm and then the coefficients of the vectors in the pretrained dictionary were estimated. The values of the estimated coefficients are shown in the row as pseudo color. Since the dictionary was constructed by using the sounds of each note from each instruments, the sequence of the estimated coefficients shown in Figure 4 should be the same



(a) $L = 600,\ \lambda = 150$



(b) $L = 900,\ \lambda = 220$



(c) $L = 1200,\ \lambda = 270$

Figure 4. Decomposition results of the test music.These figures visualize the coefficients of the dictionary components with psuedo color. The alto-saxophone part is shown in the upper half of the each figure and the contrabass part is shown in the lower half. The red color means the value of coefficient marks high, on the other hand, the blue color means that marks low.

as the MIDI sequence. This means that the coefficients from the first row to the 12th row are represented as $B4$, $B\flat 4, \ldots, C4$ in alto-saxophone and from 13th to 24th row are represented as $G\flat 3, F3, \ldots, G2$ in contrabass.

From these results we can say that the proposed decomposition algorithm can decompose the music sound played with two instruments into the notes of each instruments almost correctly. Since the cross-correlations between the vectors of alto-saxophone in the dictionary are almost zero, the accuracy of the alto-saxophone part is better while some mistakes have occurred at the contrabass part.

To improve the accuracy, we have to develop an algorithm to construct the uncorrelated vectors in the dictionary.

5. CONCLUSION

This paper proposed a musical signal decomposition algorithm based on non-negative sparse coding using pretrained dictionary. For simplicity we used the non-negative normalized auto correlation vector (nnACV) of the musical signals as the input of the proposed algorithm. The average values of the nnACVs of the signals of the individual instruments were used as elements of the dictionary. The nnACV of the music sound was approximated as the linear combinations of the vectors in the dictionary. The estimated sparse non-negative coefficients of the linear combinations give the evidence of the existence of the tone corresponding to the vector in the dictionary. We confirmed that the proposed algorithm could appropriately decompose the music sound generated using two instruments through the experiments.

Since the auto-correlation vectors in the dictionary are highly dimensional and the cross-correlations between them are not completely uncorrelated, it is necessary to develop an algorithm to construct uncorrelated dictionary with reduced dimension. This is one of our future works.

Acknowledgments

This work was partially supported by KAKENHI(23500211).

6. REFERENCES

[1] W. Xu, X. Liu, and Y.Gong, "Document clustering based on nonnegative matrix factorization," in *Proc. ACM SIGIR*, 2003, pp. 267–273.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[3] T. Kawamoto, K. Hotta, T. Mishima, J. Fujiki, M. Tanaka, and T. Kurita, "Estimation of single tones from chord sounds using non-negative matrix factorization," *Neural Network World*, vol. 10, no. 3, pp. 429–436, 2000.

[4] H. Kameoka, T. Nishimoto, and S. Sagayama, "Complex nmf: A new sparse representation for acoustic signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*, 2009, pp. 45–48.

[5] K. Yoshii, R. Tomioka, D. Mochihashi, and M. Goto, "Infinite positive semidefinite tensor factorization for source separation of mixture signals," in *International conference on machine learning 2013*, 2013.

[6] P. O. Hoyer, "Non-negative sparse coding," in *Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing)*, 2002, pp. 557–565.

[7] S. A. Abdallah and M. D. Plumbley, "Unsupervised analysis of polyphonic music using sparse coding," *IEEE Trans. on Neural Networks*, vol. 17, no. 1, pp. 179–196, 2006.

[8] P. Smaragdis and J. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics 2003*, 2003, pp. 177–180.

[9] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transaction on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, 2007.

[10] M. Nakano, J. L. Roux, H. Kameoka, Y. Kitano, N. Ono, and S. Sagayama, "Nonnegative matrix factorization with markov-chained bases for modeling time-varying patterns in music spectrograms," in *Latent Variable Analysis and Signal Separation*, 2010, pp. 149–156.

Sensor and Software Technologies for Lip Pressure Measurements in Trumpet and Cornet Playing - from Lab to Classroom

T. Grosshauser, G. Troester
ETH Zurich
Electronics Lab
grotobia@ethz.ch
gerhartr@ethz.ch

A. Thul
Alte Kantonsschule Aarau
anuschka.thul@altekanti.ch

M. Bertsch
University of Music, Vienna
bertsch@mdw.ac.at

ABSTRACT

Several technologies to measure lip pressure during brass instrument playing have already been developed as prototypes. This paper presents many technological improvements of previous methods and its optimization to use this technique as "easy to handle" tool in the classroom. It also offers new options for performance science studies gathering many intra- and inter-individual variabilities of playing parameters. Improvements include a wireless sensor setup to measure lip pressure in trumpet and cornet playing and to capture the orientation and motion of the instrument. Lightweight design and simple fixation allow to perform with a minimum of alteration of the playing conditions. Wireless connectivity to mobile devices is introduced for specific data logging. The app includes features like data recording, visualization, real-time feedback and server connectivity or other data sharing possibilities. Furthermore, a calibration method for the sensor setup is developed and the results showed measurement accuracy of less than 5 % deviation and measurement range from 0.6 N up to a peak load to 70 N. A pilot study with 9 participants (beginners, advanced students and a professional player) confirmed practical usage. The integration of these real-time data visualizations into daily teaching and practicing could be just the next small step. Lip pressure forces are not only extremely critical for the upper register of the brass instruments, they are in general crucial for all brass instruments, especially playing in upper registers. Small changes of the fitting permit the use of the sensor for all brass instruments.

1. INTRODUCTION

In music performance research and analysis, more and more sensor technologies were added to mainly audio and video based systems (see Ng et al. in [1], [2] and Grosshauser et al. in [3]). In this paper a further developed sensor setup for trumpet and cornet lip pressure measurement is introduced similar to Bertsch et al. in [4], Mayer et al. in [5], Petiot in [6] and Grosshauser et al. in [7]. The used sensors are load cells to measure lip pressure with three triangular arranged measurement points to measure the overall lip pressure and the direction of the pressure. A 9 Degree of Freedom (9 DOF) Inertial Measurement Unit (IMU) is integrated to measure three axes of acceleration and the 3d orientation of the instrument. All sensors are integrated in a module of 60 gr of weight, which is fixed on the mouthpiece and the instrument (see fig. 1 and fig. 3). The sensor module further contains a small printed circuit board (PCB) and is connected wirelessly via Bluetooth Low Energy (BLE) to a mobile device running an app for data

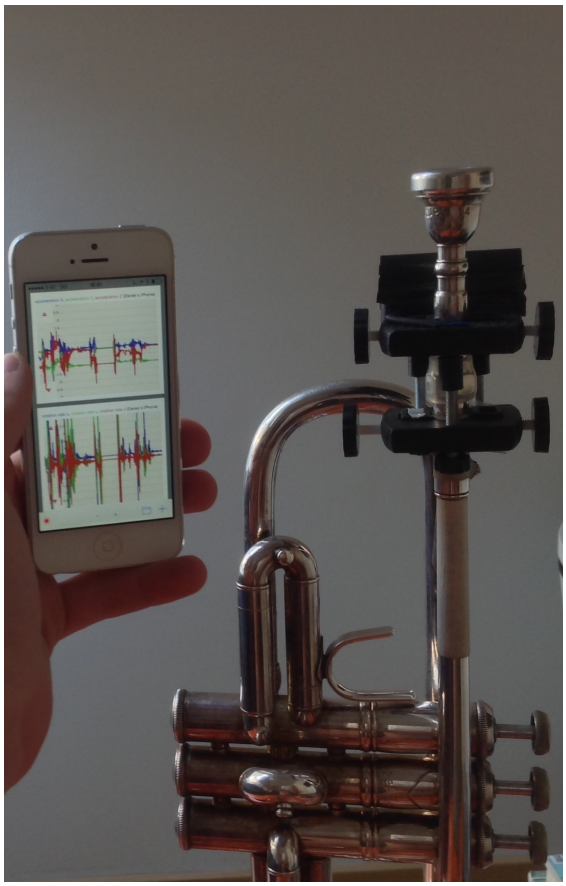


Figure 1. This figure shows the sensor module fixed on a trumpet and the data logging app running on a mobile phone. The first row shows the orientation data, the second row the 3 pressure data of each sensor.

Copyright: ©2015 T. Grosshauser, G. Troester 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.

recording, visualization, server upload and real-time feedback (see fig. 1). In an evaluation with 9 trumpet and cornet players, the module was tested regarding usability and the influence on musicians using the module while playing the sensor equipped trumpet and cornet.

Beside the technical features and the development of a calibration routine, one important point was to simplify the usage and installation of the complete system for simple integration into daily teaching and practicing scenarios. On the other side, the precision of the sensors and possibilities of extensive data calculation on the server side opens many possibilities in scientific experiments and evaluations. Furthermore the real-time possibilities allow applications in augmented instruments and experimental music.

2. TECHNICAL DESCRIPTION OF THE SETUP

Beside measurement accuracy the basic requirement was simple usage and usability to allow fluent integration into daily practicing and teaching scenarios. To reach this goal, the connection between sensor module and mobile device is established automatically after choosing the sensor in the app. After the module is connected the received sensor data are visualized in the app (see fig. 1). The data stream can be recorded, uploaded to a server or thresholds for real-time feedback can be set by simply dragging the line to the needed values. Several more features are available within this app to extend the range of applications and to simplify daily usage.

2.1 The Data Logging App

The app shows all available sensors and sensor channels. The individual set of channels needed for the experiments or measurements e.g. orientation only, pressure and orientation together, etc. are chosen by clicking on them. The connection is established automatically. In a second step, different visualizations can be selected and if necessary, thresholds for real-time feedback can be adjusted. The data streams of each selected sensor is recorded in a file in *.csv file format to allow further calculations in standard statistic programs. Additionally, sharing the recorded data via email, online storage service providers or server upload is possible. If the data are uploaded to a server, a newly developed web based interface provides multi-modal online data management, alignment, e.g. with audio or video recordings and data annotation.

2.2 Sensor Module

After numerous prototypes of the sensor module and the app, the final setup was completed. The sensor module consists of three miniature load cells and one 9 DOF sensor. For power supply a 3V battery is used (see no. 4a in fig. 3. In the housing (see no. 5a in fig. 3) a small PCB is integrated equipped with a BLE module for data transmission, 16bit analog to digital converters (ADCs), voltage regulators and a 9 DOF IMU.

The sensor module (see fig. 3), consists of a 9 DOF IMU with 3 axes accelerometer, 3 axes gyroscopes, 3 axes magnetometer. Based on the IMU data, yaw, pitch roll and w, x, y, z quaternions are calculated. This allows conclusions about the position and orientation and the acceleration of the instrument in all 3 dimensions while playing. The sampling frequency is up to 100 Hz. The final force resolution is below 1 gr, or below 0.01 N.

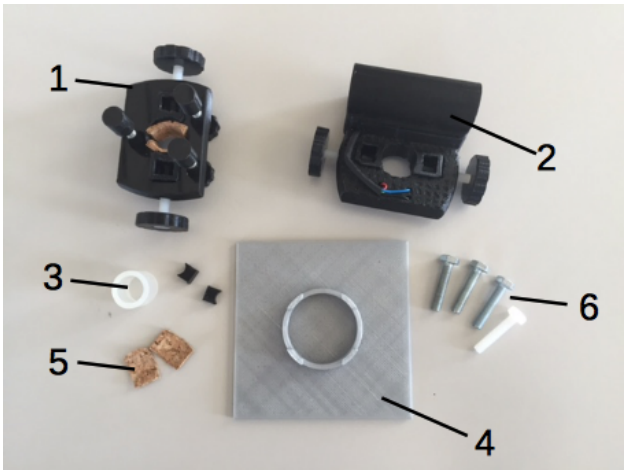


Figure 2. This figure shows the single parts of the sensor module. The part numbers are the same as in fig. 3. No. 1 is fixed on the trumpet and includes the adjustable counter-contact points for the force sensors, included in part no. 2. This part further comprises the batterie, BLE and 9 DOF IMU. Part no. 4 is an adaptor for sensor calibration and no. 3 seals the small gap between mouthpiece and trumpet. Parts 5 and 6 are further spare parts for better adjustment for specific trumpets and cornets.

2.3 Setting Up the Sensor Module

The module itself consists of 2 parts, one part including the PCB (no. 2 in fig. 2) with Battery, analog digital converters (ADCs), BLE module and the force sensors, the counter part (no. 1 in fig. 2 and no. 4a and 5a in fig. 3) with 3 adjustable contact points for load transmission. Part no. 1 and 2 in fig. 2 and fig. 3 are fixed to the mouthpiece and the instrument. The intersection between the mouthpiece and the trumpet is covered with part no. 3 in fig. 2 and fig. 3, a flexible silicon tube. The three screws no. 4 are inserted into part no. 1. These screws push part no. 1 and 2 apart from each other by touching the tip of the miniature load cells. By doing so, the mouthpiece is pushed around 1 mm out of the trumpet and the lip pressure is transmitted to the load cells only.

2.4 Calibration of the Setup

To calibrate the force sensors, on the mouthpiece of the trumpet an adapter is fixed (see fig. 4). The trumpet with the attached adapter is put on a precision scale. Different weights are put onto the adapter and the sensor data are correlated to the scale read outs. The used weights range from 50 gr up to 7 kg. The overall accuracy is below 5 % deviation. The deviation is mainly caused by the silicon tube between mouthpiece and trumpet. The setup is able to measure up to 7 kg.

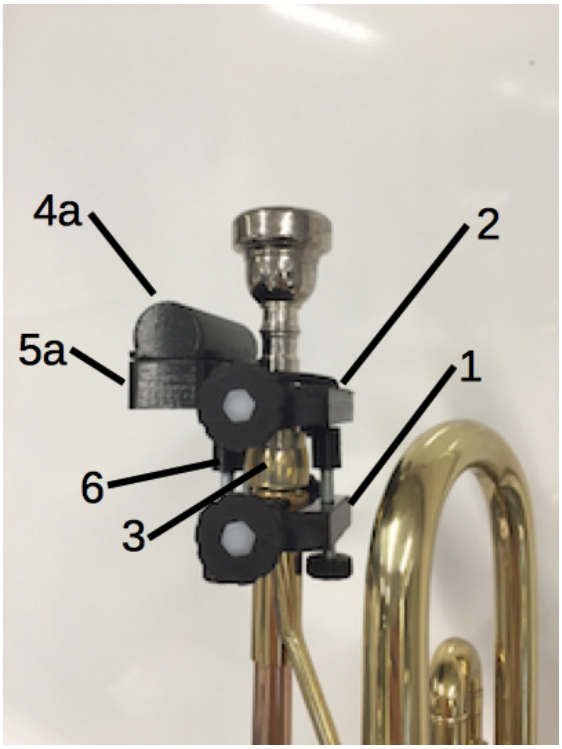


Figure 3. This figure shows the final sensor module fixed on a trumpet. The part numbers are the same as in fig. 2. 1 is the lower fixation part housing the three adjustable screws, touching the three load cells placed in part 2. These are the 3 contact points for force transmission. Between mouthpiece and trumpet there is a 1 mm gap (the mouthpiece is pulled around 1 mm out of the trumpet from the normal fixed position), bridged by 3, a silicon tube. Module 2 includes the PCB (no. 5a) with three miniature load cells, a 10 DOF IMU providing yaw, pitch roll and w, x, y, z quaternions and raw data of the 3 axes accelerometer, 3 axes gyroscopes, 3 axes magnetometer, Bluetooth based data transmission module and a battery (no. 4a).

3. EVALUATION OF THE SETUP AND RESULTS

In the following sections the evaluation is described. In this evaluation, nine students of different level, age and sex played a certain sequence of notes (see fig. 5) with their own instrument, each equipped with the sensor module. They all played in the same room, one after the other. After a short instruction, their instrument was prepared and after a warm up phase, the given phrases were played and finally a questionnaire was filled out by each participant.

The statistical data of the participants are shown in table 1 and the measurement results in fig. 6. The complete experiment took about 20 min per test subject. All subjects played in the same tempo at 60 bpm.

The sensor fits trumpets, cornets, Flugelhorn either with piston or rotary valves. None of the participants felt hindered by the sensor setup while playing and also the played sequences were "easy" or at least "OK" according to their self- assessment. A typical pressure curve is shown in fig. 6 (professional player) and in fig. 7 (amateur player). A tendency of increasing lip pressure of higher notes is clearly

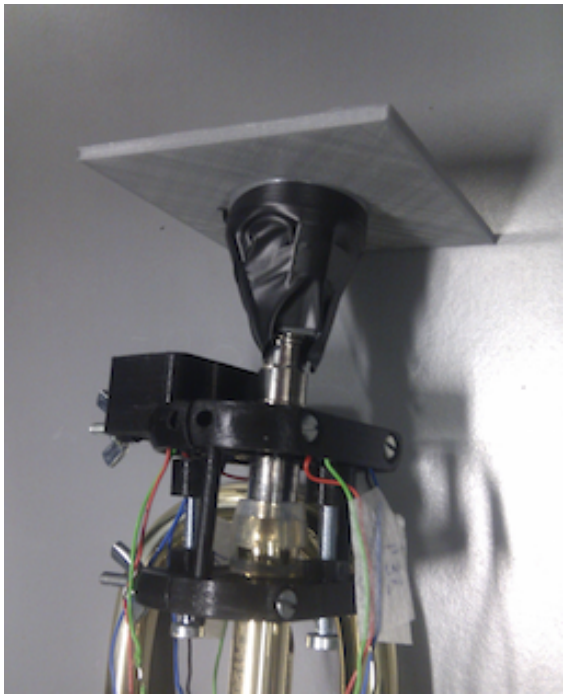


Figure 4. This figure shows an adapter (see also part no. 4 in fig. 2) mounted on the mouth pice of the trumpet to calibrate the sensor setup with defined weights. The sensor is an early prototype of the final setup shown in fig. 3. The trumpet is placed on a highly precise scale. The sensor is calibrated and the final accuracy is specified by putting different weights between 50 gr and 7 kg on the adapter fixed on the mouthpiece.

No.	Age	Exp.	Sex	Instr.	Interfer.	Level
1	15	8	m	Cornet	no	easy
2	17	8	m	Cornet	no	easy
3	18	9	f	Cornet	no	OK
4	29	21	f	Trump.	no	OK
5	17	10	m	Trump.	no	OK
6	19	14	m	Trump.	no	easy
7	17	7	m	Trump.	no	easy
8	18	11	m	Trump.	no	OK
9	21	12	m	Trump.	no	easy

Table 1. The table shows the statistical data of the 9 participants of this study. Participant no. 4 was one professional player with a German trumpet model, the others were students (participants no. 5–9) with a piston valve trumpet model or cornet had a playing experience between 7 and 21 years. The self estimation of the level of the played sequence (see fig. 5) was "easy" or "OK" for all participants and no one felt hindered by the setup (interference "no").

distinguishable and congruent with the results of Borchers et al. in [8]. But there is also a big difference in the applied pressure, the professional player uses much less overall force and also the peaks are much lower compared to the amateur players. For the teacher, the data could give a good insight into the applied pressure, which is usually

Lip Pressure: Trumpet Performance Research

LiPr – Aufnahmeprotokoll (Grosshauser / Bertsch)

Player: *Age:*

Instrument: *Years playing:*

Level: [ ] Amateur [ ] Trp-Student [ ] Semin-Pro [ ] Pro

Date

TASKS

(A) Open-Notes {Half-note, Half-note rest} **mezzoforte : c1 – g1 – c2 – e2 – g2**
Metronom : 60 !
Filename:

(B) Open-Notes {Half-note, Half-note rest} **piano (pp) : c1 – g1 – c2 – e2 – g2**
Metronom : 60 !
Filename:

(C) Open-Notes {Half-note, Half-note rest} **forte (ff) : c1 – g1 – c2 – e2 – g2**
Metronom : 60 !
Filename:

(D) Music
Metronom : 60 !
Filename:

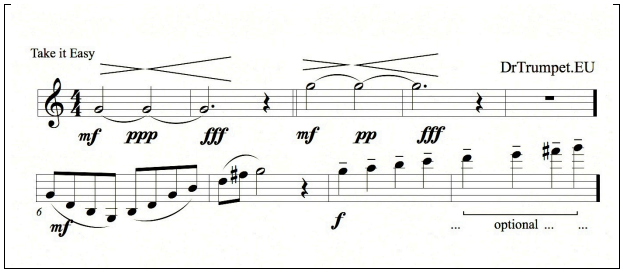


Figure 5. This figure shows the sequence, every subject played during the evaluation. Typical playing scenarios with varying lip pressure are chosen. The last four notes "optional" are hard to play and the highest pressure peaks were reached.

hard to see and even more difficult to quantify, especially during daily teaching routine.

4. PEDAGOGICAL ASPECTS AND APPLICATIONS

First of all, objective measurement and real-time data visualization of lip pressure is a complete new parameter in teaching and practicing. Already visualization itself can increase the awareness of certain problems. Based on this idea, the different live plots in the app were developed (one visualization type see in fig. 1). A second step was additional real-time feedback to inform the musician, if certain thresholds (e.g. maximum pressure in certain playing sequences) were exceeded. Certainly, every musician has her/his own playing forces, but e.g. in case of too high lip pressure the teacher can measure and visualize it or even adjust individual real-time feedback with the described set-up. Furthermore she/he can include the additional information in the daily teaching routine or additionally adjust the threshold individually for automated feedback. With these individual adjustments the student might also be able to use the system while practicing at home.

Comparisons between players (interindividual variability) can reveal the minimal force needed for certain playing techniques, and help students to relate their values to improve economic and ergonomic playing techniques. "No pressure" instructions of teachers have been confusing for a long time (see Wilken in [9]). Less pressure is positive, but there is no standard or correct pressure number to tell, since embouchures are always individual and have to fit physiological parameters (see Bertsch in [10]). Since the engaged amount of force for one player also depends on multiple factors (intraindividual variability) studies during different conditions can be of enormous pedagogic help. The integrated real-time feedback features allow several further applications e.g. to indicate if a playing break is necessary for relaxation, to study the difference between embouchure setups, or the importance and level of embouchure muscles.

Certain qualities in playing can only be achieved by play-

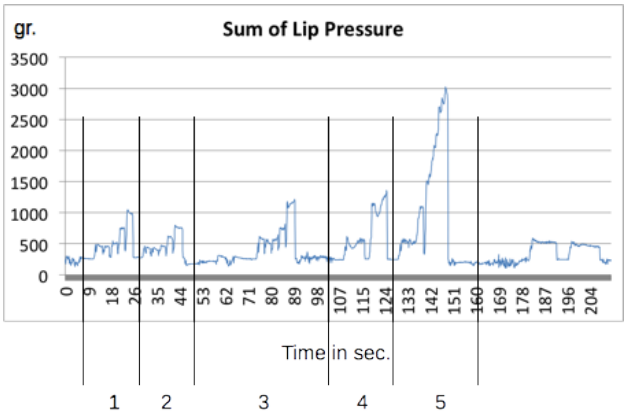


Figure 6. This figure shows the lip pressure curves of the sequence in fig. 5 of a professional trumpet player played with a piston valve trumpet. Part 1 to 3 correspond to (A) to (C) in fig. 5, part 4 is first staff, part 5 is second staff. The last four notes ("optional" end of part 5) are the most difficult one to play and a lot of force is applied.

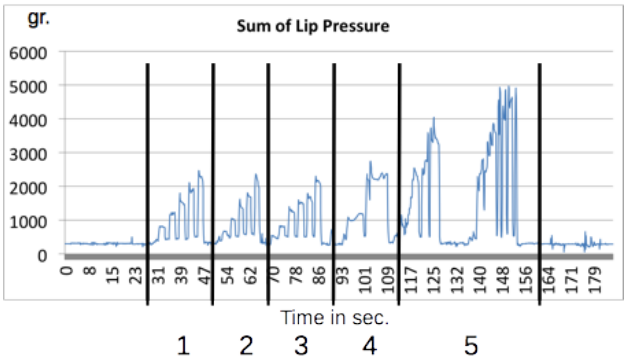


Figure 7. This figure shows the pressure curves of the sequence in fig. 5 of a student, played with a piston valve trumpet. Part 1 to 5 are the same as in fig. 6, part 5 was played two times, but not reaching the highest note. Much higher forces are clearly recognizable, with maximum peaks up to 50 N compared to 30 N of the professional player (fig. 6).

ing "on the air", which usually result in lower lip pressure. The basic principle is simple, too much pressure hinders the lips to vibrate and can end in a failure of the tone production. On the other side, the higher the note, the higher the pressure. This means, between too much and "correct" lip pressure is a fine line, which can be observed and discovered with the lip pressure measurement setup presented in this paper. Since lip pressure also influence the vibration characteristic of the lips and can change the lip opening area, the control of lip forces allows also better control of the instrument sound (see Bromage et al. in [11]).

Finally this might support trumpet/cornet students to find the correct "positive" lip pressure and additionally playing with a sensor might already have a positive effect on focusing on this specific problem. The data recording and visualization further allows teachers and students long term observations and adapting the practicing scheme more individually. But also self-observations are possible, e.g. which pressure is necessary to reach a certain tone quality or simple correlations between visualization and sound might already help to find good or bad influences of certain lip pressure conditions.

5. CONCLUSION

The described setup demonstrates a promising approach, how a practical integration of sensor technologies into daily teaching, practicing and performance science and observation could be realized. The main goal was the plug-and-play idea to use the sensors needed by simply adding them into an existing setup, here the trumpet and cornet. This is a main requirement for daily use and acceptance in the music community.

The main application fields are all kinds of teaching, practicing and learning scenarios but also in music medicine, new and augment musical instruments and performance research in general. But force data furthermore provide an insight of the applied forces while playing, which opens up many possibilities in health related questions like cramping recognition or fatigue detection.

Furthermore, considering augmented musical instruments, this additional parameter may lead to several new expression possibilities e.g. for additional parameter adjustment like sound effects in real-time or manipulation of other control signals like DMX or MIDI.

Acknowledgments

Special thanks to www.bonsai-systems.com for providing us the data logging app and further technical support.

6. REFERENCES

[1] K. Ng, "3d motion data analysis and visualisation for technology-enhanced learning and heritage preservation," in *AIKED'09: Proceedings of the 8th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2009, pp. 384–389.

[2] W. Goebl and C. Palmer, "Tactile feedback and timing accuracy in piano performance," *Experimental Brain Research*, vol. Volume 186, pp. 471–479, 2008.

[3] T. Grosshauser and G. Tröster, "Further finger position and pressure sensing techniques for strings and keyboard instruments," in *New Interfaces for Musical Expression, NIME13*, 2013.

[4] M. Bertsch and A. Mayer, "3d transducer for measuring the trumpet mouthpiece force," in *roceedings of the Forum Acusticum Budapest*, 2005.

[5] A. Mayer and M. Bertsch, "A new 3d transducer for measuring the trumpet mouthpiece force," in *Proceedings of Second Congress of Alps-Adria Acoustics Association and First Congress of Acoustical Society of Croatia*, 2005.

[6] J. F. Petiot, "Measurements of the force applied to the mouthpiece during brass instrument playing," in *Proceedings of the SMAC03 (Stockholm Music Acoustics Conference 2003)*.

[7] T. Grosshauser, B. Hufnagl, G. Tröster, and A. Mornell, "Sensor based hand weight and pressure measurements in trombone playing," in *SMAC Stockholm Music Acoustics Conference*, 2013.

[8] L. Borchers, M. Gebert, and T. Jung, "Measurement of tooth displacements and mouthpiece forces during brass instrument playing," in *Medical Engineering and Physics*, vol. 17, 1995, pp. 567–570.

[9] D. Wilken. [Online]. Available: www.wilktone.com/?p=1936

[10] M.Bertsch, *Collected Papers in Musical Acoustics 1995/2003*. Schriftenreihe des Instituts für Wiener Klangstil - Musikalische Akustik Universität für Musik und darstellende Kunst Wien. Band 6 (2003). [Online]. Available: http://personal.mdw.ac.at/bertsch/MB-PDF/2003e_MB_Collected-Papers-(Habil2).pdf

[11] S. Bromage and M. Campbell, "Open areas of vibrating lips in trombone playing," in *Acta Acoustica*, 2010.

The Virtuoso Composer and the Formidable Machine: A Path to Preserving Human Compositional Expression

Jason Cullimore and David Gerhard
University of Regina
jason@jasoncullimore.com | gerhard@cs.uregina.ca

ABSTRACT

Many contemporary computer music systems can emulate aspects of composers' behaviour, creating and arranging structural elements traditionally manipulated by composers. This raises the question as to how new computer music systems can act as effective tools that enable composers to express their personal musical vision–if a computer is acting as a composer's tool, but is working directly with score structure, how can it preserve the composer's artistic voice? David Wessel and Matthew Wright have argued that, in the case of musical instrument interfaces, a balance should be struck between ease of use and the potential for developing expressivity through virtuosity. In this paper, we adapt these views to the design of compositional interfaces. We introduce the idea of the virtuoso composer, and propose an understanding of computer music systems that may enhance the relationship between composers and their computer software tools, particularly with regard to the composition of tonal music and related traditional forms. We conclude by arguing for a conceptualization of the composer/computer relationship that supports the continued development of human expression and creativity in compositional activities.

1. INTRODUCTION

In this article we introduce the idea of the virtuoso composer. We advocate for a perspective on (composition-oriented) computer music system design that allows composers to express their musical vision effectively and transparently. We suggest that, like a virtuosic instrumentalist, the virtuoso composer works with an instrument upon which they can achieve mastery, *i.e.* a computer software system that may extend his or her compositional capabilities and support the expression of original, personal compositional ideas.

We choose to describe the composer who excels at the use of a computer music environment as a virtuoso because, like a virtuoso instrumentalist, the expression of their ideas can depend on the skill with which they can use their expressive tool, in this case a computer music system. The act of composition, of creating music through interaction with software system, does not necessarily occur in real time (as in the case of a performer playing on, for example, a violin). Nevertheless, like a violinist, a composer who writes music with the aid of a computer-based tool may benefit from experience with that software system, learning to exploit its inherent capabilities and work around its limitations through practice and study. The result is that, like a musical instrument, a system designed to enable a composer to create new musical compositions can result in more individual music (as measured, for example, by its success in projecting the expressive or intellectual aims of the composer) once the composer has achieved mastery with that system. The design of such a system may become a important factor governing that composer's workflow. This is similar to the manner in which a musician who plays the violin may find that they can achieve maximum expression only through use of that instrument; a trumpet, or even a cello, may not yield a similar degree of musical expressivity to the virtuoso violinist, despite the advanced state of his or her musical knowledge and accomplishment.

The question may be asked: how can this conception of the composer and his or her computer tools influence how a developer designs and implements a computer-based composition environment? Furthermore, does the computer, by actively taking over aspects of compositional processes traditionally in the domain of the composer, reduce or magnify the transmission of the composer's ideas? Please note that while we understand the diversity of compositional structures available to the modern composer, this paper deals primarily with tonal and traditional musical forms, an area of composition which we feel is still relevant due to its popularity and its prevalence in interactive media scores.

2. COMPUTERS AS CREATIVE TOOLS

As computers become adept at manipulating elements of score structure traditionally defined by the human composer, these composers may begin to face an existential crisis. There are ample examples of computer systems that surpass human capability: in head-to-head matches IBM's *Deep Blue* defeated the ranking chess grandmaster of its time [1], a system called *Watson* beat two *Jeopardy* champions in a trivia contest [2] and David Cope's *Experiments in Musical Intelligence* software has famously fooled a musically educated audience into believing that its own generated output was actually composed by J. S. Bach [3]. If a computer can successfully emulate aspects of composers'

Copyright: ©2015 Jason Cullimore 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.

activities, how can a composer working with such a system take ownership of the music produced with it? Will the human composer even be necessary or desirable, when such systems reach sufficiently advanced states of development?

2.1 What the Composer Brings to Music

The above questions may lead us to ask what a human composer brings to music. One possible answer, founded on comparisons between music produced with samples versus "authentic" instrumental performance, might relate to the idea of "expression". With regard to music composed with the aid of computer music systems, the composer is arguably best served when something of significance expressed by that composer can be identified within the musical output of the overall system. The degree to which they are served is a reflection of the degree to which their expressive choices, as evidence of their artistic voice, are preserved in the system's musical output. Under this definition, the most transparently expressive compositional technology might be the pencil, for it arguably adds nothing of substance to the composer's creation, serving totally as an extension of the composer's own creative processes.

However, composing with a pencil is extremely challenging, making significant demands upon a composer's memory, sight-reading skill, and ability to sustain musical imagery. The simple pencil is thus an extraordinarily demanding compositional tool. Alternatively, the use of a computer as a compositional aid can alleviate some of these issues. For example, a computer can act as an externalized memory, since it can play back a score, providing a aural representation of a musical work at any point in a composition, diminishing the composer's need to memorize the position and role of each written note. A simple pencil cannot do this.

A computer is often viewed as a source of human empowerment. Early in the development of computers, the devices were seen as a means to improve life and extend human ability, such as Vannevar Bush's theoretical *Memex*, an information-storage device which would, had it been achievable at the time, have aided in the preservation and dissemination of knowledge [4]. Other advancements such as Engelbart's computer-centric innovations (such as the mouse, window and word processor) were designed to help humanity deal with increasingly complex problems by "augmenting human intellect" [5]. This view of technology as enriching human experience seems to continue to resonate today as consumers pursue new technologies (*e.g.* tablet computers, smartwatches and social media) and improvements in existing technologies (such as video game consoles with faster processors and more storage space). It can be difficult to see why traditional technology like a pencil could be at all superior to a specially-designed computer program that allows a composer to write more rapidly, and with subjectively more impressive results, that that composer would have achieved had they been forced to wield a more humble tool. Does the incorporation of computer technology and agency into human musical activities have any potentially negative impacts upon creativity?

2.2 What the Computer can Take from the Composer

One line of enquiry relevant to this question involves an examination of how compositional skill may be replaced by computer technology. Evidence for the possibility of a computer-based compositional system to supplant compositional skill may be found in the case of Rachael Y., a composer with an acquired brain injury that caused an amusia (specifically a loss of the ability to sustain musical imagery) [6]. Prior to her accident Rachael had composed in her head, working out score structures in her imagination before committing them to paper. After her accident, she could no longer sustain the memory of her musical ideas, and composing using internal musical imagery became impossible for her. Yet she was able to return to composing despite lacking the ability to sustain the memory of her music: on the advice of a collaborator, she learned to use a computer to record and play back her musical ideas, reducing the load on her own memory. Thus, by adopting a computer system as a substitute for her own damaged cognitive systems (and with the aid of her collaborator), she once again became an active composer. In one post-accident instance, she created a partially autobiographic musical work that reflected her "rediscovery of identity" [6]. The work is deeply personal, but it must be acknowledged that the composition could only exist because of the aid of the composition software she used.

By extension, those composers who take advantage of a computer-based composition environment by using it as an external aid to memory (even if just to review the final mix of a composition to check for pitch errors) may be said to be benefitting from an externalized substitute for processes that composers who lack such technology would normally have to undertake themselves. And if such computer-using composers are unable or unwilling to learn how to enact these creative processes themselves, they risk becoming dependent on their computer-based tool, in a way that they could not were they only using a simple pencil. While pencil-based composers are comparatively limited in the sense that they lack a tool that allows them to listen to their scores before handing them to any musicians, they are also enhanced by the experience of learning to use the pencil: they develop a facility with their memory and mental imagery that may exceed those they would have possessed had they began their compositional development with the reliance on software systems that obviated the need for such extremely developed mental abilities.

3. THE VIRTUOSO INSTRUMENTALIST

Virtuosos may be described as "human beings that excel in their practice to the point of exhibiting exceptional performance." [7] It is common to apply this label to instrumentalists who excel in performance upon an instrument, and history makes mention of many virtuoso musicians from Paganini to Hendrix. Yet historically, the virtuoso does not merely perform with facility; he or she must, through performance, *express* his or her own interpretation of the musical work being performed. The success of that interpretation can be a criterion for the judgement of the virtuosity

of their performance [8].

Another significant aspect of instrumental virtuosity is that it represents action at "the limit of human capacities" [7] Virtuosity is no commonplace human attribute; it is a manifestation of rare skill or talent, and is not immediately available to the typical performer. Nevertheless, attempts to model musical virtuosity in automated improvising systems, such as Pachet's bebop phrase generator [7], have demonstrated that it is possible to understand or codify some of the processes that go into a virtuosic performance. Such an understanding is exhibited by some virtuoso musicians themselves–for example Mark Levine, an accomplished jazz pianist, has said that most of what goes into the production of a great jazz solo is both explainable and teachable [9]. Yet Levine also asserts that a great deal of thought and practise will still be required of prospective jazz virtuosos, in the same way that Pachet's bebop phrase generator required thorough analysis of the structure of existing jazz improvisations.

In terms of designing computer-based musical instruments that may support virtuosic performance, David Wessel and Matthew Wright present a perspective that proves helpful. They first argue [10] that a properly-programmed computer and interface may constitute a musical instrument. Performing music on a computer through use of a gestural interface is akin to performing on a traditional instrument such as a violin or piano. With traditional instruments, the musician is in direct control of the means of sound production. In the case of a computer system, the user controls the event-level progression of musical ideas, which Wessel and Wright argue may be most effective under a "one-gesture-to-one-acoustic-event" paradigm.

Wessel and Wright go on to examine the distinctions between instrument forms. One of the more effective ways to compare different musical instruments is based on ease-of-use. Wessel and Wright suggest that not all instruments are equally easy to perform upon, and they introduce two related factors: the ease with which an instrument can be used by an individual unfamiliar with it, and the degree to which continued practice upon the instrument promotes the development of an expressive virtuosity. In designing a musical instrument, whether computer-based or not, one is creating a system which will challenge its performer to some degree, and yield an output of particular expressive qualities. These two properties can be related, in that more difficult musical instruments often require more subtle or precise control from their performers, and that this control complexity can be associated with a greater number of dimensions of modulation. For example, the violin, often considered an extremely challenging instrument, has many control dimensions (including bow pressure and speed, rate of vibrato and fingering position), each of which can contribute to expressive variation in the violinist's performance. The instrument rewards those who commit the (often considerable) time needed to achieve mastery with the ability to use the instrument with greater expressive freedom.

4. THE VIRTUOSO COMPOSER

In traditional Western tonal music the composer works in the domain of the structural elements that are found in a musical score: pitch, timing, dynamics and timbral choices that, customarily, are not to be altered significantly by a performer. Expressive choices are available to the performer, but they should always be consistent with the musical structure represented in the score. Each note in a score is part of a statement of the composer's creative imagination.

Yet computer software systems are increasingly used by composers as tools to realize their compositions. Furthermore, computer systems themselves can take autonomous roles in the composition process, manipulating music structure independently of the composer. Some well-known systems allow for minimal expressive interaction, such as David Cope's *Experiments in Musical Intelligence*. [11] Others are built around the idea of promoting interaction, such as David Rokeby's *Very Nervous System* [12]. An illustrative example of a system that combines expression with compositional elements may be found in *Bloom*, a popular iOS app by Brian Eno and Peter Chilvers [13].

4.1 Creativity and *Bloom*

Bloom is a generative music system that functions in a semi-autonomous manner [14]. It blurs the line between composition and performance, but we would argue that it can be considered a composing tool since it allows users access to musical score structures (*e.g.* pitch and timing) normally manipulated by composers. The generated musical environment consists of two layers. One is a background "pad" sound defining a tonal centre and overall mood. The user of *Bloom* can add another layer to this musical backdrop by tapping on the screen. For each tap, the app responds by playing a note that conforms to the overall tonal nature of the music at that moment in time, and the pitch and rhythmic placement of the note is partly governed by where and when the user taps.

It may be impossible for any user, even a musically-untrained one, to play an "incorrect"-sounding note when using *Bloom*, due to the manner in which it translates user taps into musical notes. The user's performance always results in notes that enhance the calming mood of the underlying pad. Yet the subjective experience of the user is that they are in control of the music, since their taps seem to map directly to the timing and pitch of the generated notes.

Does *Bloom* make any user into an instant composer? Certainly *Bloom* does allow the user access to musical score structure, which is within the domain of the composer. However, using the app does not grant much expressive freedom–the result is always going to be similar, regardless of who is performing, because the app takes over many of the creative decisions that are normally afforded to the composer (*e.g.* the precise definition of note pitches, the exact timing of musical notes, the key of the piece, and so forth).

We have argued that expressivity is necessary for virtuosity, and *Bloom* does not grant this aspect of control to

any great extent. Under Wessel and Wright's view, *Bloom* may be the compositional equivalent to a kazoo, with an easy entry fee for a would-be composer, but a lack of the expressive performance capabilities necessary to promote virtuosity with the app. *Bloom* may allow its users access to some basic choices that composers make, but as a creative composing tool, it is fairly limited. The other aspect of virtuosity we describe, its tendency to require sustained effort to achieve, is also not in evidence in the use of *Bloom*: if all compositions made using the app are similar in structure or form, how can a user strive to achieve a transcendent level of expressivity with the software? If the practised user cannot exceed the effect of a novice, then virtuosity is impossible.

4.2 Computer Music Composition Systems As Virtuosic Tools

A computer-based composition environment that supports the development of virtuosity must do what *Bloom* cannot: offer its users the ability to express their individual ideas, and reward them with heightened expressive power through engaging with the system. What properties would such a system, if successfully implemented, possess?

If an ultimate tool enabling virtuosic self-expression for composers is the pencil, that may serve as an initial model. There are already numerous composing environments available to composers to record their ideas (*e.g.* notation software such as *Sibelius* and *Finale*, the sequencers *Cubase* and *Logic*), but computers can act as more than mere recording devices. A defining component of computers is their processor, which enables independent manipulation of musical structure. The computer processor is the computer music system's key advantage over a pencil, in part because it allows the computer to perform a work differently each time it is initiated.

In his influential essay "What a Musical Work is", Jerrold Levinson argues that "a musical work must be capable of being created, must be individuated by context of composition, and must be inclusive of means of performance" [15]. A typical musical work of the Western tonal idiom (which was the domain of Levinson's argument) has a specific composer who wrote it at a specific time and has defined its instrumentation. A computer music artwork need not conform to this definition, since the computer can act as an agent that modifies musical structure in response to its listeners, the ambient environment, and the will of its composer (whether or not the composer is actually present at the time and engaged with the performance). This is arguably the greatest contribution of computers to music composition: a musical work is no longer a single score structure composed by an individual, but rather a superset from which many musical structures may be derived. Each performance of *Bloom* generates a specific musical score, but *Bloom* itself is much more than a score, as it comprises all possible score structures that may be derived from it through interaction. The computer's agency elevates the computer music work above a mere score, adding a dimension to the archetypal musical work that was impossible before computer processing of musical scores was developed.

As van Geelen argues, while "interactive audio has its roots in linear audio," it need not be confined to the form [16]. Computational agency, when incorporated into a composer's workflow, adds a dimension to a musical work that has yet to be fully explored or understood by contemporary artist/composers. Computer music systems that enable nonlinear music represent an immense artistic opportunity, a new frontier for creativity. Yet some systems, such as *Bloom* or David Cope's *Experiments in Musical Intelligence*, assume so much responsibility for defining the event-to-event structure of their musical output that they deny the composer the freedom to explore the creative possibilities of this new artistic frontier. In other words, they do not allow composers to achieve virtuosity with their systems, robbing them of the opportunity to skillfully express their individual musical ideas. It would be tragic if the creative opportunities afforded by integrating computational agency into the creation of dynamic music were to be negated by system designers who fail to give the composer a chance to engage in personal self-expression.

We thus encounter a conundrum. In order to progress into an era of interactive music, composers will need access to computer systems that can transform musical ideas during a performance (and without the real-time intervention of the composer) in response to their environment. Yet to transfer the power to restructure musical ideas to computer software negates some of the expressive contributions of the composer. How can a computer scientist design a computer-based system for composers that allows them creative and expressive freedom, while still maintaining the ability to produce music that adapts to its ambient environment?

Currently we can provide no definitive answer to this question; however, we can provide a perspective that may support the eventual realization of such a system: when designing computer music systems to support *human* compositional creativity, begin with an understanding of the pencil. By requiring more of its user than he or she is initially comfortable with, by subjecting him or her to a period of rigorous study and sometimes uncomfortable struggle, the pencil enforces an elevation of skill and mental faculties even as it transmits an individual's creative expression in unaltered form. While computers have provided humanity with quick solutions to difficult problems, the designers of new systems for creating music should remember that they are involved in processes relating to the creation of musical art, and that art has been a deeply human expression. Such researchers are actually building upon centuries of musical development through which great composers have provided moments of transcendent emotional experience. If their system designs are to tap into this rich vein of creative activity, they must not push aside the composer's intentions by replacing them with a powerful but self-contained computational agency. Truly dynamic music that responds in real time to its listeners and may ultimately possess an awareness of its environment is becoming a reality, an opportunity for composers to redefine and reinvigorate the musical arts (which, it must be said, have

in recent times been accused of failing to find a balance between innovation and popular appeal [17].) To lose this opportunity because the tools necessary to engage directly and meaningfully with it were never developed would diminish music and reveal a particular lack of foresight on the part of computer music system designers.

4.3 A Way Forward

The composer who excels in expressing their original ideas through a computer music system can be as much a virtuoso as a great violinist. The fact that their score is not output in real time does not diminish the value of the composer's struggles with their instrument, learning to exploit its limitations, and ultimately becoming skilled enough to draw from it music that expresses profound emotion in an innovative manner. When these ideas can be expressed dynamically, with the computer responding to the performance environment while still transmitting the inspirations of the composer, a new musical art form may evolve.

Yet without the tools to realize this revolutionary new musical development, the future for music may be bleaker, with the computer becoming the source of musical invention and the human musical spirit weakening.

There is a movement gaining strength today known as "Ubiquitous Music" or UbiMus, which in part seeks to increase the accessibility and spread of creative and artistic tools, including outside of traditional studio settings [18]. Proponents of the movement have argued that "all humans are creative" and as a result will benefit from opportunities to engage in self-expression [19]. We agree that this perspective is valid and that to support the design of creative tools for all people (whether or not they have training in an art form) is one of the most exciting opportunities afforded by ubiquitous computer technology today. Yet there may be a temptation to make the "entry fee" for using a new technology lower, to make it accessible to more people. We would suggest that doing so may lead to systems that fail to support compositional virtuosity, in turn diminishing their creative and expressive potential.

5. CONCLUSION

We urge the designers of computer music systems to remember to consider the degree to which their systems promote compositional virtuosity and the creative power of their users. If the computer takes on most or all of the creative work, that is a valid innovation, but it does not lead human creativity in a particularly meaningful direction. If, however, we consider the role of the human in our creative systems, designing them as formidable machines that resist a composer's initial explorations, but ultimately reward him or her with invigorating and compelling self-expression when they overcome this resistance, we may begin to develop systems that will sustain a remarkable new era of interactive and adaptive musical innovation. Here the thoughtful and imaginative software developer meets the virtuoso composer on common ground, to the benefit and enrichment of music.

6. REFERENCES

[1] M. Campbell, A. J. Hoane, and F.-h. Hsu, "Deep blue," *Artificial intelligence*, vol. 134, no. 1, pp. 57–83, 2002.

[2] S. Baker, *Final Jeopardy: man vs. machine and the quest to know everything*. Houghton Mifflin Harcourt New York, NY, 2011.

[3] G. Johnson, "Undiscovered Bach? No, a computer wrote it," *New York Times*, November 11 1997.

[4] V. Bush, "As we may think," *The atlantic monthly*, vol. 176, no. 1, pp. 101–108, 1945.

[5] D. Engelbart, "Augmenting human intellect: a conceptual framework," in *The new media reader*, N. Wardrip-Fruin and N. Montfort, Eds. The MIT Press, 2003, pp. 95–108.

[6] O. Sacks, *Musicophilia: Tales of music and the brain*. Vintage Canada, 2010.

[7] F. Pachet, "Musical virtuosity and creativity," in *Computers and Creativity*. Springer, 2012, pp. 115–146.

[8] M. Pincherle and W. Wager, "Virtuosity," *Musical Quarterly*, pp. 226–243, 1949.

[9] M. Levine, *The jazz theory book*. O'Reilly Media, Inc., 2011.

[10] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," *Computer Music Journal*, vol. 26, no. 3, pp. 11–22, 2002.

[11] D. Cope, *Experiments in musical intelligence*. AR editions Madison, WI, 1996, vol. 12.

[12] D. Rokeby, "Transforming mirrors," *Leonardo Electronic Almanac*, vol. 3, no. 4, p. 12, 1995.

[13] B. Eno and P. Chilvers, "Bloom," 2008.

[14] D. Jones, A. R. Brown, and M. dInverno, "The extended composer," in *Computers and Creativity*. Springer, 2012, pp. 175–203.

[15] J. Levinson, "What a musical work is," *The Journal of Philosophy*, pp. 5–28, 1980.

[16] T. van Geelen, "Our interactive audio future," in *The Oxford Handbook of Interactive Audio*, K. Collins, B. Kapralos, and H. Tessler, Eds. Oxford University Press, 2014, pp. 557–569.

[17] G. E. Garnett, "The aesthetics of interactive computer music," *Computer Music Journal*, vol. 25, no. 1, pp. 21–33, 2001.

[18] M. S. Pimenta, D. Keller, and V. Lazzarini, "Ubiquitous music: a manifesto," in *Ubiquitous Music*, D. Keller, V. Lazzarini, and M. S. Pimenta, Eds. Springer, 2014, pp. xi–xxiii.

[19] N. Zagalo and P. Branco, "The Creative Revolution That Is Changing the World," in *Creativity in the Digital Age*. Springer, 2015, pp. 3–15.

Acoustically Guided Redirected Walking in a WFS System: Design of an Experiment to Identify Detection Thresholds

Malte Nogalski
University of Applied Sciences Hamburg
malte.nogalski@haw-hamburg.de

Wolfgang Fohl
University of Applied Sciences Hamburg
wolfgang.fohl@haw-hamburg.de

ABSTRACT

Redirected Walking (RDW) received increasing attention during the last decade. While exploring large-scale virtual environments (VEs) by means of real walking, RDW techniques allow to explore VEs, that are significantly larger than the required physical space. This is accomplished by applying discrepancies between the real and the physical movements. This paper focuses on the development of an experiment to identify detection thresholds for an acoustic RDW system by means of a wave field synthesis (WFS) system. The implementation of an automated test procedure is described.

1. INTRODUCTION

The basis for this paper is the development of an experiment to identify detection thresholds for an auditory application by means of wave field synthesis (WFS) using redirected walking (RDW) techniques. The results of this experiment shall be used to develop and configure an acoustically guided RDW application.

In immersive virtual environments (IVEs), in which users navigate by physically walking through the physical space (or tracking area), RDW describes a technique, which applies a controlled discrepancy between the physical movements and the virtual effect, to allow the user to explore a virtual environment (VE), which is larger than the tracking area, in which she physically navigates.

The results of the experiments described in this paper shall give further insight into the applicability of RDW techniques in acoustic environments presented by a highly sophisticated audio system. They shall be compared with previous visual and acoustic experiments, to contribute to the identification of differences between the acceptance of RDW manipulations in visual and acoustic environments and start a comparison between different audio playback systems.

This report will start with a section on related work, giving insight into general RDW methods, experiments, detection thresholds and the role of the auditory aspect so far. Following that, it will explain, which of the previously described RDW methods can be utilized in an acoustically guided experiment, and a brief overview of the laboratory will be given.

The main part will lead through the requirements, the process and the test groups of the experiment, followed by a brief insight into the the implementation of the test sequence.

Finally a summary and a glimpse into future work will be given.

Copyright: ©2015 Malte Nogalski 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.

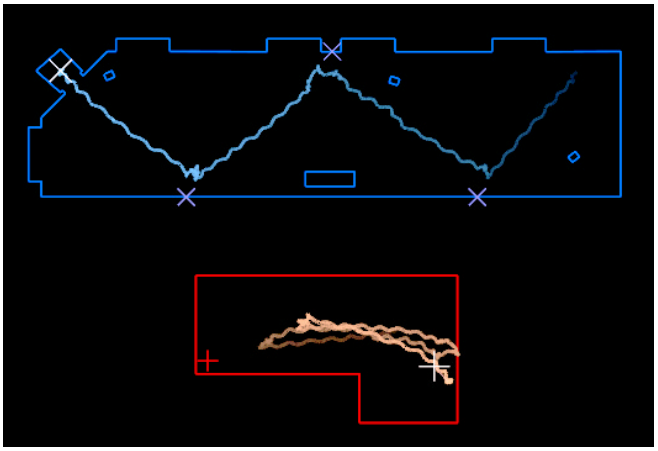


Figure 1. Overhead views of the path taken by the user in the virtual environment (above in blue) and the laboratory (below in red). Figure taken from [1].

2. RELATED WORK

This section will give an introduction to the basic concepts and algorithms of RDW. Various approaches to apply gains to manipulate users' movements are reviewed, and the reported thresholds for the identification of these manipulations are summarized for both visually and non-visually guided RDW.

2.1 General redirected walking (RDW)

According to Razzaque et al., Michael Moshell and Dan Mapes made first attempts at visual RDW in 1994 [1], but could not elude the problems of what they identified as simulator sickness and the limitations of virtual environment systems, in particular the tracking systems.

In 2001 Razzaque et al. assume, that RDW might now be possible with the recent development in VEs as well as accurate, low latency, wide-area tracking systems and therefore address the problem once again [1].

Razzaque et al. state in their theory, that humans rely primarily on vestibular, visual and auditory cues for balance and orientation, citing [2]. Furthermore they state, that these cues are used to distinguish between self-motion (the user moves) and external-motion (the objects around the user move). In respect to previous research they also state, that a consistency

of multiple of those cues may increase the chance, that external-motion may be perceived as self-motion [3].

Their technique rotates the virtual environment around the user in such a way, that the user is made to always walk towards the farthest wall of the tracking area. In theory and with a tracking area large enough, according to Razzaque et al., it should be possible to present a virtual environment of infinite extent. However, with a decrease of tracking area, eventually more rotation has to be applied to the virtual environment, to keep the user within the physical perimeter of the tracking area and each increase of applied rotation also increases the chance of detecting the manipulation. The thresholds for applied rotational distortion is therefore an tradeoff between a lower detection probability and less physical space requirement.

Razzaque et al. even claim, that "Even while standing still, the user unknowingly rotates her head and torso with the virtual scene" and assume, that the explanation lies within the user's own balance system in regard to [2].

Within their experiment, which is often referred to as the first working case of RDW [4], Razzaque et al. used a head mounted display with stereo headphones to present the visuals and spatialized audio. Users had to reach a way point, then first turn towards the next way point, before moving straight towards that new way point without wandering around. During these turns towards the next way point, a rotational scaling was applied to the representation of the virtual environment to point the user towards the direction she came from, while making her see, hear and believe, that she turned towards a point further down the hallway. The blue box in figure 1 shows the path, the user has taken within the virtual environment. The red box below shows the path she took within the tracking area at the same time. Small misalignments after the turns were corrected by further applying small rotational distortions while the user was walking towards the next target. This explains the arcs in figure 1. So while actually walking back and forth in a rather small physical room, the user had the impression of having passed through a significantly lager area. This experiment should also work for a virtual hallway of infinite length.

2.2 An algorithm to dynamically apply gains

The algorithm used in [1] is shown in figure 2. As mentioned before, in [1] users even compensated for small amounts of rotational distortion to the virtual environment while standing still. This is the *baseline constant rotation* and the first of three basic factors to the resulting rotational distortion. The other two are a scaling to the users real rotation and a rotation proportional to the users linear velocity (i. e. walking speed), but only the maximum of those three would find consideration and be scaled by the sine of the angle between the next virtual target and the next real target. Finally the resulting distortion rate was limited by a fixed threshold, which was determined as being the threshold for imperceptible rotational distortion by previous tests.

2.3 The human locomotion triple

In [5] Steinicke et al. introduce the user's locomotion triple (in [6] then named human locomotion triple (HLT)). The HLT consists of three normalized vectors: (s,u,w). The strafe vector s is orthogonal to the walking direction and parallel to the

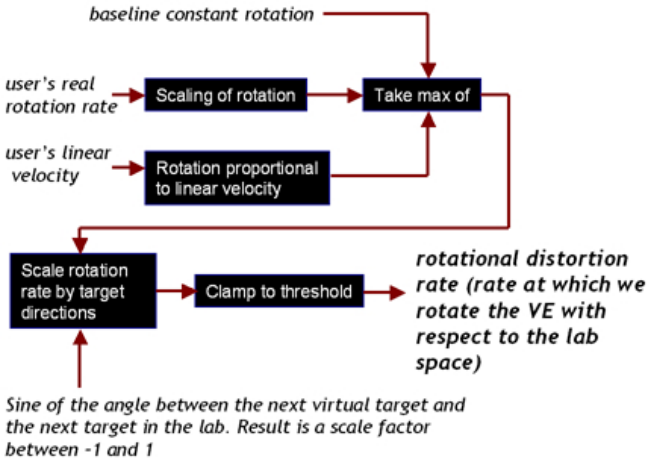
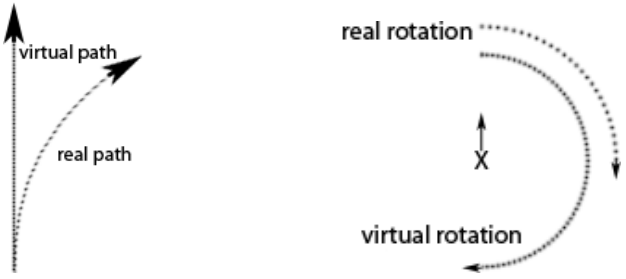


Figure 2. The algorithm for computing the rotational distortion rate. Figure taken from [1].



(a) The curvature gain bends a real path into a distorted virtual path. The user unknowingly walks on a curved path.

(b) The rotation gain scales a rotation with the effect, that the virtual rotation is greater or smaller, than the real rotation.

Figure 3. The curvature gain bends a path and the rotation gain scales a rotation.

walking plane, the up vector u represents the tracked head orientation and the walk-direction vector w represents the tracked direction of walk. Through the HLT, manipulations can be applied to users' paths by various gains as described during the next sections.

2.4 Gains to manipulate the users' movements

While the tracking system constantly provides up-to-date data for the users real world position and orientation defined as P_{real} and R_{real}, the translation is defined by

$$T_{real} = P_{cur} - P_{pre}$$

where P_{cur} is the current real position and P_{pre} the previous/last considered real position. A translation gain $g_T \in \mathbb{R}^3$ is defined for each component of the HLT: $g_T[s]$,$g_T[u]$,$g_T[w]$ by

$$g_T := \frac{T_{virtual}}{T_{real}}$$

.

By such gains the mapping of real world movements (in this last case translations) can be scaled up or down, depending on the values of the gain. A $g_T < 1$ would result in a smaller translation within the virtual world ($T_{virtual}$) in respect to the

tracked translation in the real world (T_{real}), while a $g_T > 1$ would result in a larger translation in the virtual world and such enabling the users to cover a larger virtual distance. A $g_T = 1$ would draw the real world and the virtual world to scale as if no gain was applied at all.

In the same manner, Steinicke et al. introduce gains for rotation, curvature and displacement as well as time-dependent gains. Rotation gains are applied to rotations of the head and result in a greater or smaller rotation of the virtual world, while a rotation of the head is tracked, as illustrated in figure 3b. Rotation gains are defined by the quotient of the considered component of a virtual world rotation $R_{virtual}$ and the real world rotation R_{real}:

$$g_R := \frac{R_{virtual}}{R_{real}}$$

The curvature gain stimulates users to unknowingly walk an arc in the tracking area while walking on a straight line in the virtual environment even when she does not willingly rotate, as illustrated in figure 3a. curvature gains are defined by a segment of a circle with the radius r:

$$g_C := \frac{1}{r}$$

Displacement gains map real world rotations into virtual world translations ($R_{real} => T_{vritual}$). Time-dependent gains can be defined like all the other gains, though they are not triggered by real world movements, but by time elapsed. The virtual environment is manipulated with no regard to the users movements and such even when she is not moving at all.

2.5 Experiments for detecting thresholds

In March 2008 Steinicke et al. published results of a pilot study [7] within a tracking area of 10m x 7m x 2.5m, in which they identified the following thresholds for RDW without letting the users notice the manipulation:

- Rotations can be compressed or gained up to 30%,
- distances can be downscaled to 15% and upscaled to 45%,
- users can be redirected to unknowingly walk on a circle with a radius as small as 3.3m,
- objects and the virtual environment can be downscaled to 38% and upscaled to 45%.

2.6 Non-visual redirected walking by acoustic stimuli

While a lot of research has been committed to RDW during the last couple of years, almost all contributions are based upon the visualization of the virtual environment as primary stimuli. Some authors state, that the acoustic factor helps users to adjust to the virtual world and that RDW works best, when multiple cues, such as vestibular, visual and auditory, are consistent with each other, as this helps the user to perceive external-motion as self-motion [3, 1]. Even though, the auditory aspect had been paid little attention so far [8].

Razzaque et al. used circumaural stereo headphones to deliver spatialized environmental sounds and prerecorded instructions to the user, where the sounds were intended to be "plausible in both content and source location". The analysis of the effect of the sound was limited to the lack of negative comments by the users [1].

Steinicke et al. used ambient city noise to mask real sounds, but auditory cues were not used to directly aid the RDW technique [9, 6, 10]. A lot of contributions do not mention auditory components at all.

To our knowledge, currently Serafin et al. are the only ones, who really concentrated on the auditory component of RDW techniques. They conducted two different experiments to determine thresholds for acoustic based RDW techniques [8]. To that goal, they adapted two of the experiments conducted in [9, 10], to be used exclusively with auditory feedback. Their experimental setup consisted of a surround system with 16 MB5A Dynaudio speakers in a circular array with a diameter of 1.7 meters, and subjects wore an deactivated head mounted display (HMD) to block out their vision. The only audible feedback in both experiments was the sound of an alarm clock. This choice was meant to be especially fitting, since the sounds of alarm clocks are usually associated with stationary objects and often occur in darkened otherwise quiet places. The sound was delivered through the speaker array by the technique of vector base amplitude panning (VBAP), which, in such a setup, allows the placement of sounds within the circular array of speakers on a plane parallel to the ground level

The first experiment tested the ability to detect rotation gains during self-motion rotations on the spot. The second experiment tested the detection of curvature gains while walking on a virtually straight line while walking from one point on the perimeter of the circular speaker array to a point roughly on the opposite side. Due to the limited space, only short distanced could be covered during each test.

During the first experiment the subjects were asked to turn on the spot towards the sound of the alarm clock. While they were turning, a rotation gain would rotate the alarm clock around the subjects. When they perceived the sound as in front of them, they were asked whether they perceived the virtual rotation as larger (rotation gain < 0) or smaller (rotation gain > 0), than the real world rotation. The virtual rotation is perceived through auditory cues by locating the position of the sound source, while the real rotation mainly by the vestibular system. During the 22 subsequent trials per test subject, 11 different rotation gains were applied. Each gain was applied twice during the course of an experiment. For the evaluation Serafin et al. followed Steinicke et al. [10] and used a psychometric function to determine a bias for the point of subjective equality (PSE). The PSE, where subjects perceived the real and virtual rotation as equal, was determined at 1. Serafin et al. also chose an outbalance of 75% to 25% of the given answers as the detection threshold and these thresholds were reached at gains of 0.82 for greater and 1.2 for smaller responses. This led them to the conclusion, that users can not reliably distinguish between a $90°$ real rotation and a virtual rotation between $75°$ and $109°$. So users can be turned 20% more or 18% less than the perceived virtual rotation. This range is smaller than the one determined in [10], which can be attributed to the fact, that "[...] vision generally is considered superior to audition when it comes to the estimation of spatial location of objects." Goldstein [11] cited by Serafin [8]. In other words, visual cues dominate vestibular, proprioceptive, etc. cues by more,

than auditory cues would and therefore discrepancies between visual and other cues would be accepted to a higher degree.

During the second experiment users were asked to walk on a straight line towards the alarm clock. During their movement 10 different curvature gains were applied (each one twice), which led them on an arced real path and users were asked whether and at which threshold they notice the direction of the bended path reliably. During this experiment the PSE was determined at a curvature gain of -5. The detection thresholds of 75% were reached at gains of -25 and 30 [8].

3. CHOICE OF GAINS TO BE TESTED

This section explains, how the gains for the experiment at hand were selected.

In [6] Steinicke et al. explain five different types of gains to manipulate users' movements within a redirected walking application: the translation gain, rotation gain, curvature gain, displacement gain and time-dependent gain. Time-dependent gains however have to be subdivided into at least time-dependent rotation gains and time-dependent translation gains.

Translation gains, as a means to scale translations and described in [6] were not considered for these experiments for long, since the acoustic perception of distance of humans is much worse then the visual perception. It is presumed, that vast manipulations can be made with translation gains but that these will most likely not be perceived as a different self-motion. Much rather, the initial distance to the object might be perceived differently and such give the impression of false dimensions of the virtual environment. This might be interesting for acoustic RDW applications in general but is not considered as one of the main methods. More interesting might be a version of the translation gain, which translates i. e. physical forward movements into virtual displacements orthogonal to the walking direction, which would have an effect close to the the one of the curvature gain. To keep the amount of tests for each test person in check and because of the close relatedness to the curvature gain it would not become part of this study either.

The displacement gain specifies physical rotations to virtual translations, which might be useful in some situations but will most likely not become a significant part in most redirected walking application, due to high detection of the displacement or in conclusion small manipulation potential.

Time-dependent translation gains have been neglected due to the the reasons stated above. Time-dependent rotation gains have been considered and were part of a pilot study, but were discarded, because no test subjects compensated for the manipulations.

Curvature gains and even more so rotation gains seem to be the most promising and raised the most attention so far, next to translation gains [5, 7, 12, 6, 8]. Especially Serafin et al. also concentrated on these two types of gains during their work about acoustic RDW [8]. The self conducted pilot studies also showed great potential for acoustic tests with the wave field synthesis system (WFS system) and therefore these two types became the center of this research.

Figure 3a illustrates an example of a bended path by the appliance of an curvature gain. The user perceives the virtual path as straight, while she really walks on a curved path. Figure 3b illustrates an example of a scaled rotation by the appliance of an rotation gain. The user perceives the rotation as 180°, but really only rotates by 90°.

4. THE LABORATORY

This section will give a brief overview of the laboratory, in which the experiment will be conducted. An illustration of the laboratory can be found in figure 4 and a more detailed description in [13].

The area of the WFS system is defined by the speaker arrays and covers roughly 5x6 meters. The height of the lower edges of the speakers is just over 2 meters. The reproduction component of the system consists of 26 speaker modules which contain 676 single speakers equally divided amongst 208 channels. The distance between the centers of the channels within each speaker array is 10 cm. All modules are slightly tilted downwards and both of the 6 meter arrays and one of the 5 meter arrays is backed by a wall of the room.

The tracking area is defined by six infrared cameras which are arranged in a square formation of 4x4 meters parallel to the ground in the height of about 2.5 meters with the two extra cameras mounted below two adjoining corners at about 1.5 meters height. All cameras are roughly aligned towards the middle of the tracking area. Due to the range of the cameras, the tracking area is slightly larger than the 4x4 square but the tracking is most robust within these boundaries.

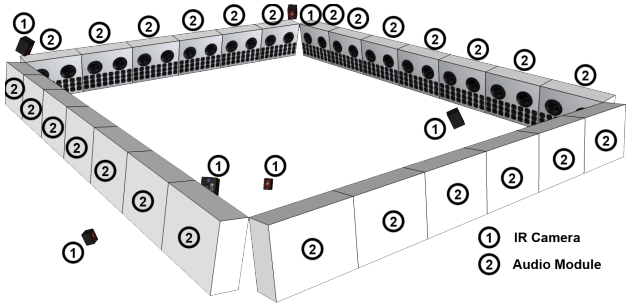


Figure 4. Layout of the laboratory with 26 audio modules and 6 infrared cameras.

The WFS terminal is located in one of the corners of the room. The terminal serves as the primary user interface for the WFS system. Within are also all instances and programs for playing back sounds, propagating commands and rendering for the speaker modules. The main digital audio workstations (DAWs) in use are Ardour [1] and Cubase [2].

5. THE EXPERIMENT DESIGN

This section will explain the prepared experiment, starting with the requirements followed by a description of the test procedure, and closing with the choice of gain values and different test groups.

5.1 Requirements

Some basic requirements had to be met during development of the experiment, to enable success.

[1] https://ardour.org/

[2] www.steinberg.net/de/products/cubase/

- Virtual ambient noise
  - To mask laboratory noises...
  - and to help perceive the external-motion as self-motion.
- Sounds, that stimulate the test persons to turn on the spot.
- Sounds, that stimulate the test persons to walk.

5.1.1 Ambient noise

The ambient noise has two main purposes. The first purpose is to mask laboratory noises, and the second purpose is to give more acoustic cues for orientation and essentially to make the RDW manipulations more plausible. The majority of experiments referred to in section 2 (Related Work) used headphones to play back the acoustic aspects of the immersive virtual environment. In [9] for example headphones were used to play back some kind of ambient city noise "such that an orientation by means of auditory feedback in the real world was not possible" and in [4] noise-canceling headphones playing white noise were used to mask real ambient noise. For RDW by WFS noise-canceling headphones are out of the question, since they would, of course, cancel out the acoustic cues for the techniques as well. Actually, any kind of headphones were considered disturbing and unnecessary. Instead, four virtual sound sources were used to play back the ambient sounds through the same medium as the sounds to direct the test subjects. Through the WFS system. For the ambient sounds, linear sound sources were chosen. Through the randomized sequence of the successive tests, that will be performed and explained later in this section, the path of the test subjects within the consistent virtual environment can not be foreseen. Test subjects could therefore close in on these virtual sound source during their tasks. If they were point sound sources, the test subjects would notice that and if they got too close, they could disturb the experiment, as they are not meant to be located within the focus of the test subjects. Linear sound sources however merely define a direction for the sound, which does not change when test subjects get close to them.

The four linear sound sources were oriented towards the four cardinal directions. Two opposite sound sources were mapped to the same but phase shifted monotonous city noise with some cars driving by but little distinctive aspects [3] . Their main function is, to mask the laboratory noises. One of the other sources got the sound of a playground [4] and the opposite one of a small pedestrian zone [5] . These are mainly meant to give some secondary orientation.

5.1.2 Sounds to turn

To stimulate the test subjects to turn, the sound of a small constantly barking dog [6] was chosen and mapped to the turn-to target. Any sound would have sufficed, but intuitively people would turn and look towards a barking dog and it appears plausible, that this dog may change its position, such giving more freedom in designing the experiment while staying somewhat plausible.

[3] https://www.freesound.org/people/balou82/sounds/184356/

[4] https://www.freesound.org/people/music.boy/sounds/119121/

[5] https://www.freesound.org/people/mario1298/sounds/155346/

[6] https://www.freesound.org/people/felix.blume/sounds/199261/ (This sound was modified to provide more regularity)

5.1.3 Sounds to walk

As a sound to stimulate the test persons to walk somewhere the sound of an ice cream truck [7] was chosen and mapped to the go-to target. Again, any sound would have sufficed but in addition to the dog, this seemed like a plausible choice.

5.2 Automated test sequence

To meet these requirements, an automation was designed, which leads the test subjects through all the tests of their experiment. This section starts with the conditions, the automation will have to meet and finally describes the sequence of the automation.

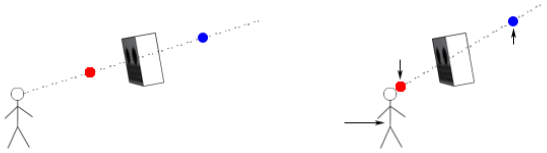
5.2.1 Conditions

One fundamental goal of the experiment design was to conduct all tests for a test subject in one piece without breaks in presences. At no time during the experiment, test subjects should use other cues but aural, vestibular or proprioceptive cues. Some tests however have certain demands towards the starting position and / or orientation of the test subject. A curvature gain test for example can not start with the test subject facing a wall. For these tests, prior to defining the individual starting position and orientation, a likely path of the test subject will have to be predicted. While rotation gains could theoretically be tested in any position and angle, some aspects led to a closer definition of the starting conditions. The height of virtual sound sources within this WFS system is not position preservative. While auditors move through the plane, the perceived elevation of the virtual sound sources is dependent of the proximity to the relevant speakers. A virtual sound source outside of the physical WFS area will always be "behind" one of the speakers (see figure 5a). The speaker can be seen as the angle point of a seesaw with the aural perception system of an auditor on one side and the virtual sound source on the other side, as illustrated in figure 5. While the slope of this seesaw in- and decreases with the proximity of the auditor to the relevant speaker, the perceived elevation of the virtual sound source also in- and decreases (see figure 5). The slope also changes, when the virtual sound source moves behind other speakers which are closer to or further away from the auditor. Since this effect presumably can give cues about the movement of virtual sound sources, rotation gain tests are defined to always take place in the center of the physical WFS area, where the variation of the distance between auditor and speaker, as well as between virtual sound source and speaker is least.

5.2.2 Path Prediction

For curvature gain tests, it is crucial to predict the path, the test subject is likely to take during the test. Even though the ideal path, given a starting position, starting orientation and curvature gain value, can be calculated, it is highly unlikely, that the test subject will take this ideal path, considering that the RDW technique is based on the principle of leading users to believe, that they strayed from the path. Amongst others, aspects like the precision of detecting the direction of the targeted virtual sound source and the responsiveness to the manipulations will

[7] https://www.freesound.org/people/ericstrausser/sounds/106238/ (This sound was cut, at the beginning and the end of the tune)



(a) Virtual sound sources are located on a straight defined by head and speaker.

(b) As the auditor is moving, the virtual sound sources seem to change their height.

Figure 5. The height of virtual sound sources is dependent on the relative position of the auditor to the relevant speakers. While the auditor moves closer to the speaker, the red virtual sound source will stay between the head and the speaker and such changing its height for this auditor.

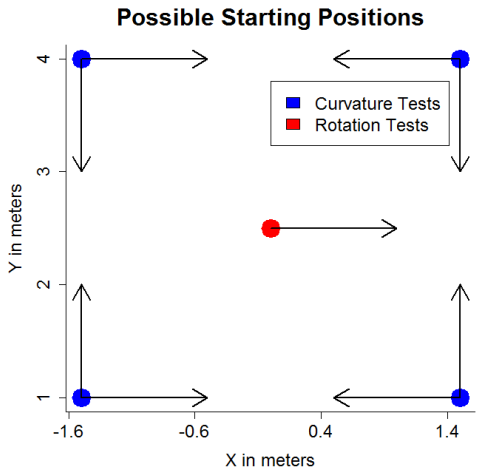


Figure 6. All possible starting positions for curvature gain and rotation gain tests with corresponding orientations.

influence the path the test subject is going to take. Minor manipulations will be calculated in a rapid succession, which are always dependent on the current position of the test subject and such, even the ideal path changes with every stray from the previous ideal path.

Even though the exact path, which the test subject is going to take can not be foreseen, an estimation can be made. The value of the curvature gain determines, in which direction the test subject will be redirected. Assumed the test subject is oriented towards the target before the test starts, it is highly unlikely, that she will veer into the opposite direction by an significant amount. So the path ahead and towards the direction of the manipulation should be free. These tests will therefore start in corners of the room with an orientation towards the adjacent corner, which opens the room towards the direction of the manipulation. Figure 6 illustrates the possible starting positions and orientations.

5.2.3 *Automation*

The automation is to ensure, that all test will be executed in a random order seamlessly and without any further instructions or interferences during the experiment. The state diagram in figure 7 illustrates the individual steps of the automation.

Choose Test - The automation follows an iterative pattern and for each test, the first step is always the randomized selection of a test, that had not been conducted yet.

Positioning - Depending on the type of test and the current position of the test subject, a starting position for the test is determined. Rotation gain tests will always be performed in the middle of the physical WFS area, while curvature gain tests will start in the closest corner. See figure 6 for all possible starting positions. The go-to target is then placed on the starting position for the test and its sound is activated. The sound will then keep running in a loop, until the test subject has reached a certain minimum proximity to the position. Upon arrival, the sound is turned of immediately and positioning is done.

Aligning - The starting alignment is also dependent on the type of test. Rotation gain tests always start with an alignment along the x-axis and curvature gain tests with an alignment, that allows the test subject to follow the manipulation towards the center of the room. See figure 6 for all possible starting alignments in regards to the starting position. The turn-to target is placed on a position in the proper direction and its sound is activated. The sound will then keep playing a loop, until a head orientation of the test subject had been captured, which is within a certain deviation to the calculated orientation towards the turn-to target. Then the sound will stop playing immediately and after a pause of two seconds, the aligning is done. This pause is to ensure, that the test subject has settled into that direction and to help distinguish between the sounds of the alignment and the following test.

Starting Tests - Before the manipulation can start, the corresponding virtual sound source has to be placed and activated. For curvature gain tests, the go-to target is placed at the position of the corner towards the test subject had just been oriented during the previous step. For rotation gain tests, the turn-to target is positioned outside of the physical WFS area in the opposite direction from the previous alignment. The next step is highly dependent on the type of test.

Curvature gain - For a curvature gain test, each decrease in distance towards the go-to target is multiplied by the value of the curvature gain to calculate the corresponding rotation of all virtual sound sources around the test subject. This state will be active, until the test subject has reached a distance to the starting position, which is higher, than the distance from the starting position to the go-to target. In this case, the test subject passed the go-to target, is most likely very close to the target and would now turn around to locate the exact position. Since this is not so much subject of the experiment and to speed up the process, the conditions are considered met.

Rotation gain - When a rotation gain test had been selected, all head rotations of the test subject will be multiplied with the value of the rotation gain and result in a corresponding rotation of all virtual sound sources around the test subject. This state will be active, until the difference between the orientation of the test subjects head and the angle towards the turn-to target is below a certain threshold. Then the manipulation will stop, but the sound will keep playing for another second, so the target does not just vanish during the rotation of the test subject.

Wrapup Test - The last step for each iteration is the wrap up. During the wrap up, the turn-to target and go-to target are muted, the test will be marked as done and removed from the to-do list and the to-do list is checked for more elements. If more tests need to be done, the iteration starts from the

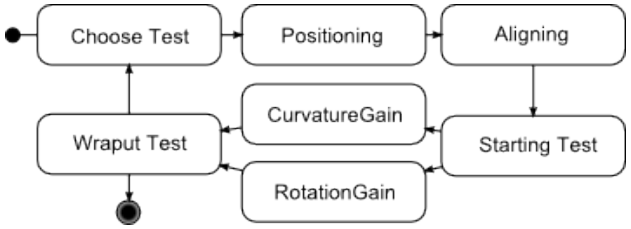


Figure 7. State diagram of the automated test sequence.

beginning and otherwise, the experiment will end.

5.3 Gain values

The choice of the values of the gains to be tested is based on related work by Steinicke et al. [6] and modified by a couple of self conducted tests, to get good limits and distribution in between. Ideally, the greatest manipulations should be noticed by all test subjects and the smallest by none, to cover the whole range.

For rotation gains the range is -60% <= g_R <= +60% with an increment of 5%. For curvature gains the range is -1.0 <= g_C <= +1.0, with $g_C := \frac{1}{r}$ and r being the radius of the circle that is defined by the curve, also with an increment of 0.05. Tests with a value of 0 are included four times each.

5.4 Test groups

The test subjects are to be divided into three major groups.

Group 1 are the *naive* test subjects. They do not know, that they are participating in a RDW experiment. Instead they are told, that the focus of the experiment is on the position preservation of focused sound sources. They will be asked to report whenever they feel, that a virtual sound source would move, even though it might just be a subjective perception.

Group 2 are the *aware* test subjects. They will be aware, that they are participating in a RDW experiment. They are instructed to report any notice of the virtual environment rotating around them. Presumably, this knowledge will influence the sensitivity of detecting the manipulations [10]. This is considered the most relevant group. Prospective users of an immersive virtual environment, which is using auditory RDW techniques, will most likely be aware, that these manipulations may occur.

Group 3 are the *expert* test subjects. They will know exactly, how the automation of the experiment is functioning. They will know the proportion of movements with manipulations to movements without manipulations and will most likely know in which situations a manipulation is likely to occur. All of them will have witnessed an experiment as a bystander and have an extensive discussion about the experiment beforehand. This group is considered the most challenging with presumably the highest detection rate.

The division into these groups will allow evaluation of the sensitivity to the manipulations regarding perceptibility in dependency to the knowledge of the system and situation.

6. IMPLEMENTATION

The system is implemented in Java 1.6 and based on the previously developed Motion Tracker-Wave Field Synthesis-Connector (MoWeC) [14]. This section will give a brief overview over the processing of the MoWeC, explaining the connection between the tracking system and the WFS system and how the RDW component had been integrated within.

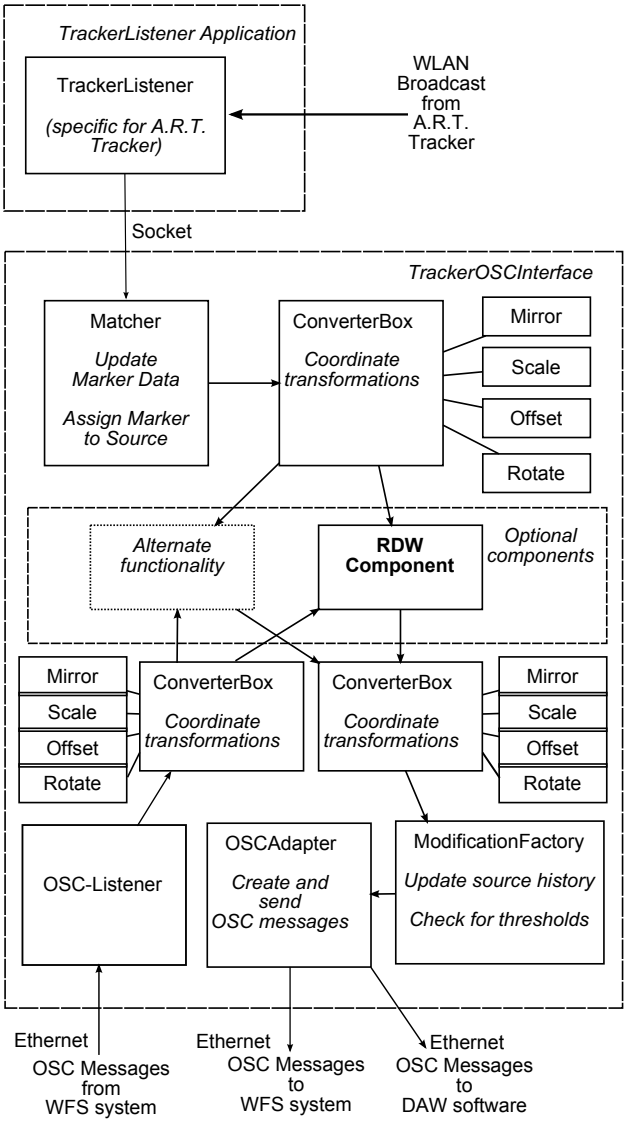


Figure 8. Processing sequence of the MoWeC.

6.1 Motion Tracker-Wave Field Synthesis-Connector (MoWeC)

The MoWeC is described in detail and in German language in [14]. Its processing sequence is illustrated in figure 8.

The MoWeC is driven by incoming network packages carrying tracking data, which is forwarded by the TrackerListener Application. This application provides a loose coupling between the MoWeC and its tracking system, while supplying the tracking data. The tracking data is then parsed and converted into the internal data type MoWeC source. The MoWeC is designed to work with these MoWeC sources.

The center of the MoWeC is defined by the optional components section. This is where the application logics can be implemented. Each optional component provides an application logic or a part of the overall application logic. Multiple

optional components can run at the same time, as long as they do not object each other logically. Optional components are modifying the MoWeC sources in respect of their position and orientation parameters etc..

After the optional component section, the MoWeC sources are converted into the target coordinate system of the WFS system. The changes are then transmitted to the WFS system and DAWs via open sound control (OSC) messages.

Simultaneously, the OSC listener receives all changes to the WFS system, which is converted into the internal coordinate system of the MoWeC and provided to the optional components.

6.2 Redirected walking component

As mentioned before, the RDW component is designed as an optional component for the MoWeC, as is illustrated in figure 8. As such, it is provided with tracking data from the tracking system and WFS data from the WFS system, both in form of MoWeC sources.

The tracking data from the head mounted target of the test subject provides the RDW component with position and orientation data for the head of the test subject. According to the currently defined gain, a rotation value is calculated as described in section 5.2.3. This rotation value will be used to calculate new coordinates and orientations for all other MoWeC sources, to give the impression, that the whole scene is rotating around the test subject. The MoWeC delivers these changes to the WFS system and the DAW in form of OSC messages.

The RDW component has its own graphical user interface (GUI). Starting this, the MoWeC's GUI is bypassed and a number of settings are predefined, which otherwise would have to be set manually. Processing can be started and stopped and different gains can be set manually. Further the activated gain is shown, logging and tracking can be (de-)activated, as well as the automated test started. The automated tests can be named for logging purposes. For the experiment at hand, manual (de-)activation of gains is not necessary however. By starting the automated test, gains will be chosen and applied automatically.

7. CONCLUSIONS

Methods have been developed, to identify thresholds for the detection of RDW manipulations via auditory cues by a WFS system. Even though general RDW had increasing attention, the auditory aspect was mostly neglected. This paper describes the preparation for an experiment to identify detection thresholds for a selection of RDW methods.

Following this paper, the experiment will be conducted with approximately 30 test subjects to identify thresholds depending of various factors like knowledge of the system, time in the system, walking and turning speed.

8. REFERENCES

[1] S. Razzaque, Z. Kohn, and M. C. Whitton, "Redirected Walking," Chapel Hill, NC, USA, Tech. Rep., 2001.

[2] J. Dichgans and T. Brandt, "Visual-Vestibular Interaction: Effects on Self-Motion Perception and Postural Control," in *Perception*, ser. Handbook of Sensory Physiology, R. Held, H. Leibowitz, and H.-L. Teuber, Eds. Springer Berlin Heidelberg, 1978, vol. 8, pp. 755–804.

[3] J. Lackner, "Induction of illusory self-rotation and nystagmus by a rotating sound-field," *Aviation, space, and environmental medicine*, vol. 48, no. 2, pp. 129–131, February 1977.

[4] C. Neth, J. Souman, D. Engel, U. Kloos, H. Bülthoff, and B. Mohler, "Velocity-Dependent Dynamic Curvature Gain for Redirected Walking," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 7, pp. 1041–1052, July 2012.

[5] F. Steinicke, G. Bruder, L. Kohli, J. Jerald, and K. Hinrichs, "Taxonomy and Implementation of Redirection Techniques for Ubiquitous Passive Haptic Feedback," in *Cyberworlds, 2008 International Conference on*, Sept 2008, pp. 217–223.

[6] F. Steinicke, G. Bruder, K. Hinrichs, J. Jerald, H. Frenz, and M. Lappe, "Real walking through virtual environments by redirection techniques," *JVRB*, vol. 6(2009), no. 2, 2009.

[7] F. Steinicke, T. Ropinski, G. Bruder, K. Hinrichs, H. Frenz, and M. Lappe, "A Universal Virtual Locomotion System: Supporting Generic Redirected Walking and Dynamic Passive Haptics within Legacy 3D Graphics Applications," in *Virtual Reality Conference, 2008. VR '08. IEEE*, March 2008, pp. 291–292.

[8] S. Serafin, N. Nilsson, E. Sikstrom, A. De Goetzen, and R. Nordahl, "Estimation of Detection Thresholds for Acoustic Based Redirected Walking Techniques," in *Virtual Reality (VR), 2013 IEEE*, March 2013, pp. 161–162.

[9] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe, "Analyses of Human Sensitivity to Redirected Walking," in *Proceedings of the 2008 ACM Symposium on VRST*, ser. VRST '08. New York, NY, USA: ACM, October 2008, pp. 149–156.

[10] ——, "Estimation of Detection Thresholds for Redirected Walking Techniques," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 1, pp. 17–27, Jan 2010.

[11] E. B. Goldstein, *Sensation and Perception*. Boston, MA: Cengage Learning, 2010.

[12] F. Steinicke, G. Bruder, T. Ropinski, K. H. Hinrichs, H. Frenz, and M. Lappe, "Generic Redirected Walking & Dynamic Passive Haptics: Evaluation and Implications for Virtual Locomotion Interfaces," in *Proceedings of IEEE Symposium on 3DUI (Poster Presentation)*. IEEE Press, 2008, pp. 147–148.

[13] W. Fohl, "The wave field synthesis lab at the haw hamburg," in *Sound-Perception-Performance*. Springer, 2013, pp. 243–255.

[14] M. Nogalski, "Gestengesteuerte Positionierung von Klangquellen einer Wellenfeldsynthese-Anlage mit Hilfe eines kamerabasierten 3D-Tracking-Systems," Bachelor Thesis, Hamburg University of Applied Sciences, 2012.

Guided improvisation as dynamic calls to an offline model

Jérôme Nika[1,2], **Dimitri Bouche**[1,2], **Jean Bresson**[1,2], **Marc Chemillier**[3], **Gérard Assayag**[1,2]
[1] IRCAM, UMR STMS 9912 CNRS, [2] Sorbonne Universités UPMC,
[3] Cams, Ecole des Hautes Etudes en Sciences Sociales
{jnika,bouche,bresson}@ircam.fr, chemilli@ehess.fr, assayag@ircam.fr

ABSTRACT

This paper describes a reactive architecture handling the hybrid temporality of guided human-computer music improvisation. It aims at combining reactivity and anticipation in the music generation processes steered by a "scenario". The machine improvisation takes advantage of the temporal structure of this scenario to generate short-term anticipations ahead of the performance time, and reacts to external controls by refining or rewriting these anticipations over time. To achieve this in the framework of an interactive software, guided improvisation is modeled as embedding a compositional process into a reactive architecture. This architecture is instantiated in the improvisation system ImproteK and implemented in OpenMusic.

1. INTRODUCTION

Human-computer improvisation systems generate music on the fly from a model and external inputs (typically the output of an "analog" musician's live improvisation). Introducing authoring and control in this process means combining the ability to react to dynamic controls with that of maintaining conformity to fixed or dynamic specifications.

This paper describes an architecture model (instantiated in the improvisation software ImproteK) where the specification is a formal structure, called *scenario*, that has to be followed during the performance. The scenario enables to anticipate the future and to generate music ahead of the current time. In this context, the system reactions to changes and external controls should use this knowledge of what is expected to generate an updated future. In addition, if the initial specification itself gets modified during the performance, the system may have to ensure a continuity with the past generated material at critical times.

To achieve this, a *scenario/memory* generation model is embedded in a reactive agent called *improvisation handler* which translates dynamic controls from the environment into music generation processes. This agent is in continuous interaction with an *improvisation renderer* designed as a scheduling module managing the connection with the real performance time.

Copyright: ©2015 Jérôme Nika 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.

After briefly discussing the existing approaches in guided improvisation systems in section 2, section 3 describes the scenario/memory generation model and gives some musical directions related to guided (or composed) improvisation. Then, section 4 describes how anticipation and dynamic controls are combined by embedding this generation model into the reactive improvisation handler. Finally, section 5 presents the improvisation renderer interweaving calls to the improvisation handler, scheduling and rendering of the generated material.

2. RELATED WORK

A number of existing improvisation systems drive the music generation processes by involving a user steering their parameters. In a first approach this user control can concern (low-level) system-specific parameters. This is for example the case of OMax [1, 2] or Mimi4x [3].

We refer here to *guided improvisation* when the control on music generation follows a more "declarative" approach, i.e. specifying targetted outputs or behaviours using an aesthetic, musical, or audio vocabulary independent of the system implementation. On the one hand, *guiding* is seen as a purely reactive and step by step process. SoMax [4] for instance translates the musical stream coming from an improviser into activations of specific zones of the musical memory in regards to a chosen dimension (for example the harmonic background). VirtualBand [5] and Reflexive Looper [6] also extract multimodal observations from the musician's playing to retrieve the most appropriate musical segment in the memory in accordance to previously learnt associations.

On the other hand, *guiding* means defining upstream temporal structures or descriptions driving the generation process of a whole improvisation sequence. Pachet and Roy [7] for instance use constraints in such generation process. In the works of Donzé *et al.* [8] the concept of "control improvisation" [9] applied to music also introduces a guiding structure via a reference sequence and a number of other specifications. This structure is conceptually close to the *scenario* used in our approach. PyOracle [10] proposes to create behaviour rules or scripts for controlling the generation parameters of an improvisation generation using "hot spots" (single event targets). Wang and Dubnov [11] extend this work in an offline architecture using sequences instead of single events as query targets. This idea of mid-term temporal queries is close to the musical issues raised with the notion of *dynamic scenario* in this paper.

Two conceptions of time and interactions are actually emphasized in these different approaches. The purely reactive one offers rich interaction possibilities but does not integrate prior knowledge about the temporal evolution. On the other hand, steering music generation with mid- or long-term structures enables anticipation but lacks reactivity with regard to external or user controls.

This bi-partition in improvisation systems actually reflects the offline/online paradigmatic approaches in computer music systems regarding time management and planning/scheduling strategies. On the one hand, "offline" corresponds to computer-aided composition systems [12] where musical structures are computed following best effort strategies and where rendering involves static timed plans (comparable to timed priority queues [13]). In this case, the scheduling only consists in traversing a precomputed plan and triggering function calls on time. On the other hand, "online" corresponds to performance-oriented systems [14] where the computation time is part of the rendering, that is, computations are triggered by clocks and callbacks and produce rendered data in real-time [15]. In this case, only scheduling strategies matter and no future plan is computed.

3. GUIDED MUSIC IMPROVISATION AND DYNAMIC CONTROLS

Our objective is to devise an architecture at an intermediate level between the reactive and offline approaches for guided improvisation, combining dynamic controls and anticipations relative to a predefined plan.

The proposed architecture is structured around an offline generation module based on a scenario, embedded in a reactive framework and steered/controlled by external events. This generation module produces short-term anticipations matching its scenario, and may eventually rewrite these anticipations over time according to incoming control events.

3.1 Scenario/memory generation model

The offline generation process consists in finding a path matching the scenario through a structured and labeled memory, where:

- the *scenario* is a symbolic sequence of labels defined over an alphabet,
- the *memory* is a sequence of musical contents labeled by letters of the same alphabet.

The scenario can be any sequence defined over an arbitrary alphabet depending on the musical context, for example a harmonic progression in the case of jazz improvisation or a discrete profile describing the evolution of audio descriptors for the control of sound synthesis processes. The memory can be constituted online (recorded during a live performance) or offline (from annotated audio or MIDI files).

This model is used for example to improvise on a given chord chart using a memory constituted by live inputs and heterogenous set of jazz standards recordings as memory. The contents of the memory can be audio slices [1], MIDI notes or events [2], parameters for sound synthesis, etc.

The scenario/memory model is implemented in ImproteK [16, 17], a co-improvisation system inheritor of the software environment OMax, which specifically addresses the issues of authoring and control in human-computer improvisation. This generation model is a priori offline in the sense that one run produces a whole timed and structured musical gesture satisfying the designed scenario, which will then be unfolded through time during performance. Furthermore, it follows a compositional workflow: 1) chose or define an alphabet for the labels and describe its properties, 2) compose at the structure level (i.e. define a scenario).

3.2 In-time reaction

In the scope of music improvisation guided by a scenario, a *reaction* of the system to dynamic controls cannot be seen as a spontaneous instant response. The main interest of using a scenario is indeed to take advantage of this temporal structure to anticipate the music generation, that is to say to use the prior knowledge of what is expected for the future in order to better generate at the current time. Whether a reaction is triggered by a user control, by hardcoded rules specific to a musical project, or by an analysis of the live inputs from a musician, it can therefore be considered as a revision of the mid-term anticipations of the system in the light of new events or controls.

To deal with this temporality in the framework of a real-time interactive software, we consider guided improvisation as embedding an offline process into a reactive architecture. In this view, reacting amounts to composing a new structure in a specific timeframe ahead of the time of the performance, possibly rewriting previously generated material. The synchronization with the environment and the management of high-level temporal specifications are handled by a dynamic score launching the calls to the generation model. This module, using the environment Antescofo and the associated programming language [18, 19], is described in [16].

In this paper, we focus on handling these reactive calls to combine anticipation relative to the scenario and dynamic controls, by proposing an architecture made of two main components:

- An *improvisation handler* embedding the scenario/ memory articulations and generating musical sequences on request.
- An *improvisation renderer* handling the temporality and interactions in a system run.

After mentioning some musical issues raised by guided improvisation, we will introduce these two agents (both implemented in the OpenMusic [20] environment) in more details and formalize their interactions in sections 4 and 5.

[1] **See videos: `http://repmus.ircam.fr/nika/ImproteK`**
[2] **See videos: `http://improtekjazz.org`**

3.3 Playing with the scenario and with the dynamic controls

The articulation between the formal abstraction of "scenario" and reactivity enables to explore different musical directions with the same objects and mechanisms, providing dynamic musical control over the improvisation being generated.

In first approach, we differentiate two playing modes depending on the hierarchy between the musical dimension of the scenario and that of control. When scenario and control are performed on different features of the musical contents (3.3.1), the model combines long-term structure with local expressivity. When scenario and dynamic controls act on the same musical feature (3.3.2), it deals with dynamic guidance and intentionality.

3.3.1 Long-term structure and local expressivity

We firstly consider the case where the specification of a scenario and the reaction concern different features, conferring them different musical roles (for example: defining the scenario as a harmonic progression and giving real-time controls on density, or designing the scenario as an evolution in register and giving real-time controls on energy). In this case, a fixed scenario provides a global temporal structure on a conduct dimension, and the reactive dimension enables to be sensitive to another musical parameter. The controlled dimension has a local impact, and deals with expressivity by acting at a secondary hierarchical level, for example with instant constraints on timbre, density, register, syncopation etc.

This playing mode may be more relevant for idiomatic [21] or composed improvisation with any arbitrary vocabulary, in the sense that a predefined and fixed scenario carries the notions of high-level temporal structure and formal conformity to a given specification anterior to the performance, as it is the case for example with a symbolic harmonic progression.

3.3.2 Guidance and intentionality

When specification and reaction act on the same musical dimension, the scenario becomes dynamic. A reaction does not consist in a local control on a secondary parameter as in the previous playing mode, but in the modification of the scenario itself.

In this case, the current state of a dynamic scenario at each time of the performance represents the short-term "intentionality" attributed to the system, which becomes a reactive tool to guide the machine improvisation by defining instant queries with varying time windows. The term "scenario" may be inappropriate in this second approach since it does not represent a fixed general plan for the whole improvisation session. Nevertheless, we will use this term in the following sections whether the sequence guiding the generation is dynamic or static (i.e. whether the reaction impacts the guiding dimension or another one) since both cases are formally managed using the same mechanisms.

4. EMBEDDING AN OFFLINE GENERATION MODEL INTO A REACTIVE ENVIRONMENT

Thanks to the scenario, music is produced ahead of the performance time, buffered to be played at the right time or rewritten. For purposes of brevity (and far from any anthropomorphism),

- *anticipations* will be used to refer to pending events: the current state of the already generated musical material ahead of the performance time,
- *intentions* will be used to refer to the planned formal progression: the current state of the scenario and other generation parameters ahead of the performance time.

This section presents how the evolving anticipations of the machine result from successive or concurrent calls to the generation model. Introducing a reaction at a time when a musical gesture has already been produced amounts then to rewrite buffered anticipation. The rewritings are triggered by modifications of the intentions regarding the scenario itself or other generation parameters (these two different cases correspond to the different musical directions introduced in 3.3).

4.1 Improvisation handler

To give control over these mechanisms, that is dynamically controlling improvisation generation, we define an *improvisation handler* agent (H) which contains and articulates the generation model with:

- a scenario (S);
- a set of generation parameters;
- current position in the improvisation t^p (*performance time*);
- the index of the last generated position t^g (*generation time*);
- a function f responsible for the output of generated fragments of improvisation (*output method*).

This improvisation handler agent H links the real time of performance and the time of the generation model embedded in an *improviser* structure (see Figure 1). The improviser structure associates the generation model and the memory with a set of secondary generation parameters and an execution trace described below.

The set of *generation parameters* contains all the parameters driving the generation process which are independent from the scenario: parametrization of the generation model (e.g. minimal / maximal length or region of the sub-sequences retrieved from the memory, measure of the linearity/non-linearity of the paths in the memory etc.) and content-based constraints to filter the set of possible results returned by the scenario matching step (e.g. user-defined thresholds, intervals, rules etc.).

The *execution trace* records history of paths in the memory and states of these generation parameters for the last runs of the generation model so that coherence between

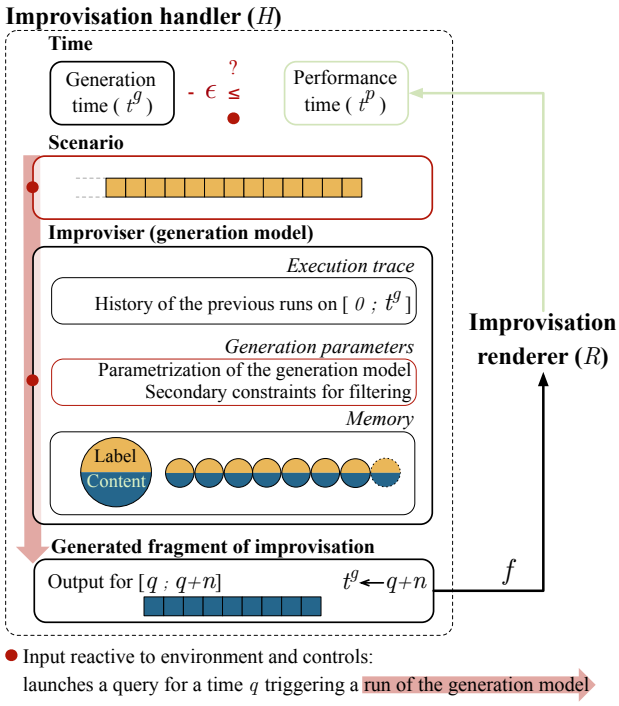


Figure 1. *Improvisation handler* agent.

successive generations phases associated to overlapping queries is maintained. This way, the process can go back to an anterior state to insure continuity at the first position where the generation phases overlap.

The interactions of the improvisation handler with the environment consist in translating dynamic controls on reactive inputs into reactive queries and redirecting the resulting generated fragments. We call *reactive inputs* the entities whose modifications lead to a reaction: the scenario and the set of generation parameters. In this framework, we call *reaction* an alteration of the *intentions* leading to a call to the generation model to produce a fragment of improvisation starting at a given position in the scenario.

We note Q a *query* launched by a reaction to generate an improvisation fragment starting at time q in the scenario [3]. Q triggers a run of the improviser to output a sub-sequence (or a concatenation of sub-sequences) of the memory which:

- matches the current state of the scenario from date q (i.e. a suffix S_q of the scenario, see [17]),
- satisfies the current state of the set of generation parameters.

The *output method* of the improvisation handler (f) is a settable attribute, so that generated improvisations can be redirected to any rendering framework. For instance, the improvisation handler can interface with Max via the dynamic score written in the Antescofo language mentioned in 3.2. In this case, f determines how resulting improvisation segments are sent back to the dynamic score where they are buffered or played in synchrony with the non-metronomic tempo of the improvisation session. Section 5 details how f is used to couple the improvisation handler with an improvisation renderer in order to unify music generation, scheduling and rendering.

[3] q is the time at which this fragment will be played, it is independent from t^p and from the date at which the query is launched by the improvisation handler.

4.2 Triggering queries for rewriting anticipations

We describe here the way control events are translated into generation queries triggered by the improvisation handler. This mechanism can be *time-triggered* or *event-triggered*, i.e. resulting respectively from depletion of previously generated material or from parameters modifications.

4.2.1 Time-triggered generation

Rendering may lead to the exhaustion of generated improvisation. New generation queries have therefore to be launched to prevent the time of the generation t^g from being reached by the time of the performance t^p. To do so, we define ϵ as the maximum allowed margin between t^p and t^g. Consequently, a new query for time $q = t^g + 1$ is automatically triggered when the condition $t^g - t^p \leq \epsilon$ becomes true.

Depletion of the previously generated improvisation generation occurs when generation over the whole scenario is not performed in a single run. Figure 2 illustrates two successive generation phases associated to queries Q_1 and Q_2 for time q_1 and q_2 respectively. A generation *phase* matches a scenario sub-sequence starting at a queried position q to a sub-sequence of the memory, i.e. the generation model searches for a prefix of the suffix S_q of the scenario S in the memory (phase q_1 in figure 2) or an equivalent non-linear path (phase q_2 in figure 2). The generation process waits then for the next query.

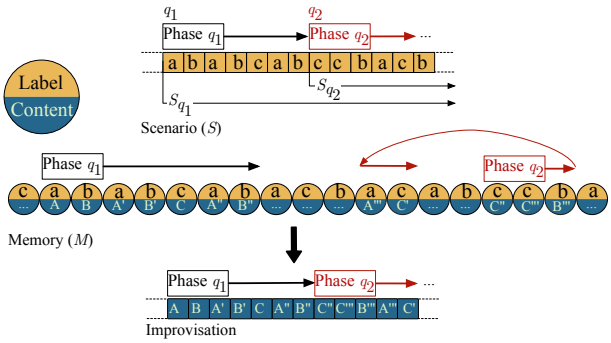


Figure 2. Phases of the guided generation process.

Defining such phases enables to have mid-term anticipations generated ahead of the performance time while avoiding generating over the whole scenario if an event modifies the intentions.

A generated fragment of improvisation resulting from a query Q for time q contains n slices where:

$$1 \leq n \leq length(S) - q, n \in \mathbb{N}$$

The search algorithm of the generation model runs a generation phase [4] to output a sub-sequence of the memory in time $\Theta(m)$ and does not exceed $2 * m - 1$ comparisons, where m is the length of the memory. In first approximation, the maximum margin ϵ is empirically initialized with a value depending on the initial length m. Then, in order to take into account the linear time complexity, ϵ increases proportionally to the evolution of m if the memory grows as the performance goes. Future works on this point will consist in informing the scheduling engine with the similarities between the scenario and the memory to optimize anticipation. Indeed, the number of calls to the model depends on the successive lengths n of the similar patterns between the scenario and the memory. For example, the shorter the common factors, the higher the number of queries necessary to cover the whole scenario.

4.2.2 Event-triggered generation

As introduced previously, the musical meanings of a reaction of the improvisation handler impacting the scenario itself (3.3.2) or an other musical dimension (3.3.1) are quite different. Yet, both cases of reaction can be formally managed using the same mechanisms of event-triggered generation. The reactive inputs (4.1) are customizable so that any relevant slot of the improvisation handler can easily be turned into a reactive one. Modifying the scenario or one of these reactive slots launches a generation query for the time q affected by this modification. The triggering of a query by a reaction can indeed take effect at a specified time q independent of performance time t^p.

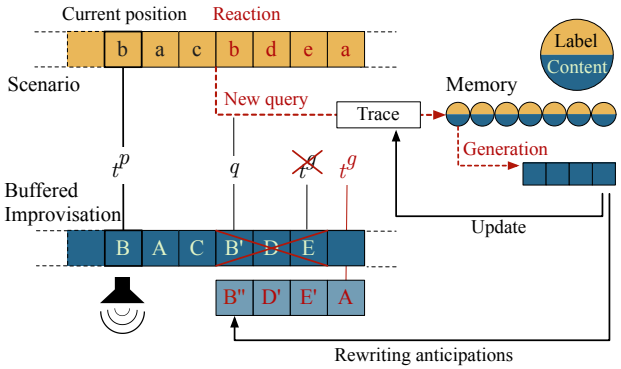


Figure 3. Reactive calls to the generation model.

As illustrated in figure 3, the new improvisation fragment resulting from the generation is sent to the buffered improvisation while the improvisation is being played.The new fragments overwrites the previously generated material on the overlapping time interval. The execution trace introduced in 4.1 enables to set mechanisms providing continuity at the tiling time q.

4.3 Rewriting intentions: concurrent queries

Anticipation may be generated without ever being played because it may be rewritten before being reached by the time of the performance. Similarly, an intention may be defined but never materialized into anticipation if it is changed or enriched by a new event before being reached by a run of generation.

Indeed, if reactions are frequent or defined with delays, it would be irrelevant to translate them into as many independent queries leading to numerous overlapping generation phases. We then define an intermediate level to introduce evolving queries, using the same principle for dynamically rewriting intentions as that defined for anticipations.

This aspect is dealt with by handling concurrency and working at the query level when the improvisation handler receives new queries while previous ones are still being processed by the generation module. Algorithm 1 describes how concurrency is handled, with:

- Run(Q): start generation associated to Q. This function outputs generated data when it finishes,
- Kill(Q): stop run associated to Q and discard generated improvisation,
- Relay(Q_1,Q_2,q): output the result of Q_1 for $[q_1; q[$, kill Q_1 and run Q_2 from q. The execution trace is read to maintain coherence at relay time q,
- WaitForRelay(Q_1,Q_2,q): Q_2 waits until Q_1 generates improvisation [5] at time q. Then Relay(Q_1,Q_2,q).

Algorithm 1 Concurrent runs and new incoming queries

Q_i, query for improvisation time q_i
RQ, set of currently running or waiting queries
CurPos(Q), current generation index of Run(Q)

when new Q received **do**
1: **for** $Q_i \in RQ$ **do**
2: **if** $q = q_i$ **then**
3: **if** Q and Q_i from same inputs **then**
4: Kill(Q_i)
5: **else**
6: Merge Q and Q_i
7: **end if**
8: **else if** $q > q_i$ **then**
9: **if** $q <$ *CurPos* (Q_i) **then**
10: Relay(Q,Q_i,q)
11: **else**
12: WaitForRelay(Q, Q_i, q)
13: **end if**
14: **else if** $q < q_i$ **then**
15: WaitForRelay(Q_i, Q, q_i)
16: **end if**
17: **end for**

This way, if closely spaced in time queries lead to concurrent processing, relaying their runs of the generation model at the right time using the execution trace enables to merge them into a "dynamic query".

[4] 1) Index the prefixes of the suffix S_q of the scenario in the memory, 2) select one of these prefixes depending on the generation parameters, 3) output this prefix or an equivalent non-linear path.

[5] More precisely, new generation phases are launched if needed until q is reached.

5. RENDERING AND SCHEDULING: IMPROVISATION RENDERER

The reactive architecture presented in the previous section embeds the data, specifications, and mechanisms managing music generation, reaction, and concurrency in the *improvisation handler*. This improvisation handler receives from the environment: (1) the live inputs (in the case of online learning), (2) the control events, and (3) the current performance time t^p. In return, it sends improvisation fragments back to the same environments (4).

Live inputs and interaction (1,2) are managed autonomously by the improvisation handler. The connection with the real performance time (3,4) is managed in an unified process through the continuous interaction with an *improvisation renderer*. This renderer is designed as a scheduling module which runs in parallel to the improvisation handler all along the performance.

5.1 Scheduling strategy

To describe the scheduling architecture we need to introduce a number of additional concepts: an *action* is a structure to be executed (including a function and some data); a *plan* is a list of ordered timed actions; the *planner* is an entity in charge of extracting plans from musical structures; and the *scheduler* is the entity in charge of rendering plans.

A hierarchical model is used to represent musical data (for example, a chord as a set of notes, a note as a set of MIDI events...) and to synchronize datasets rendering. To prepare a musical structure rendering, the planner converts the object into a list of timed actions. Triggering the rendering of a "parent object" synchronizes the rendering of its "children" [6] . Then, the scheduler renders the plan, i.e. triggers the actions on time [24].

The planner and scheduler cannot operate concurrently on a same plan, but they can cooperate. Scheduling is said *dynamic* when the scheduler is likely to query the planner for short-term plans on the fly, and/or when the planner is likely to update plans being rendered by the scheduler [25, 26]. Our strategy is based on a *short-term lookahead* planner: instead of planning a list of actions representing the whole content of a musical object, the planner is called on-time by the scheduler and outputs plans applicable in a specified time window.

The flowchart on figure 4 summarizes the plan extraction algorithm used by the scheduler to render musical objects. Typically, the scheduler calls the planner for a plan applicable in a time window W of duration w, then the scheduler can render this short-term plan on time and query the planner for the next one. The lower w, the most reactive the system is, at a cost of more computations (w can be tweaked accordingly). If the planner returns no plan (i.e. there is nothing to render in the queried time interval), the scheduler can query again for the next time window until a plan is returned. Therefore, the time window W can be far ahead of the actual rendering time of the structure,

[6] As introduced in [22]. In terms of scheduling, the hierarchical representations also eases the development of optimized strategies [23].

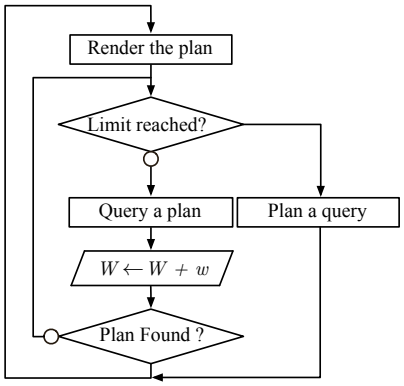


Figure 4. Short-term plan extraction flowchart.

and might not be the same across concurrently rendered objects. Plan queries themselves can also be planned as actions to execute in the future. For instance, a limit of successive plan queries can be set to avoid overload (e.g. if there is nothing else to play): in this case sparse planning queries can be planned at the end of each time windows.

5.2 Application for guided improvisation

The *improvisation renderer* (R) connected to the improvisation handler:

- receives and renders the produced fragments,
- communicates the current performance time t^p.

With regard to the scheduling architecture, R is a structure containing two children objects, the mutable priority queues:

- RC (*render action container*) containing actions to render, extracted from improvisation fragments.
- HC (*handler action container*) containing time marker actions to send back to the handler H.

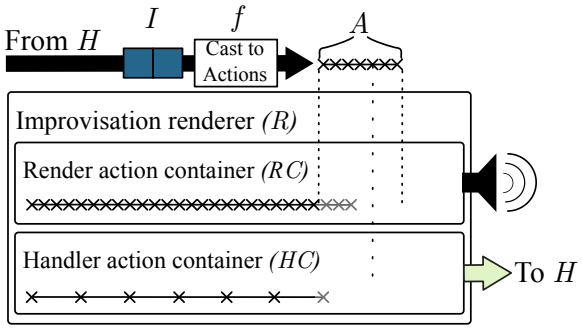


Figure 5. The *improvisation renderer*.

Figure 5 depicts the improvisation renderer and its communication with the improvisation handler. An improvisation fragment I is outputted from the improvisation handler, and this fragment is casted into a list of actions A integrated into the render action container RC. This translation can be defined depending on the type of improvised

data. For instance, if the improvisation slices contain MIDI, actions will consist in calls to *midi-on* and *midi-off*. If the list of actions is overlapping with existing content (i.e. with previously generated fragments of improvisation already stored as actions in RC), the new actions substitute the overlap and add the last generated improvisation to RC. At the same time, information about slices timing of I is extracted to feed the handler action container HC with time markers that will be sent back on time to the improvisation handler.

To perform the previous operations, we define the following functions:

- $Cast(I)$: cast an improvisation fragment I into a list of timed actions A,
- $Timing(I)$: extract a list of actions from an improvisation fragment I, corresponding to the slices' respective times,
- $Tile(C, A)$: integrate an action list A in the action container C, overwriting the overlapping actions.

In order to connect the improvisation handler (section 4.1) to the improvisation renderer, the output method f of the improvisation handler shall therefore be the function of an improvisation fragment I and the improvisation renderer R defined as:

$$f(I, R) = \begin{cases} Tile(RC, Cast(I)) \\ Tile(HC, Timing(I)) \end{cases}$$

R can then be planned and rendered as a standard musical object, although this object will be constantly growing and changing according to performer's inputs or user controls. The short-term planning strategy will allow for changes not to affect the scheduled plans if they concern data at a time ahead of the performance time by at least w. In the contrary case (if data is modified inside the current time window) a new short-term plan extraction query is immediately triggered to perform a re-planning operation.

6. CONCLUSIONS AND PERSPECTIVES

We presented an architecture model to combine dynamic controls and anticipation with respect to a formal structure for guided human-computer music improvisation. It is achieved by embedding offline processes into a reactive framework, out of the static paradigm yet not using pure last moment computation strategies. It results in a hybrid architecture dynamically rewriting its musical output ahead of the time of the performance, in reaction to the alteration of the scenario or of a reactive parameter. The generation model and the improvisation handler agent are customizable and designed in such a way that it can be easily interfaced with any rendering environment (for example Antescofo/Max or OpenMusic). This architecture is instantiated in the improvisation software ImproteK.

In the scope of guided improvisation, the reactive architecture described in this paper proposes a model to answer the question "how to react?", but does not address the question "when to react and with what response?". Indeed, the model defines the different types of reactions that have to be handled and how it can be achieved. It choses to offer genericity so that reactions can be launched by an operator using customized parameters, or by a composed reactivity defining hardcoded ruled specific to a particular musical project. Yet, integrating approaches such as that developed in [4] could enable to have reactions launched from the analysis of live musical inputs.

Considering the genericity of the alphabets in the generation model and that of the reactive architecture presented in this paper, future works will focus on chaining agents (using the output of an improvisation handler as an input for an other). A preliminary study of such chaining will involve using an ascending query system through the tree of plugged units to avoid data depletion, and message passing scheduling between multiple agents [27] to ensure synchronization. Other perspectives suggest to make use of such reactive music generation to produce evolving and adaptive soundscapes, embedding it in any environment that generates changing parameters while including a notion of plot, as video games for example.

Acknowledgments

The authors would like to thank José Echeveste and Jean-Louis Giavitto for fruitful discussions. This work is supported by the French National Research Agency projects EFFICACe ANR-13-JS02-0004 and DYCI2 ANR-14-CE2 4-0002-01.

7. REFERENCES

[1] G. Assayag, G. Bloch, M. Chemillier, A. Cont, and S. Dubnov, "OMax brothers: a dynamic topology of agents for improvization learning," in *1st ACM workshop on Audio and music computing multimedia*, Santa Barbara, CA, USA, 2006, pp. 125–132.

[2] B. Lévy, G. Bloch, and G. Assayag, "OMaxist dialectics," in *12th International Conference on New Interfaces for Musical Expression*, Ann Arbor, MI, USA, 2012, pp. 137–140.

[3] A. R. François, I. Schankler, and E. Chew, "Mimi4x: an interactive audio–visual installation for high–level structural improvisation," *International Journal of Arts and Technology*, vol. 6, no. 2, pp. 138–151, 2013.

[4] L. Bonnasse-Gahot, "An update on the SOMax project," *Ircam - STMS, Internal report ANR project Sample Orchestrator 2, ANR-10-CORD-0018*, 2014.

[5] J. Moreira, P. Roy, and F. Pachet, "Virtualband: Interacting with Stylistically Consistent Agents," in *14th International Society for Music Information Retrieval*, Curitiba, Brazil, 2013, pp. 341–346.

[6] F. Pachet, P. Roy, J. Moreira, and M. d'Inverno, "Reflexive Loopers for Solo Musical Improvisation," in *14th SIGCHI Conference on Human Factors in Computing Systems*, Paris, France, 2013, pp. 2205–2208.

[7] F. Pachet and P. Roy, "Markov constraints: steerable generation of markov sequences," *Constraints*, vol. 16, no. 2, pp. 148–172, 2011.

[8] A. Donzé, R. Valle, I. Akkaya, S. Libkind, S. A. Seshia, and D. Wessel, "Machine improvisation with formal specifications," in *40th International Computer Music Conference*, Athens, Greece, 2014, pp. 1277–1284.

[9] D. J. Fremont, A. Donzé, S. A. Seshia, and D. Wessel, "Control improvisation," *arXiv preprint arXiv:1411.0698*, 2014.

[10] G. Surges and S. Dubnov, "Feature selection and composition using pyoracle," in *9th Artificial Intelligence and Interactive Digital Entertainment Conference*, Boston, MA, USA, 2013.

[11] C. Wang and S. Dubnov, "Guided music synthesis with variable markov oracle," in *3rd International Workshop on Musical Metacreation*, Raleigh, NC, USA, 2014.

[12] G. Assayag, "Computer Assisted Composition Today," in *1st symposium on music and computers*, Corfu, Greece, 1998.

[13] M. Kahrs, "Dream chip 1: A timed priority queue," *IEEE Micro*, vol. 13, no. 4, pp. 49–51, 1993.

[14] R. Dannenberg, "Real-Time Scheduling and Computer Accompaniment," in *Current Directions in Computer Music Research*, Cambridge, MA, USA, 1989, pp. 225–261.

[15] P. Maigret, "Reactive Planning and Control with Mobile Robots," in *IEEE Control*, 1992, pp. 95–100.

[16] J. Nika, J. Echeveste, M. Chemillier, and J.-L. Giavitto, "Planning Human-Computer Improvisation," in *40th International Computer Music Conference*, Athens, Greece, 2014, pp. 1290–1297.

[17] J. Nika and M. Chemillier, "Improvisation musicale homme-machine guidée par un scénario temporel," *Technique et Science Informatique, Numéro Spécial Informatique musicale*, vol. 7, no. 33, pp. 651–684, 2015, (in french).

[18] J. Echeveste, A. Cont, J.-L. Giavitto, and F. Jacquemard, "Operational semantics of a domain specific language for real time musician-computer interaction," *Discrete Event Dynamic Systems*, vol. 4, no. 23, pp. 343–383, 2013.

[19] J. Echeveste, J.-L. Giavitto, and A. Cont, "A Dynamic Timed-Language for Computer-Human Musical Interaction," INRIA, Rapport de recherche RR-8422, 2013.

[20] J. Bresson, C. Agon, and G. Assayag, "OpenMusic. Visual Programming Environment for Music Composition, Analysis and Research," in *ACM MultiMedia 2011 (OpenSource Software Competition)*, Scottsdale, AZ, USA, 2011.

[21] D. Bailey, *Improvisation: its nature and practice in music*. Da Capo Press, 1993.

[22] X. Rodet, P. Cointe, J.-B. Barriere, Y. Potard, B. Serpette, and J.-P. Briot, "Applications and developments of the formes programming environment," in *9th International Computer Music Conference*, Rochester, NJ, USA, 1983.

[23] R. J. Firby, "An Investigation into Reactive Planning in Complex Domains," in *6th National Conference on Artificial Intelligence*, Seattle, WA, USA, 1987, pp. 202–206.

[24] D. Bouche and J. Bresson, "Planning and Scheduling Actions in a Computer-Aided Music Composition System," in *Scheduling and Planning Applications woRKshop, 25th International Conference on Automated Planning and Scheduling*, Jerusalem, Israel, 2015, pp. 1–6.

[25] M. E. desJardins, E. H. Durfee, J. Charles L. Ortiz, and M. J. Wolverton, "A Survey of Research in Distributed, Continual Planning," *AI Magazine*, vol. 20, no. 4, 1999.

[26] E. C. J. Vidal and A. Nareyek, "A Real-Time Concurrent Planning and Execution Framework for Automated Story Planning for Games," in *AAAI Technical Report WS-11-18*, 2011.

[27] J. Lee, M. J. Huber, E. H. Durfee, and P. G. Kenny, "UM-PRS: An implementation of the Procedural Reasoning System for multirobot applications," in *AIAA/NASA Conference on Intelligent Robotics in Field, Factory and Space*, Houston, TX, USA, 1994, pp. 842–849.

A LOOP SEQUENCER THAT SELECTS MUSIC LOOPS BASED ON THE DEGREE OF EXCITEMENT

Tetsuro Kitahara, Kosuke Iijima, Misaki Okada, Yuji Yamashita, and Ayaka Tsuruoka
College of Humanities and Sciences, Nihon University, Japan
{kitahara,iijima,misaki,yuji,tsuruoka}@kthrlab.jp

ABSTRACT

In this paper, we propose a new loop sequencer that automatically selects music loops according to the *degree of excitement* entered by the user. A loop sequencer is expected to be a good tool for non-musicians to compose music because it does not require expert musical knowledge. However, it is not easy to appropriately select music loops because a loop sequencer usually has a large scale of loop collection (e.g., more than 3000 loops). It is therefore necessary to automatically select music loops based on the user's simple and easy input. In this paper, we focus on the degree of excitement. In typical techno music, the temporal evolution of excitement is an important feature. Our system allows the user to enter the temporal evolution of excitement by drawing a curve, then the system automatically selects music loops according to the entered excitement. Experimental results show that our system is easy to understand and generates satisfying musical pieces for non-experts of music.

1. INTRODUCTION

A loop sequencer such as ACID PRO 7 or GarageBand is a popular tool for composing musical pieces. It enables the user to compose musical pieces by concatenating short musical materials called *music loops*. Because music loops are usually audio data, the user can easily compose high-quality pieces (e.g., without expert knowledge) as compared to inputting musical notes, one by one, in a MIDI sequencer.

A loop sequencer requires a huge number of music loops in order to enable users to compose a wide variety of music. For example, ACID PRO has more than 3,000 music loops. However, it is not easy for most users to listen to all 3,000 music loops, consider which loops match their inspiration, and then select the appropriate loops. It is therefore not common to enjoy composing music with a loop sequencer, even though expert musical knowledge is not required to use this tool.

There are many studies that investigate automatic music composition and computer-aided music composition. Fukayama et al. proposed a web-based automatic music composition system based on a probabilistic model [1].

Copyright: ©2015 Tetsuro Kitahara 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.

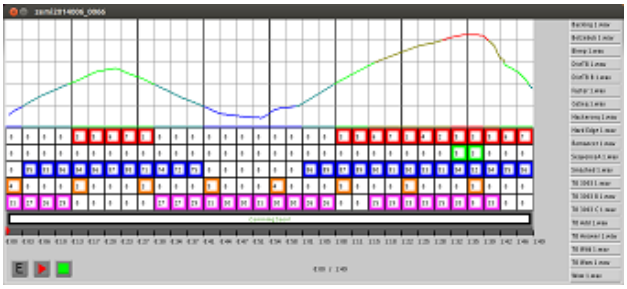


Figure 1. Screenshot of our system

Ando et al. proposed the use of interactive evolutional computation to automatically generating melodies [2]. In addition, many researchers have explored new automatic and/or computer-aided music composition methods [3]. However, all of these studies were aimed at automating and/or supporting note-level music composition, and thus they did not focus on audio-based music composition tools like a loop sequencer.

In this paper, we narrow the target to techno music, and propose a loop sequencer that automatically selects music loops according to the user's input of the desired *degree of excitement*. The temporal evolution of excitement is one of the most important features in techno music, and is expressed by what kinds of music loops are selected and how many. A typical piece of techno music may start with a simple repetition of the bass drum. Then, the excitement may increase as new loops (i.e., those of other drums, bass lines, and melodic lines) are added. This system enables users to input the temporal evolution of the excitement by drawing a curve, and then the system generates a musical piece that most matches the desired degree of excitement.

2. SYSTEM OVERVIEW

The screenshot of this system is shown in Figure 1. A list of music loops is shown in the right side of the screen. The panel for drawing an excitement curve is placed in the upper part of the screen, and the panel for displaying the selected music loops is placed in the lower part. The user first draws an excitement curve. Then, the system automatically determines how many loops will be placed and which loops will be placed at each measure. The user can remove, add, and change these loops at a later point.

3. METHOD FOR SELECTING MUSIC LOOPS

3.1 Formulation with hidden Markov model

The automatic selection of music loops is formulated using a hidden Markov model (HMM). For simplicity, we assume that all loops have the same length (one measure). A musical piece consists of several instrumental parts (e.g., drums, bass, and synth), and all music loops have been classified into the instrumental parts in advance. Let $\mathcal{M}_i$ be a set of music loops for the part i.

The problem of selecting music loops is to find the most likely $\boldsymbol{s}_n = (s_{n,1}, \cdots, s_{n,I})$ $(s_{n,i} \in \mathcal{M}_i \cup \{0\})$ for each measure n, where $s_{n,i}$ represents the music loop for the n-th measure of the part i. When we do not select any loops for the n-th measure of the part i, we denote this by $s_{n,i} = 0$. Our system supposes that a sequence of the excitement, $\boldsymbol{x} = [x_1, \cdots, x_N]$, is given. Therefore, the selection of music loops is formulated as:

$$\hat{\boldsymbol{S}} = \underset{\boldsymbol{S}}{\operatorname{argmax}}\, P(\boldsymbol{S}|\boldsymbol{x}),$$

where $\boldsymbol{S} = [\boldsymbol{s}_1, \cdots, \boldsymbol{s}_N]$. The excitement curve is freely drawn, so its value may vary within each measure. Therefore, we use the mean value of the excitement within the n-th measure for x_n.

Using Bayes' theorem, this equation is expanded to:

$$\hat{\boldsymbol{S}} = \underset{\boldsymbol{S}}{\operatorname{argmax}}\, P(\boldsymbol{x}|\boldsymbol{S})P(\boldsymbol{S}).$$

When we assume that x_n depends only on $\boldsymbol{s}_n$ and $\boldsymbol{s}_n$ depends only on $\boldsymbol{s}_{n-1}$, the equation becomes the following:

$$\hat{\boldsymbol{S}} = \underset{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_N}{\operatorname{argmax}} \prod_{n=1}^{N} P(x_n|\boldsymbol{s}_n) \cdot P(\boldsymbol{s}_1) \prod_{n=2}^{N} P(\boldsymbol{s}_n|\boldsymbol{s}_{n-1})$$

This equation is equivalent to the use of an HMM in which $\boldsymbol{x}$ and $\boldsymbol{S}$ are regarded as a sequence of observations and a sequence of state transitions, respectively. The most likely $\boldsymbol{S}$ can be determined with the Viterbi algorithm, if $P(x_n|\boldsymbol{s}_n)$, $P(\boldsymbol{s}_1)$, and $P(\boldsymbol{s}_n|\boldsymbol{s}_{n-1})$ are appropriately designed or trained.

3.2 Simplification of formulation

The above-mentioned formulation is difficult to directly apply because it includes a large number of parameters. We therefore simplify the formulation based on the following policies:

Policy 1 We discretize the degree of excitement, in other words, the degree of excitement is any of $1, \cdots, d$ ($d = 5$ in the current implementation).

Policy 2 We separately consider (1) whether a certain loop should be placed for the n-th measure of the part i, and (2) if so, which loop should be placed there. Also, we assume that (1) and (2) are independent.

Policy 3 We assume that which loop should be placed at each measure in each part is determined independently of other parts and/or other measures. However, we select the same loop within every eight measures in order to keep a consistent mood.

Policy 4 For every music loop, the degree of excitement of this loop itself is annotated as any of $1, \cdots, d$.

Based on Policy 2, we reformulate the problem by separately considering (1) whether a certain loop should be placed and (2) which loop should be placed. The random variables representing the former and latter are denoted by $q_{n,i} (\in \{0, 1\})$ and $s'_{n,i} (\in \mathcal{M}_i)$, respectively. Thus, $P(s_{n,i})$ can be expanded to:

$$\begin{aligned} P(s_{n,i}) &= P(q_{n,i},\ s'_{n,i}) \\ &= P(q_{n,i})\, P(s'_{n,i}|q_{n,i}) \\ &= P(q_{n,i})\, P(s'_{n,i}). \end{aligned}$$

Therefore, $P(x_n|\boldsymbol{s}_n)$ is expanded as follows:

$$P(x_n|\boldsymbol{s}_n) = \alpha P(x_n|\boldsymbol{q}_n) \prod_{i=1}^{I} P(x_n|s'_{n,i}),$$

where $\boldsymbol{q}_n = (q_{n,1}, \cdots, q_{n,I})$ and α is a constant, because $s'_{n,1}, \cdots, s'_{n,I}$ are independent according to Policy 3.

Here, $P(x_n|\boldsymbol{q}_n)$ models the relationship between the excitement and the number of parts that play music loops. In general, music becomes increasingly exciting as more loops are simultaneously played back. We manually design $P(x_n|\boldsymbol{q}_n)$ based on this idea.

On the other hand, $P(x_n|s'_{n,i})$ models the relationship between the excitement and the music loop itself. According to Policy 4, the music loop $s'_{n,i}$ has an annotation of its excitement $X(s'_{n,i})$. We therefore design $P(x_n|s'_{n,i})$ to assume that $X(s'_{n,i}) - x_n$ follows a normal distribution with zero mean (the variance is experimentally determined).

Similarly, $P(\boldsymbol{s}_1)$ and $P(\boldsymbol{s}_n|\boldsymbol{s}_{n-1})$ are expanded to:

$$\begin{aligned} P(\boldsymbol{s}_1) &= P(\boldsymbol{q}_1) \prod_{i=1}^{I} P(s'_{1,i}), \\ P(\boldsymbol{s}_n|\boldsymbol{s}_{n-1}) &= P(\boldsymbol{q}_n|\boldsymbol{q}_{n-1}) \prod_{i=1}^{I} P(s'_{n,i}|s'_{n-1,i}) \\ &= P(\boldsymbol{q}_n|\boldsymbol{q}_{n-1}) \prod_{i=1}^{I} P(s'_{n,i}). \end{aligned}$$

When we assume that $s'_{n,i}$ is selected at random with equal probabilities, $P(\boldsymbol{s}_1) \propto P(\boldsymbol{q}_1)$, $P(\boldsymbol{s}_n|\boldsymbol{s}_{n-1}) \propto P(\boldsymbol{q}_n|\boldsymbol{q}_{n-1})$.

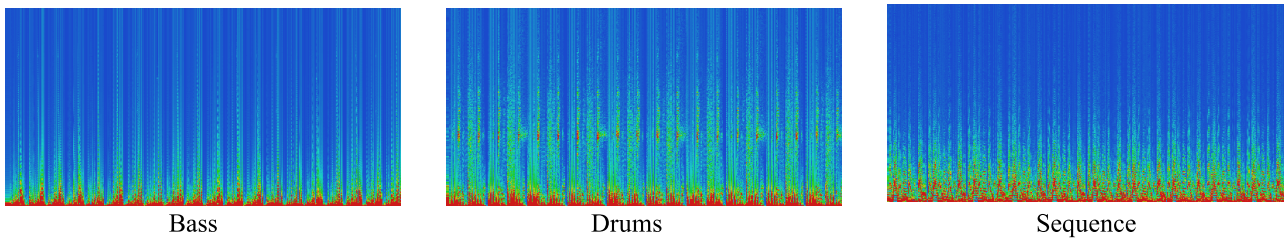
Therefore, the selection of music loops is achieved with the following equation:

$$\hat{\boldsymbol{Q}} = \underset{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_N}{\operatorname{argmax}} \prod_{n=1}^{N} P(x_n|\boldsymbol{q}_n) \cdot P(\boldsymbol{q}_1) \prod_{n=2}^{N} P(\boldsymbol{q}_n|\boldsymbol{q}_{n-1}),$$

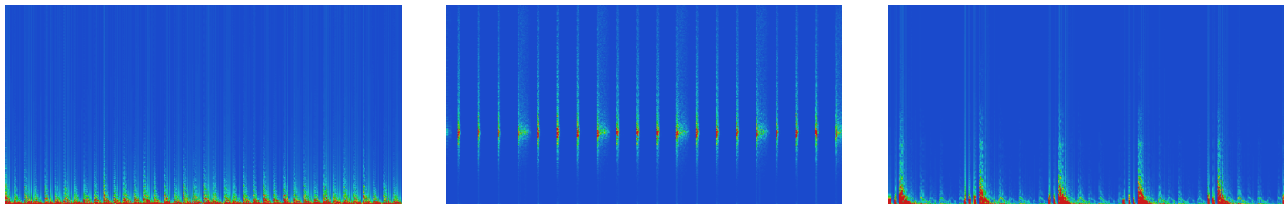
$$\hat{\boldsymbol{S}'} = \underset{\boldsymbol{s}'_1, \cdots, \boldsymbol{s}'_N}{\operatorname{argmax}} \prod_{n=1}^{N} \prod_{i=1}^{I} P(x_n|s'_{n,i}).$$

3.3 Estimation of the degree of excitement for music loops

When a music loop contains sound at a wide range of frequencies (from low-frequency regions to high-frequency regions) from the beginning to the end, this music loop is



(a) High degree of excitement (4 or 5)



(b) Low degree of excitement (1 or 2)

Figure 2. Examples of music loops with high and low degrees of excitement

considered to have a high degree of excitement (Figure 2). We therefore propose the following method for determining the degree of excitement. For a given music loop, a spectrogram is calculated using the short-term Fourier transform with a 4096-point Hamming window shifted by 10ms. Let $A_{t,f}(s)$ be the amplitude at the time t and the frequency f of the music loop s. The function $\sigma(A_{t,f}(s))$, ranging from 0.0 to 1.0, is calculated from $A_{t,f}(s)$ at every time and every frequency. The function $\sigma(A_{t,f}(s))$ has a value near 0.0 when $A_{t,f}(s)$ is near zero, whereas it has a value near 1.0 when $A_{t,f}(s)$ is higher than a certain value. In the current implementation, this function is calculated as follows:

$$\sigma(A_{t,f}(s)) = \begin{cases} 0.0 & (A_{t,f}(s) \leq 0.1) \\ 0.2 & (0.1 < A_{t,f}(s) \leq 0.2) \\ 0.4 & (0.2 < A_{t,f}(s) \leq 0.3) \\ 0.6 & (0.3 < A_{t,f}(s) \leq 0.4) \\ 0.8 & (0.4 < A_{t,f}(s) \leq 0.5) \\ 1.0 & (A_{t,f}(s) > 0.5) \end{cases}$$

This can be considered an approximation of the sigmoid function. After $\sigma(A_{t,f}(s))$ is calculated, its average is calculated along both the time and frequency axes:

$$R(s) = \frac{1}{TF} \sum_{t=1}^{T} \sum_{f=1}^{F} \sigma(A_{t,f}(s)).$$

It is then normalized by dividing it by the maximum value in all music loops of the same instrument part and is transformed to an integer value of 1 to d:

$$X(s) = \left\lceil d \frac{R(s)}{\max\limits_{s' \in \mathcal{M}_i} R(s')} \right\rceil,$$

where $\lceil \cdot \rceil$ is a ceiling function.

4. IMPLEMENTATION AND EXPERIMENTS

4.1 Implementation

We implemented this system using Processing 1.5.1. Music loops were taken from *Sound PooL* [4], which consists of music loops for each of the five instrumental parts of *Sequence*, *Synth*, *Bass*, *Percussion*, and *Drums*. Of these five parts, Percussion has an unique different feature in that; it mainly include sounds that provide an accent (e.g., a crash symbal). For this reason,we place a music loop of Percussion at the beginning of every four measures, independent of the HMM-based formulation described in Section 3. The HMM-based formulation is applied to the remaining four parts of Sequence, Synth, Bass, and Drums. The parameters $P(x_n|\boldsymbol{q}_n)$, $P(\boldsymbol{q}_1)$, $P(\boldsymbol{q}_n|\boldsymbol{q}_{n-1})$ were experimentally determined.

4.2 Experimental methods

We conducted two experiments, on which evaluated the excitement estimation method through paired comparisons (Experiment 1) and the system's usability (Experiment 2).

In Experiment 1, we asked participants to listen to pairs of music loops and to answer which loop in each pair had the higher excitement. We prepared 20 pairs (i.e., four pairs for each part) at random.

In Experiment 2, we asked participants to use our system to compose a piece of background music for a silent movie. Focusing on the techno's feeling of lively motion, we used promotion movies of sport cars [1]. These movies include both slow parts and fast parts that express the evolution of excitement. The time for the composition was limited to 30 minutes, and during the 30 minutes we allowed the participants to redraw the excitemment curve and to add, remove, and change music loops as needed. After they completed each composition, we asked them to rate their

[1] 1) Daihatsu, Tokyo Motor Show 2013 KOPEN future included, https://www.youtube.com/watch?v=zca6Lv4PW8M
2) Honda, NSX Concept GT Shakedown Test, https://www.youtube.com/watch?v=Bip5j00I6Hw

Table 1. Results of Experiment 2

	Baseline	Proposed
Q1	3.13	4.02
Q2	2.54	4.46
Q3	3.94	3.59
Q4	2.39	3.15
Q5	4.98	5.31

experience with the system that they used by answering the following questions on a scale of one to seven:

Q1 How easy or difficult was it to understand the system?

Q2 How easy or difficult did you feel it was to compose this piece of music without any prior musical knowledge?

Q3 How much or how little did the composed music have the degree of excitement that you intended?

Q4 Do you feel that you could compose satisfactory music?

Q5 How would you describe your level of interest in music composition?

The experiment was conducted using both the baseline system and our system. In the baseline system, the automatic music loop selection function was removed from our system. To reduce any effect of that the order of system use might have on study results, half of the participants first used our system, and the other half first used the baseline system. To reduce the effect of that participant fatigue might have on study results, we allowed the participants to have a sufficient rest between use of the two systems.

The participants were 10 university students. Of the 10 participants, seven had prior experience with playing an instrument, and two knew and/or had used a MIDI sequencer.

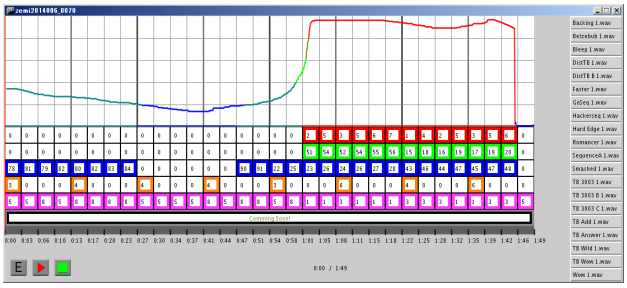
4.3 Experimental results

Experiment 1

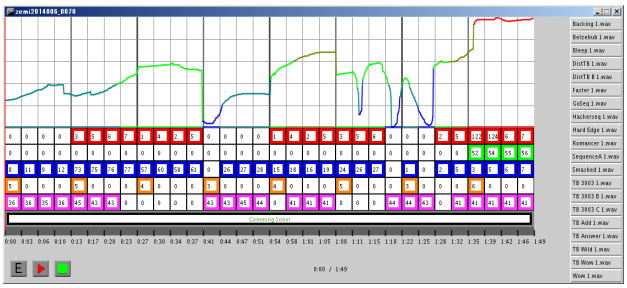
In 16.6 of the 20 pairs on average (approximately 83%), the system's estimation of the degree of excitement agreed with the participants' judgement. It was thus shown that our method for estimating the degree of excitement was appropriate.

Experiment 2

The results are listed in Table 1. For Questions 1 and 2, our system improved a mean score by 0.89 points and 1.92 points, respectively, as compared to the baseline system. This is because participants gave our system a high rating for ease of drawing a curve. On the other hand, for Question 3 our system scored 0.35 points lower than the baseline system. One participant pointed out that the timing of changes in the excitement sometimes did not match the drawn curve. Because our system places music loops measurewise, changes in the excitement occur only at changes in measures. This may explain why our system ranked slightly lower than the baseline system for Question 3. For Question 4, our system scored 0.76 points higher than the baseline system. For Question 5, our system scored 0.33 points higher than the baseline system.



(a) Participant 1



(b) Participant 10

Figure 3. Examples of the screenshots of the musical pieces composed by the participants, using our system

Figure 3 shows excerpts of the screenshots of the musical pieces that participants composed. It is apparent that participants tried to create an evolution of excitement according to the lively motion of the given movie, and that the entered excitement reflected in selection of music loops.

5. CONCLUSION

In this paper, we focused on a new musical feature, the *degree of excitement*, and proposed a loop sequencer that allows the user to directly input the degree of excitement as a curve on a computer screen. Once the degree of excitement is entered, the system determines what loops and how many loops should be placed to reflect the curve in the music. With this function, our system enabled users to quickly and easily compose techno music. Future issues that need to be addressed include reimplementing this system as a Web-based application and acquiring log data by a variety of users to improve the system's behavior.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 26240025.

6. REFERENCES

[1] 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, pp. 299–302.

[2] D. Ando, P. Dahlstedt, M. G. Nordaxhl, and H. Iba, "Computer aided composition by means of interactive GP," in *Proc. ICMC*, 2006, pp. 254–257.

[3] G. Nierhaus, *Algorithmic Composition*. Springer, 2009.

[4] AH-Software, "Sound PooL." [Online]. Available: http://www.ah-soft.com/soundpool/

Granular Model of Multidimensional Spatial Sonification

Muhammad Hafiz Wan Rosli[1], Andrés Cabrera[2], Matthew Wright[3], and Curtis Roads[4]

Media Arts and Technology,
University of California,
Santa Barbara CA 93106-6065 USA

Email: {[1]hafiz, [2]andres, [3]matt, [4]clang}@mat.ucsb.edu

ABSTRACT

Sonification is the use of sonic materials to represent information. The use of spatial sonification to represent spatial data, i.e., that which contains positional information, is inherent due to the nature of sound. However, perceptual issues such as the *Precedence Effect* and *Minimum Audible Angle* attenuate our ability to perceive directional stimuli. Furthermore, the mapping of multivariate datasets to synthesis engine parameters is non-trivial as a result of the vast information space. This paper presents a model for representing spatial datasets via spatial sonification through the use of granular synthesis.

1. INTRODUCTION

Sonification is the process of representing data through the sound domain. It has been proven that sonification is able to augment the visual display, and provide a platform for perceiving data for the visually impaired. Furthermore, a greater number of variables can be presented in one display, and some types of data are better suited for the sound domain, such as rapidly changing information.

However, the representation of data via visualization has been much more developed, as opposed to its auditory counterpart. In the case of spatial datasets (which contain location components along with other dependent variables), the mapping process from data to sound is inherently complex due to the dimensionality of the dataset.

In order to map the spatial data to spatial sound via the use of multiple loudspeakers (surround sound), perceptual attributes that pertain to the auditory system, such as the limitation of spatial acuity, has to be taken into consideration.

Through our research, we have developed a granular model for multimodal representation of spatial data via spatial sonification. Perceptual issues that pertain to spatial attributes (surround sound), and microsound are discussed, and taken into consideration in developing the system. Mapping strategies for granular synthesis (in the context of sonification) are also explored and described. Finally, we present how transformation of (micro) sounds, and interactivity could assist in the discovery of unknown patterns in a dataset.

Copyright: ©2015 Muhammad Hafiz Wan Rosli 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.

2. MOTIVATION

One family of sounds is known as *microsound* [1]– sound particles in the range of 1 ms to 100 ms, that span the boundary between what can be perceived as an individual entity, and what has no distinct perceptual characteristics.

Motivated by the interest of this specific family of sounds, we started to search for natural occurrences that fall into the same category. One particular phenomena that interested us was the sound of thunder, as this is, in fact the result of a burst of electrical energy lasting for about approximately 20 µs. As the mass of energy is introduced, part of the energy is transferred into light (lightning) and sound (thunder).

On April 30th 2014, the *High Definition Earth Viewing* experiment was activated aboard the *International Space Station*, which presents viewers with images of Earth seen from outer space. One of the elements that was clearly shown was lightning occurrences. Guided by the interest in this particular phenomenon, we started to inquire if there were any correlations between lightning strikes in different geographic locations. This curiosity lead us to examine NASA's lightning dataset, and develop a research based on multimodal data representation, which resulted in a perceptual model of sonification– shown in the artwork *Point cloud* [2].

Sonification as a means to understand and display datasets, as well as the use in aesthetic explorations has been explored by various disciplines [3–8].

3. DATA SOURCE

The data for *Point Cloud* were gathered from the NASA Marshall Space Flight Center (MSFC) [9]. The data were generated by two of their space-borne optical sensors: the Optical Transient Detector (OTD) and the Lightning Imaging Sensor (LIS). The actual data span more than 16 years of lightning information, but for the purpose of this project, only a single year's worth of data was used.

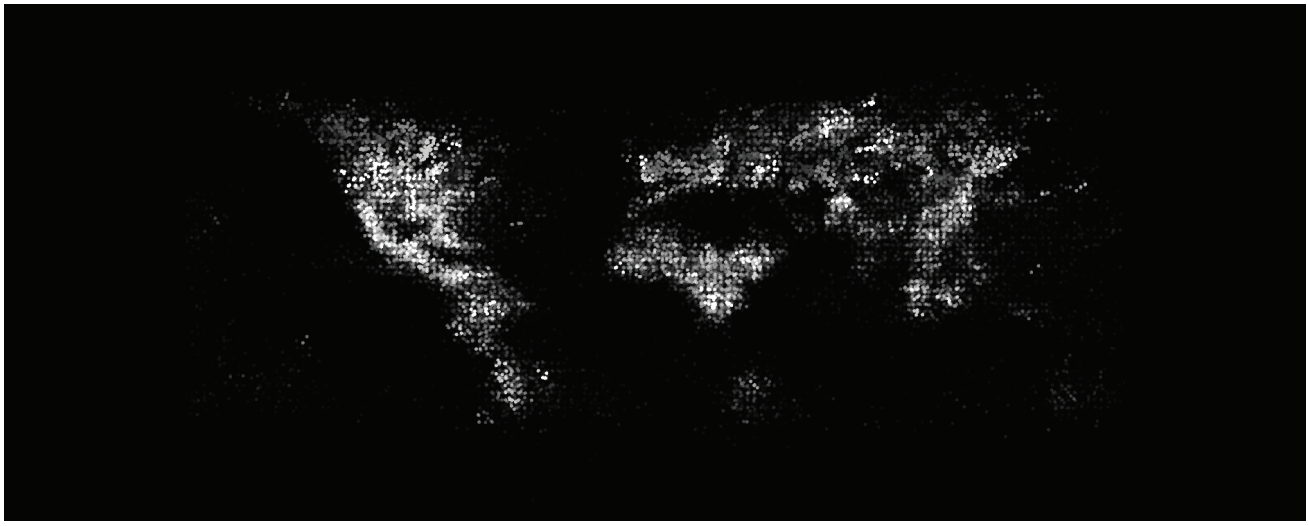


Figure 1. Visualization of *Point Cloud*: A map of the Earth showing lightning occurrences over the course of a single year (day 333).

The data take the form of an annual cycle of flash rates with a product dimension of 720 x 360 x 365 (bin size of 0.5 degrees $\times$ 0.5 degrees $\times$ one day) (Fig. 2). Each *data slice* (single day) contains a *flash rate* (number of occurrences per day) for each spatial *data point* (720 x 360).

The MSFC dataset is distributed in Hierarchical Data Format [10] and was parsed using the Python PyHDF library.

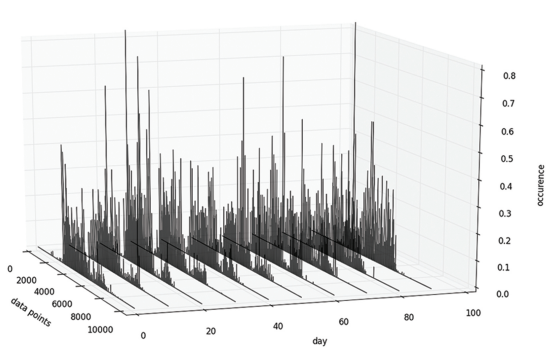


Figure 2. Lightning occurences for flattened [720 x 360] data points over series of days

4. SONIFICATION

Data sonification in auditory displays can provide information to complete, augment, or replace visual displays. As with most systems that present data, sonification techniques aim to provide the means for extracting information from a dataset to be parsed by the perceiver [3]. Additionally, sonification allows us to potentially extract new patterns and relationships, which are not necessarily perceived by simply analyzing the dataset.

The domain of data visualization faces similar issues, as it deals not only with the dataset being represented, but has to also consider the user as part of the process in parsing information.

4.1 Multimodality and Sonification

We typically take for granted how our senses parse information in our daily lives. When we interact with the real-world, we almost always receive feedback through various modalities, allowing us to understand our surrounding, and successfully navigate through the environment. Our perceptual system functions by correlating the information from these various senses to construct a mental image of our surrounding.

In order to design an effective sonification system, these well developed mechanisms of perception need to be involved. Auditory and visual stimuli needs to be coupled by the same mechanism which couples perceptual units in the real world to create a cohesive *environment* [3]. There have been efforts to simultaneously present data in both the auditory and the visual domain, in order to give the perceiver a more concise understanding of the dataset [3]. Such an example can be seen in *Point Cloud* (Fig. 1) [2].

4.2 Perceptual Issues

Our perception tends to fail at grasping the bigger picture when presented with a single viewpoint of information. This phenomena is exemplified, for instance, by the need to interactively change perspectives while viewing a 3-dimensional structure in the real world. When doing so, we allow ourselves to acquire different views, which then provide better sense of the object.

Sonification suffers from an analogous problem. When a complex data space is projected onto a linear audio signal, we are unable to gain different sonic views of the dataset, rendering the system less effective. One solution is to change the mapping parameters so that the perceiver

is able to acquire a variety of sonic perspectives [3], discussed in Section 5.2. Additionally, the use of sound spatialization enables us to address this problem [11].

Sound spatialization is used in our system to assist a perceiver in gaining multiple perspectives of the dataset. In doing so, the complex data space is not collapsed into a single audio signal originating from one direction.

4.3 Spatial Sound

> "A cascading sequence of sound objects, each emanating from a different virtual space, provides the dimension of spatial depth to an otherwise flat perspective and articulates a varying topography" [12].

The use of multiple loudspeakers for spatial sound reproduction allows stimuli to be presented from different locations, preventing the complex data space from collapsing into a single audio signal originating from one direction. This, in turn allows a perceiver to interactively navigate around the acoustic environment in order to acquire different sonic views. Similarly, this mode of interaction is how humans localize acoustic energy in the physical world [8, 13]. Nasir and Roberts [11] explains that "location information can be used to enhance the sonification, or can be used to represent qualitative information."

Similarly, representation of spatial data via the means of visualization has had its share of exposure, dating back to the thorough dissection by semiologist Jacques Bertin [11, 14].

As we are dealing with a dataset that presents spatial information within specifically localized regions, it is only natural to include spatialization as a key aspect of the representational system. By correlating each spherical coordinate of the earth to the spatial position in the rendered (sound) field (longitude and latitude mapped to azimuth and elevation), we enable the auditory stimulus to be localized at its respective position. In other words, a perceiver would be "looking" at the dataset from a viewpoint inside the Earth.

One of the main objectives of the underlying research is to represent a single spatial dataset via multimodal stimuli. However, spatial acuity is much finer for vision than it is for hearing [3]. In order to allow users to distinguish stimuli coming from separate distinct locations, some consideration needs to be addressed.

4.4 Localization of sound

In order to effectively convey the perception of space, we have explored methods of sound localization, specifically those that pertain to sonification, as discussed in [11]. These methods are also thoroughly examined in various texts concerning sound spatialization, such as [3, 8, 15].

Non- spatial audible variables: These are the building blocks of sonification, which typically includes synthesis parameters such as pitch, loudness, and tempo. As discussed in Section 5.2, we have mapped the flash rate value of every data point to its corresponding granular stream's grain density, and grain amplitude.

Non- spatial motifs: These higher order components are intended to provide a better system for the perceiver to understand patterns in the dataset. Description of our implementation can be found in Section 6. Although these specific structures typically needs to be learned, we believe that the human brain is able to adapt, and find patterns in these higher-level dimensions– if there are patterns to be perceived.

IID & ITD: The Duplex Theory [16] states that we perceive directionality (and auditory space in general), through the use of *Interaural Time Difference* and *Interaural Intensity Difference*. The use of multiple loudspeakers allow us to successfully use this mechanism in placing a localized stimuli in a radial space. Due to the fact that this mechanism is a well developed component of our perception, the spatial data could be perceived without further training.

Time-based effects: Temporal factors provide excellent cues for sonic data exploration. The ability to traverse different time scales provide the means to understand various hidden structures in a dataset. We have explored temporal transformations to analyze *Microstructures* and *Macrostructures* in the dataset (Section 6.3).

5. SYNTHESIS ENGINE

Some of the various synthesis techniques used for sonification are more suitable than others, depending on the data that is analyzed (for example, in the case of multidimensional datasets). Undoubtedly, most of the "effective" sonification systems consider the dataset, and implements techniques that would best fit the data.

Our system is implemented using *Parameter Mapping Sonification* due to its effectiveness in displaying multivariate data [3]. This technique involves the mapping of data features onto parameters of sonic events, such as pitch, level, and onset time. Our model implements granular synthesis as the synthesis engine in order to render short, discrete events in the dataset (lightning occurrence). Furthermore, these short bursts of energy resembles the sound of thunder, which in turn enhances the effect of *Gestalt Principle of Past Experience*.

As the visualization algorithm displays a unit of occurrence for a brief period of time, the sonification engine renders a short burst of sonic energy. This allows the auditory and visual stream to be coupled together, as discussed in 4.1. We respect natural physical coherences by binding the visual and auditory events together temporally, giving the impression of causality [3, 14].

5.1 Granular Reverberation

Thunder is the result of a shock wave caused by a sudden thermal expansion as lightning passes through the air. A typical lightning bolt lasts for about approximately 20 µs. As the mass of energy is introduced into the cumulus cloud enclosure, its impulse response takes the shape of irregu-

larly spaced delays as a result of spectral reflections in the cloud formation [17].

This effect can be synthesized using a technique known as granular reverberation. The foundation for granular reverberation is *Asynchronous Granular Synthesis*, which scatters grains statistically within a region defined on the time/ frequency plane (Fig. 3) [18]. The number of grains in a particular region determines the density of sound particles (for each granular cloud).

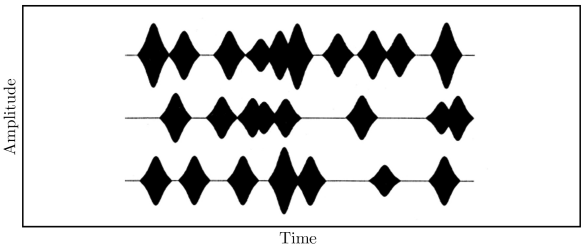


Figure 3. Varying density of 3 granular streams mapped to flash rate [19]

5.2 Mapping Strategies

The data of lightning occurrences is presented as a time series corresponding to the days in a single year. Every *data point* (Fig. 4) in the dataset holds a value corresponding to the amount of lightning (flash rate) for a particular geographic coordinate (longitude, latitude) of a given day. To simulate the individual lightning strikes (while retaining the ratio between each data point), we have chosen to map the flash rate values to the *density* of grains (discussed in Section 5.2.1).

Although this is not the "actual individual lightning occurrence," the triggering rate of the grains somewhat gives us a cue of how "dense" the occurrences are around a particular part of the world. The *maximum amplitude* and the *duration* of the grains in a particular stream are also correlated to the values of each data point, as a means to intensify the data mapping. Other synthesis parameters such as *grain triggering rate* and *grain length random deviation* are mapped to stochastic processes.

The use of synthetic grains with sharp *attack* and *decay*, and a lifespan between 10 ms to 50 ms enables us to evoke the idea of individual bursts of lightning bolts. The short barrage of energy also results in a very strong association with the rendered points in the visualization. This, in turn, results in the tendency for these two different stimuli to be grouped together as an interconnected event.

The nature of this technique allows for a multitude of low-level parameter manipulations. In contrast, the usage of granular synthesis in creative applications (as opposed to sonification practices) often requires thousands of control parameters per second, resulting in the need of higher-dimensional control parameters.

> "Granular synthesis requires a massive amount of control data. If *n* is the number of parameters per grain, and *d* is the density of grains per second, it takes *n* times *d* parameter values to specify one second of sound" [1].

On the other hand, multivariate datasets provide us with a wide range of parameters that could be assigned to granular synthesis' control data. The issue lies in fine tuning the synthesis engine to fit the dataset, and finding the best ways to *parameterize*, so as to allow changes in the data to be perceived by the user. This is where aesthetic features of the sonification plays an important role– to allow the dataset to be cognized by the user.

5.2.1 Parameterization

For every flash rate value in the dataset, we create a Gaussian function (1) centered on the *data point*. The normalized flash rate is then set to the height of the curve's peak (Fig. 5). In effect, sonic grains are statistically rendered around the data points, based on their actual location in the dataset (Fig. 6). Consequently, the density of grains in a particular area now gives a perceptual description of occurrences in that region.

$$f(x) = a \exp(-\frac{(x-b)^2}{2c^2}) \quad (1)$$

where a, b and c are real constants

Additionally, this allows the algorithm to retain each data point's *relative weight* compared to other points on the same data slice, independent of the temporal scaling (discussed in Section 6.3). One could implement an algorithm that is set to render a non-statistical element at each data point, but the result would be a repeating cycle, akin to looping an audio file. Instead, the synthesis engine renders a sequence of grains for each data point on the grid. As such, the density of grains at a particular location in time retains its overall weight every cycle. However, it does not appear to be an exact repetition of the previous cycle as a result of statistically generating new grains every time the data point is updated.

5.3 Grain Density

The attempt to provide perceptually distinct cues for individual lightning occurrence also causes the data to be obscured. The number of grains (per data slice) at an instance becomes far too dense for the differences to be perceived [1] . We discuss the technique of focusing on individual streams in Section 6.2.

Our auditory system perceives events happening with intervals less than 20 Hz as distinct events. However, as these events are sped up to more than 20 Hz, they are perceived as a continuous stream. The visual equivalent of this can be seen in the phenomenon called *Persistence of Vision*. At 30 frames per second, our brain processes these visual stimuli as a continuous event.

In the case of the represented dataset, the speed at which the grains are generated would correspond to grain density per data point. When we take the whole data slice into account, we get a number of grains that is generated at a rate that temporally smears the grains into a continuous tone.

[1] https://soundcloud.com/muhammad-hafiz-wan-rosli/graindensity

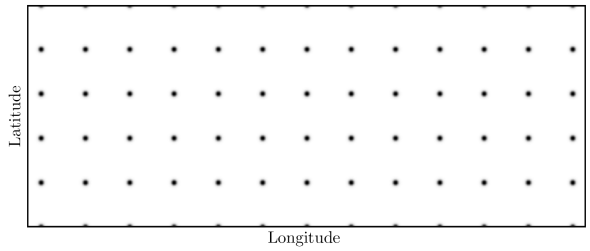


Figure 4. An array of data points for a single day

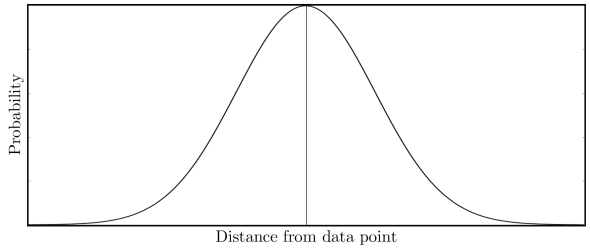


Figure 5. Probability curve of one data point in Figure 4

Suppose we have an average grain density of 100 grains per second for a single data point. The number of grains in one second of time would be:

[longitude resolution ∗ latitude resolution ∗ grain density]

$$720 \times 360 \times 100 = 25,920,000 \text{ grains per second} \quad (2)$$

Therefore, the problem of data representation through the means of granular synthesis is somewhat reduced to a perceptual and psychoacoustical problem. How do we pose a potential solution to parse this dense information space acoustically?

6. GRANULAR TRANSFORMATION

> "Xenakis observed how sound particles could be viewed as short vectors within a three dimensional space bounded by frequency, amplitude and time" [1].

6.1 Frequency Transformation

As is well known, the Cocktail Party Effect illustrates that our brain is capable of focusing auditory attention on a particular stimulus while filtering out a range of stimuli [8, 20] [2] . However, this effect is influenced not only by our ability to segregate sounds based on their spectral and temporal qualities, but also by the spatial relationships between the sounds.

Consider, for example, the ability to segregate multiple instruments in a recording, and to focus on a specific instrument. We are able to do so because we associate each unique instrument with a specific timbre (and melodic motives). Furthermore, our ability to identify unique instruments is also affected by the direction of which the sounds originate from [8, 13]. These cues help us localize sound sources, which ultimately contributes to recognizing a stimulus as a continuous pattern.

[2] http://sonification.de/handbook/media/chapter3/SHB-S3.1b.mp3 [20]

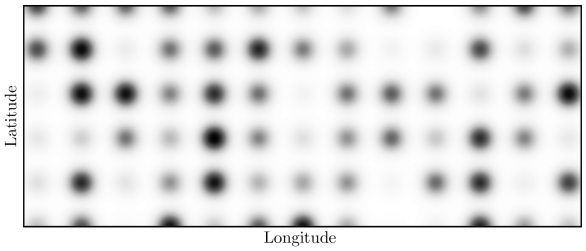


Figure 6. Grain distribution for data points in Figure 4

6.1.1 Unique bands per quantized longitude

> "The ability to selectively attend to simultaneously sounding *auditory objects* is an ability that is not yet completely understood. Nonetheless it provides fertile ground for use by designers of auditory displays" [3].

Bandlimiting a set of grains allow us to theoretically differentiate between separate groups of stimuli, i.e "granular streams". As discussed in section 5, the synthesis engine exploits our perceptual ability by bridging the connection between what is seen and what is heard. However, the downside of using granular means for synthesizing the stimuli disables us from clearly identifying the separate grains, causing the mass of sounds to be perceived as a single evolving event. Although this effect is useful in parsing the overall macro pattern, the microstructure tends to lose its meaning through the dense cloud of sound particles.

By mapping the differences in azimuth (of the dataset) not only to its corresponding spatial position (in the rendered field), but also to a specific frequency band (Fig. 7), we allow the data to be segregated based on its spectral content and spatial location. However, the generation of synthetic (sinusoidal) grains is far too similar to one another, even with the assistance of spatial relationships.

This effect is further diminished due to the Minimum Audible Angle, which is defined as the *Just Noticable Difference* (in azimuth) for listeners [21]. One solution might be to fine tune the quantization of longitudinal space to fit the space where the model will be rendered in. The number of speakers used affects the ability to render directional stimuli, which, in turn, helps in segregating directional sources. We intend to further explore this possibility in the near future via the use of the *Allosphere* [22].

6.1.2 Granular clouds

The grouping of elements is further explored by segregating groups of grains to form what is known as granular clouds. If a set of grains are bounded by a pre-determined set of rules and parameters, then they would appear to morph "in unison". We implemented this technique for the different continents, which allowed us to analyze the trend of change per continent, and how one continent's flash rates relate to another's. In doing so, we now reduce

the amount of concurrent events to "concentrate" on, enabling us to analyze the macroscale patterns. Here lies another example of how mapping a dataset to a higher-level representation could give rise to new meanings, and allow us to find patterns that were otherwise difficult to perceive (or even non-existent).

6.2 Amplitude Transformation

As discussed in [2], our visual system is able to focus on a group of stimulus in a specific position, while disregarding the other stimuli– akin to looking through a magnifying glass. Although our auditory system is able to segregate, and focus on different stimulus [20], the dense spatial dataset prevents a perceiver to tune in to specific areas of interest. To achieve a similar result (as the visual senses), we have implemented a means to "blur out" or "smear" the dataset– except for the area being viewed. This notion of *Interactive Data Selection* has been implemented, and discussed in the context of Parameter Mapping Sonification [3, 23, 24].

6.2.1 Acoustic focus

To focus on a specific area of the dataset, we *pass* the sonic grains through a conditional construct that checks if the generated grains are within a specific boundary. If these grains are within the boundary, then they are rendered. Otherwise, the grains are not rendered.

If we were to perform the conditional statement on a data point, instead of the rendered grains, we would not be able to render areas in between the data points. Instead, we are now able to seamlessly move the focal point around the dataset, while rendering grains that are only within the boundary.

Implementing this type of control not only enables us to focus on a specific data point, but also allows us to control the width of the scope, i.e the number of data points to be included in the focused region. Another parameter that is now at our disposal is the ability to control the loudness roll-off of the regions around the focused area.

6.2.2 Multiple focus

The number of focused regions could also be controlled (Fig. 8), so as to allow the perceiver to, for example, compare the data of several areas of interest. This control scheme reinforces the effect of temporal, and frequency transformations.

6.3 Temporal Transformation

Time domain transformation is well known in the realm of electronic music, discussed in depth by composers and musicians alike, including Stockhausen in his 1972 lecture entitled *Four Criteria of Electronic Music*. Speeding up a sequence of rhythmic events causes a transformation from distinct individual events perceived as rhythmic, into a continuous tone. Further increment of the speed creates an increase in pitch, whilst a decrease in speed results in a lowering of the pitch.

Temporal transformations allow us to traverse between time scales to perceive different relationships in the dataset's temporal structure. In the case of our implementation, it allows us to perceive differences in *Microstructure* and *Macrostructure*.

6.3.1 Microstructure

The analysis of fluctuations in lightning occurrences for a particular location might not be a trivial task, as there are a multitude of concurrent granular streams. Coupled with the ability to segregate granular streams via amplitude transformations (Section 6.2), the relationship of one particular data point through time would be easier perceived if the temporal domain is stretched.

6.3.2 Macrostructure

On the other hand, if we were to compress the temporal domain, the distinct granular streams would be transformed into continuous tones [3] . A crucial point to note is that these manipulations do not effect the ratio between data point values (in a data slice). Therefore, the individual flash rates per time frame retains the same weight throughout the temporal transformation. What seemed to be a mass of micro-events resembling noise is now transformed into (720 x 360) pitched tones.

As a result, we can now compare data points over time by listening to the differences in pitches: The higher the pitch of a particular data point, the higher the lightning occurrence. Additionally, we can now analyze the data to extract higher level information, such as the ratios between tones (how the flash rate of a particular point relates to the flash rate of another point), the amount of frequency shift (glissando) corresponding to the changes in flash rates of a particular location (data point), and the rate of frequency shift in relation to another data point (rate of change for flash rates).

[3] https://soundcloud.com/muhammad-hafiz-wan-rosli/graintemporaltransformation

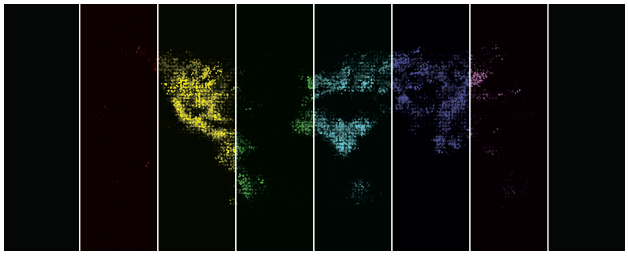


Figure 7. Bandlimiting grains per quantized longitude

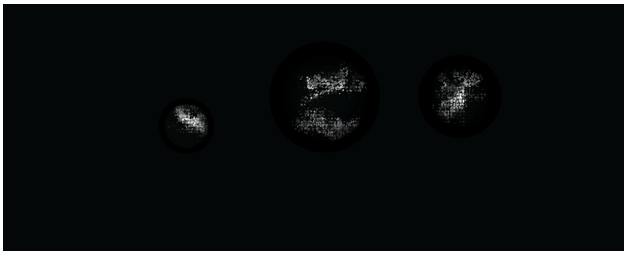


Figure 8. Multiple focus and loudness roll-off

If a granular stream does not change through time, then the number of occurrence for that data point does not fluctuate. In effect, we can hear the fluctuating changes of a particular location by listening to the granular streams that change from tone to rhythm, and vice versa.

7. INTERFACE FOR EXPLORATION

The mapping of spatial data to spatial sound was indeed one of the crucial components of this research. However, as the perceiver is an important component to the sonification, interface, naturally becomes an important factor as well.

We are currently in the process of designing custom hardware that would enable us to address spherical coordinates via a multi-touch spherical interface. However, we have explored (off-the-shelf) interfaces to navigate the system, which has produced satisfactory results.

Frequency and Amplitude transformations, as well as zooming capabilities are executed via the use of trackpad or *Wacom* tablet [25]. The *Griffin Powermate* [26] was used to achieve temporal transformations, whereas a *Graphical User Interface* was used for parameter control.

As a compositional aesthetic, we have also included a mode where the rate of reading the data slice is increased after every yearly iteration. In the "final" iteration, the whole year's worth of data would be presented as a single impulse, which contains the "energy contained in one year's worth of lightning/ thunder".

8. TOOLS

The initial version of the system was realized using a heterogeneous setup to allow fast prototyping. The data parsing, handling and processing was done using python, in particular the interactive ipython notebook. The sonification was done using Csound [27] within the ipython notebook, and the visuals were done using processing [28]. The synchronization and data interchange between applications was done using Open Sound Control [29].

9. CONCLUSION AND FUTURE WORK

We have explored, and described a granular model for multimodal representation of spatial data via spatial sonification. We have also discussed the perceptual issues that are taken into consideration, and propose solutions to overcome them. Interactivity, mapping strategies and transformations related to granular synthesis have also been examined to provide a platform for pattern discovery. These techniques can be used as an auditory display to complement a visualization component, as well as an interactive tool for sonic data exploration.

We are currently in the process of porting the system from its *surround-sound* version to a large immersive 3D space, the *Allosphere*. This would allow us to render sounds in a *periphonic* (full 360°) environment. Explorations on spatial interfaces will also be carried out as we believe interactivity is a major component to sonification. The Allosphere is a 3-story facility that contains a 10 meter diameter sphere that provides 360° realtime stereographic visualization using a cluster of servers driving 26 high-resolution active-stereo projectors. Audio is projected by 54 loudspeakers positioned along three rings of the sphere [22, 30].

Another approach to the granular paradigm is through the use of granulation, which divides a sample into short enveloped grains, and reproduces them in high densities (as opposed to generating *synthetic* grains via granular synthesis). Granulation of samples possesses a unique, *organic* aesthetic quality which could assist in unraveling the "poetics" of the dataset, which in turn could allow users to be more perceptually engaged. Unique spectral transformations could also be applied to selected areas in order to assist the user in data exploration– examples of these transformations range from user defined systems to algorithmic processes, such as *Dictionary- Based Methods* [12].

9.1 Perceptual validation

We plan to conduct several user studies to analyze the effectiveness of our system. The following are potential scenarios of a user-study.

The users are exposed to multiple 10 second segments of the sonified dataset, specifically those which contain correlation in the change of lightning occurrences over time between two data points. The excerpts would contain a combination of various segments (both sparse and dense) to be analyzed by the user. In each segment, the user is presented with the dataset, sonified using well-known spatial sonification techniques [11], followed by our system.

For every segment, the user is asked:

- **If there were any correlations in the data**. The user would be asked to determine the number of perceived distinct stimuli. They are are also asked to determine which data point contains more occurrences using the frequency and/ or density difference (Section 6.3).
- **To point towards the source of the incoming stimuli**. Users are allowed to navigate (Section 7) via discussed transformations (Section 6). The accuracy is measured based on radial distance from actual stimuli.
- **Which type of sonification is preferred**. This qualitative selection is compared to the quantitative result of the tests, and the correlation between aesthetics, and accuracy (function) is measured to determine their interdependence.

The result of this user-study would allow us to quantitatively measure our system, and show if multimodality in spatial data representation assists the accuracy of data perception. Furthermore, it would also show if the aesthetics of a system play an important role in the perception of data (in a data representational system). Additional tests would include different datasets, and different synthesis techniques.

10. ACKNOWLEDGMENTS

We would like to thank CREATE and the Allosphere Research Group for the use of facilities. This work was supported in part by the Graduate Program in Media, Arts and Technology at the University of California, Santa Barbara.

11. REFERENCES

[1] C. Roads, *Microsound*. MIT Press, 2004.

[2] M. H. W. Rosli and A. Cabrera, "Application of gestalt principles to multimodal data representation." IEEE VIS, 2014.

[3] T. Hermann, A. Hunt, and J. G. Neuhoff, Eds., *The Sonification Handbook*. Berlin, Germany: Logos Publishing House, 2011.

[4] J. A. Dribus, "The other ear: A musical sonification of eeg data." International Conference on Auditory Display, 2004.

[5] H. van Raaij, "Listening to the mind listening - a sonification/composition." International Conference on Auditory Display, 2004.

[6] E. Childs, "Icad 2006: Global music 8212;the world by ear," *Computer Music Journal*, vol. 31, no. 1, pp. 95–96, March 2007.

[7] M. H. W. Rosli and A. Cabrera, "Gestalt principles in multimodal data representation," *Computer Graphics and Applications, IEEE*, vol. 35, no. 2, pp. 80–87, Mar 2015.

[8] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1994.

[9] "Nasa marshall space flight center," http://www.nasa.gov/centers/marshall/, accessed: 2014-06-28.

[10] "The HDF group - information, support, and software," http://www.hdfgroup.org/, accessed: 2014-06-28.

[11] T. Nasir and J. C. Roberts, "Sonification of Spatial Data." International Conference on Auditory Display, June 2007, pp. 182–196.

[12] A. McLeran, C. Roads, B. L. Sturm, and J. J. Shynk, "Granular sound spatialization using dictionary-based methods," in *Proceedings of the 5th Sound and Music Computing Conference*, Berlin, Germany, 2008.

[13] L. Rayleigh, *The Theory of Sound*. Dover Publications, 1945.

[14] M. Chion, C. Gorbman, and W. Murch, *Audio-vision: Sound on Screen*, ser. Film and Culture. Columbia University Press, 1994.

[15] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*. MIT Press, 1997.

[16] L. Rayleigh, "On our perception of the direction of a source of sound," *Proceedings of the Musical Association*, vol. 2, pp. pp. 75–84, 1875.

[17] M. Uman, *Lightning*, ser. Dover Books on Physics. Dover Publications, 1969.

[18] C. Roads, *The Computer Music Tutorial*. MIT Press, Jan. 1996.

[19] ——, "Representations of musical signals," G. De Poli, A. Piccialli, and C. Roads, Eds. Cambridge, MA, USA: MIT Press, 1991, ch. Asynchronous Granular Synthesis, pp. 143–186.

[20] A. W. Bronkhorst, "The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions," *Acta Acustica united with Acustica*, vol. 86, no. 1, pp. 117–128, January 2000.

[21] A. W. Mills, "On the minimum audible angle," *The Journal of the Acoustical Society of America*, vol. 30, no. 4, pp. 237–246, 1958.

[22] X. Amatriain, J. Kuchera-Morin, T. Hollerer, and S. T. Pope, "The allosphere: Immersive multimedia for scientific discovery and artistic exploration," *IEEE Multimedia*, vol. 16, no. 2, pp. 64–75, 2009.

[23] D. Ó. Maidín and M. Fernström, "The best of two worlds: Retrieving and browsing," in *In Conference of Digital Audio Effects*, 2000.

[24] T. Hermann, T. Henning, and H. Ritter, "Gesture desk, an integrated multi-modal gestural workplace for sonification." Springer Berlin Heidelberg, 2004, vol. 2915, pp. 369–379.

[25] "Wacom pen tablet," http://www.wacom.com/en-us/products/pen-tablets, accessed: 2015-03-03.

[26] "Griffin powermate," https://store.griffintechnology.com/powermate, accessed: 2015-03-03.

[27] R. C. Boulanger, *The Csound book: perspectives in software synthesis, sound design, signal processing, and programming*. MIT press, 2000.

[28] "Processing.org," https://www.processing.org/, accessed: 2014-06-28.

[29] M. Wright and A. Freed, "Open sound control: A new protocol for communicating with sound synthesizers." Thessaloniki, Hellas: International Computer Music Association, 1997.

[30] J. Kuchera-Morin, M. Wright *et al.*, "Immersive full-surround multi-user system design," *Computers & Graphics*, vol. 40, pp. 10–21, May 2014.

On the Musical Opportunities of Cylindrical Hexagonal Lattices: Mapping Flat Isomorphisms Onto Nanotube Structures

Hanlin Hu, Brett Park and David Gerhard
Department of Computer Science, University of Regina
hu263@cs.uregina.ca | brett@shiverware.com | gerhard@cs.uregina.ca

ABSTRACT

It is possible to position equal-tempered discrete notes on a flat hexagonal grid in such a way as to allow musical constructs (chords, intervals, melodies, etc.) to take on the same shape regardless of the tonic. This is known as a musical isomorphism, and it has been shown to have advantages in composition, performance, and learning. Considering the utility and interest of such layouts, an extension into 3D interactions was sought, focussing on cylindrical hexagonal lattices which have been extensively studied in the context of carbon nanotubes. In this paper, we explore the notation of this class of cylindrical hexagonal lattices and develop a process for mapping a flat hexagonal isomorphism onto such a lattice. This mapping references and draws upon previous explorations of the helical and cyclical nature of western musical harmony, but is not limited to 12-tone equal tempered scales.

1. INTRODUCTION

The tiling problem in music theory describes the challenge of using periodic or aperiodic congruent tiles to partition a plane into a representation of musical significance. One solution is to tile triads into vertices in an equilateral triangle lattice. Based on the dual map of the equilateral triangle lattices, congruent hexagonal lattices are introduced to present isomorphic layouts, which have the following characteristics of musical keyboard design [1]: Transposition Invariance, where each construct such as interval, chord and scale have the same geometric shape regardless of the root key; and Tuning Invariance, where all constructs must have the same geometric shape in all tunings of the continuum (which allows for an extension of this theory from common 12-tone equal tempered usage into microtonal tunings).

Since the tonnetz was first introduced by Euler in the 1700s, many physical instruments have been developed which use isomorphic layouts on flat hexagonal lattices, including the AXiS Keyboard, the Hex player and others. Most of the keyboards are not reconfigurable, only providing a single layout, as is the case with the Opal, the Thummer and the like. Keyboards like AXiS-49, AXiS-64 and Rainboard can be reconfigured to alternate isomorphisms, but they present only a subset of the layout in a fixed window, rather than the standard 88 keys / eight octaves, or more. The reason full sized isomorphic keyboards have not been developed is that the boundary shape of keys in one octave are not uniform. As discussed in Section 2, different isomorphic layouts have different base intervals, and therefore when the layouts are extended to include 8 octaves (or whatever constraint may be applied), the overall shape and structure will be different. In order to construct an instrument that has access to 8 octaves, in a reconfigurable arrangement, without shifting positions of notes, a very large number of controller buttons would be needed and the object itself would be unnecessarily expensive and unwieldy.

Taking into account the cyclic, helical, repetitive nature of musical harmony, especially as it appears on hexagonal musical isomorphisms, it is possible to represent all notes in a much tighter arrangement, by curling a flat hexagonal isomorphism through the third dimension and aligning repeating octaves along the circumference of the resulting cylinder. Drawing on mathematics already competed in the study of buckminsterfullerene (Carbon Nanotubes), this paper describes the mathematical theory behind the appropriate amounts of curl for different isomorphic layouts, and presents a framework for constructing any such cylindrical hex isomorphic layout.

The paper is organized as follows: First, we explore the current state of isomorphic keyboard layouts and present some historical examples. Next, we explore the mathematics of carbon nanotubes. Third, we explore the orientation of an isomorphism to a nanotube using pitch axes and chiral angles. We then present details of the various cases that arise with specific arrangements, classify those cases, and present solutions for each. Finally, we show some examples of nanotube isomorph curlings, and discuss some possible directions for instrument design and playablity.

2. ISOMORPHIC MUSICAL LAYOUTS

Isomorphic layouts are the product of research on the tiling problem, as well as geometries of musical theory. Euler [2] was the first to introduce such an arrangement, based on whole-number ratios of related frequencies mapped into a lattice, shown in Fig. 1. Later, Riemann presented a similar lattice [3], by using triangles to represent major and minor thirds, shown in Fig. 2. The dual of this triangular tessellation of vertices is a hexagonal grid.

Copyright: ©2015 Hanlin Hu 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.

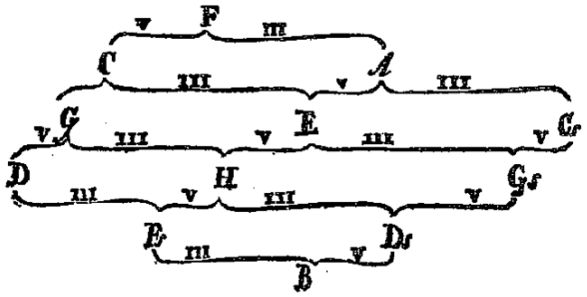


Figure 1: Euler's tonnetz.

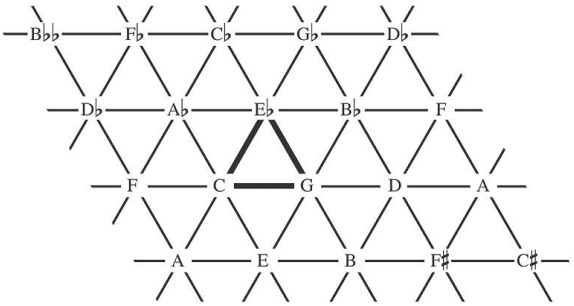


Figure 2: Riemann's triangular lattice.

Paul von Jankó designed a piano with horizontally whole tone and vertically semitone steps in 1882 [4], the arrangement of which is shown in Fig. 3. However, this piano did not achieve wide popularity because of the expense and weight of the instrument itself. The Wicki-Hayden layout was introduced as distinguishing seven white keys into two groups [5]. It benefits performance with shorten distance between keys in two groups, reducing learning time by unifying fingering into one pattern, and removing ambiguities by separating black keys into flat and sharp groups respectively. However, the keys in the Wicki-Hayden layout were not in a chromatic order, making it more difficult to learn for musicians used to adjacent semitones. Other popular isomorphic layouts such as Bajan, B-system, C-system, Gerhard and Park layouts are described in [6]. The AXiS keyboard, Opal, Thummer and Rainboard are physical constructions of these isomorphic layouts.

All of the isomorphic layouts mentioned above are based on flat, two-dimensional tessellations, which are usually hexagonal, but also can be rectangular, as in the case of the Jankó or Linnstrument layouts. Based on group theory and the tiling problem in mathematics, it is possible to map 2-d tessellations into higher dimensional space [7]. After carbon nanotubes (CNTs) were discovered from observations of formations of Fullerenes, the mathematical topology of carbon nanotubes became a subject of scrutiny in mathematical chemistry research [8, 9]. CNTs consisting of hexagons in their side-face are supersets of a 2-d hexagonal grid. By extensively exploring CNTs structure, overlaying existing hexagonal isomorphisms, and applying the constraints of transposition invariance and tuning invariance from isomorphism theory, we can construct musical keyboard layouts based around these shapes and perhaps open a new area of keyboard design.

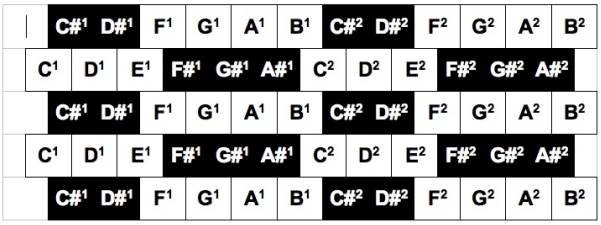


Figure 3: Jankó's piano

3. CYLINDRICAL HEXAGONAL TUBE LATTICES FROM A 2-D HEXAGONAL GRID

In this section we introduce two separate representations of coordinate systems for hexagonal lattices and see where they may meet. First, we introduce the notation used by carbon nanotube research, and second, we introduce the notation used by isomorphic musical keyboard research.

3.1 Hexagonal co-ordinates from nanotubes

If you start with a flat hexagonal lattice and begin curling, you will notice that there are a discrete number of ways that you can turn a sheet of hexagons into a tube of hexagons and have the hexagons line up properly. In order to make sure that the hexagons line up and make a complete cylindrical lattice, we explore the mathematics of the *chiral vector* a term taken from the study of nanotubes that indicates the direction in which hexagons will repeat.

A cylindrical hexagonal tube (n, m) , where $n \geq m$, is defined by a chiral vector. The definition of the chiral vector is

$$\overrightarrow{Ch} = n\overrightarrow{a1} + m\overrightarrow{a2}, \tag{1}$$

where $\overrightarrow{a1}$ and $\overrightarrow{a2}$ are two vectors within $60°$ on the grid.

In Fig. 4, we can see that $\overrightarrow{a1}$ and $\overrightarrow{a2}$ can be expressed in Cartesian coordinates (x, y) as

$$\overrightarrow{a1} = \left(\frac{3}{2}, \frac{\sqrt{3}}{2}\right) a \tag{2}$$

and

$$\overrightarrow{a2} = \left(\frac{3}{2}, -\frac{\sqrt{3}}{2}\right) a, \tag{3}$$

where a is a the length between two vertices in a hexagon.

As shown in Fig. 4, each intersection point on a 2-d hexagonal grid can be represented by using these two vectors ($\overrightarrow{a1}$ and $\overrightarrow{a2}$). When we choose an origin, the other points are labelled with the hexagonal coordinate (n, m). In Fig. 4 these points are vertices of the lattice, but this vector representation is not limited to such vertices; it can be any point inside the hexagon or on the boundary. Section 4 describes how a vector chosen in such a representation may be used to describe the tube that is produced by cutting and curling the hexagonal lattice through the third dimension, and the three varieties of tube that can be generated depending in the way in which hexagons in the lattice line up and repeat.

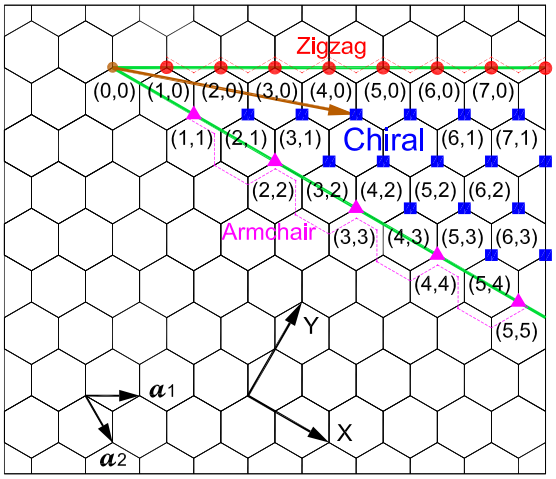


Figure 4: A 2-d hexagonal grid in Cartesian coordinates

3.2 Hexagonal co-ordinates from isomorphs

Musical isomorphisms have a different strategy for representation, one which is based on musical intervals. As described in Section 2, many different isomorphisms exist, depending on which musical interval is placed along which axis. In the original tonnetz, these intervals are major thirds and minor thirds, together making a major triad or minor triad depending on the order. Any two intervals can be combined to make an isomorphic layout, and a complete theory has been developed and presented in [10], wherein the LGD notation is introduced, with G being the greater of the two intervals, L being the lesser, and D being the difference. Hexagonal isomorphisms are thus represented based on their intervals as well as a possible rotation R and mirroring M factor, as well as shear S and an indication of the number of tones in the scale T, since this theory can be extended beyond the familiar 12-tone equal tempered scale into microtonal applications.

4. CHIRAL ANGLE AND THREE TYPES OF CYLINDRICAL HEXAGONAL TUBES

Hexagonal lattices can be curled into tubes in three different ways, shown in Fig. 5, based on the angle that we choose in Fig. 4. If we choose hexagons that are flat against each other and wrap them around to form the circumference of the tube, the pointed ends of the hexagons stick out and we call this "zigzag". If we choose to go in the other direction, with the pointed ends of the hexagons touching, we get a notched tooth appearance for the end of the tube, and this is called "armchair". If, instead, we spiral the hexagons around in a helix so that one layer builds upon the next, these are other chiral tubes, and there are many different ways we could do this.

We can also group into these three types by distinguishing the chiral angle Θ, as the angle between the chiral vector and the zigzag direction shown in Fig. 4:

$$\Theta = \tan^{-1}\left[\frac{\sqrt{3}m}{m+2n}\right] \tag{4}$$

By following Equation (4), the three types are:

Armchair $(m = n)$: $\Theta = \tan^{-1}\left[\frac{1}{\sqrt{3}}\right] = 30°$, the trace shown by purple dash line with purple triangles in Fig. 4.

Zigzag $(m = 0)$: $\Theta = \tan^{-1}[0] = 0°$, the trace shown by red dash line with red dots in Fig. 4.

Other chiral tubes (called "Chiral"): $0° < \Theta < 30°$, the area between the zigzag and armchair angles shown in Fig. 4.

Once cut and curled, these three vectors produce three types of cylindrical hexagonal tubes, shown in side view in Fig. 5.

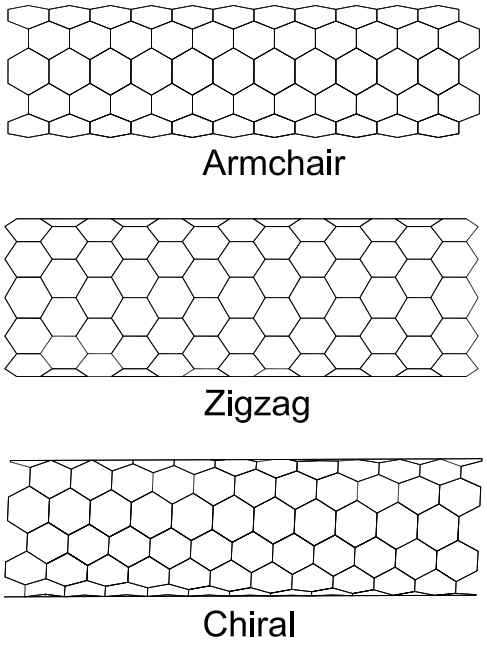


Figure 5: Three types of cylindrical hexagonal tubes

5. TUBE LENGTH AND DIAMETER

Because we need hexagons to line up perfectly in order to produce a viable tube lattice, we can't produce a tube of just any size. For example, if we consider the zigzag tube, we can only produce tubes which are a whole number of hexagons "around". The same holds true for any type.

The diameter of a tube is decided by the length of chiral vector. From equations (1), (2) and (3), the length of chiral vector is the peripheral length of the tube:

$$\|\overrightarrow{Ch}\| = \sqrt{3}a\sqrt{n^2 + nm + m^2}, \tag{5}$$

where a is the length of an edge between two vertices in a hexagon. The diameter of such a tube is therefore:

$$D = \frac{\|\overrightarrow{Ch}\|}{\pi} = \frac{\sqrt{3}a\sqrt{n^2 + nm + m^2}}{\pi}, \tag{6}$$

and for the Armchair $(m = n)$ and Zigzag $(m = 0)$ cases, Table 1 shows the corresponding tube parameters.

Tube	Chiral Length	Tube Diameter
Armchair	$3na$	$3na/\pi$
Zigzag	$\sqrt{3}na$	$\sqrt{3}na/\pi$

Table 1: Parametric descriptors of chiral vector length and tube diameter for armchair and zigzag cases.

6. MAPPING A 2-D HEXAGONAL GRID INTO A 3-D CYLINDRICAL HEXAGONAL LATTICE

If we are to be successful in curling a flat isomorphism into a tube, we must ensure that the intervals are preserved. For example, if we use a harmonic table layout (a well known layout closely resembling the original tonnetz), then adjacent hexagons must be minor thirds, major thirds or fifths. If we were to wrap this into a tube, then if we start at a given note and proceed around the tube, we must end up back at the note we started with. This puts a strict constraint on the way that isomorphisms can be wrapped: *The circumference of the tube must be in a direction in which repeated notes will be found on the original flat layout*.

Conveniently, the *LGD* representation of isomorphisms provides such a direction. In [10], the isotone axis is defined as a line contains all the instances of a particular note in an isomorphic layout. The pitch axis is a line perpendicular to the isotone axis, and is the direction in which pitch increases. Figures 6 through 13 show examples of some of the more common hexagonal isomorphisms, with their pitch axis indicated with a green arrow, and their isotone axis indicated with a dashed green line.

By setting the chiral vector direction in hexagonal coordinates (n, m), to be equal to the isotone axis in the *LGD* representation of a musical isomorphism, with an appropriately chosen chiral vector length, each isomorphic layout can be mapped from a 2-d grid into a 3-d cylindrical hexagonal lattice.

6.1 Mapping the isotone axis to the chiral vector

In *LGD* notation, either the isotone axis range or pitch axis range can be transposed by using rotation and reflection. However, since the hexagon is a member of the dihedral group [1], it is possible to focus on the area in hexagonal coordinates (n, m) with Θ (the chiral angle) as $0° \leq \Theta \leq 30°$. Besides, either *D*, $-L$ directions or $-D$, *L* directions has 60 degree opening which is the same as $\overrightarrow{a1}, \overrightarrow{a2}$ vectors. We can therefore set the isotone axis in each isomorphic layout equal to a chiral vector direction, by mapping *D* and -*L* into $\overrightarrow{a1}$ and $\overrightarrow{a2}$ directions respectively.

We can now define a new notation (D, L) which fully represents the isomorphic cylinder corresponding to the isotone axis range in the *LGD* notation. Correspondingly, the vector perpendicular to the chiral vector which is called the *translation vector* goes in the same direction as the pitch axis, and represents the direction of the axis of the resulting cylinder.

[1] Dihedral group: A mathematicaly defined set of symmetries of a regular polyhedron which includes reflection and rotation



Figure 6: Jankó (1,1)

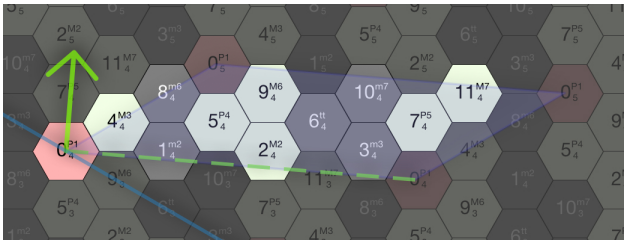


Figure 7: Harmonic Table (4,3)

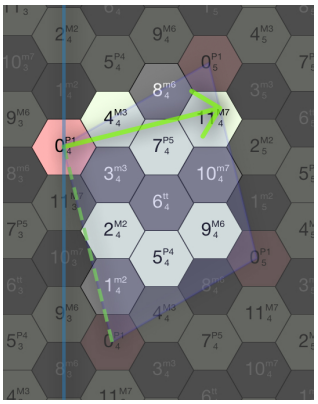


Figure 8: Gerhard (3,1)

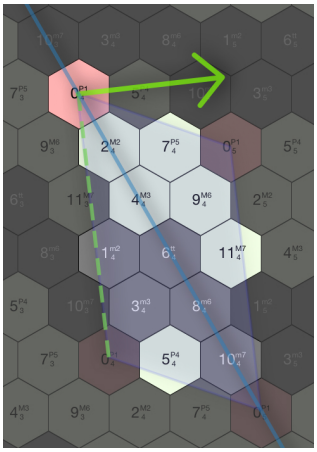


Figure 9: Park (3,2)

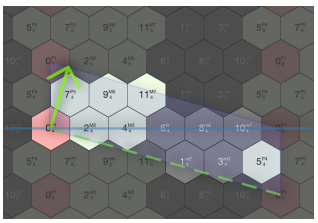


Figure 10: Wicki-Hayden (5,2)

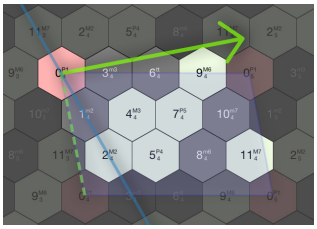


Figure 11: Bajan (2,1)

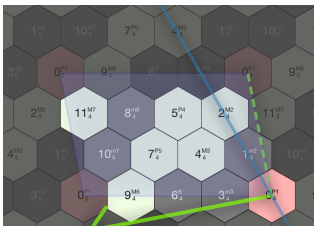


Figure 12: B-system (2,1)



Figure 13: C-system (2,1)

We can consider a subset of an isomorphic layout consisting of a single copy of each note from a single octave (in this case 12 notes since the system we are using is 12-tone equal tempered, but this could be extended to microtonal systems). This sample patch represents the smallest unit that can be considered when curling such an isomorph into a tube. Along the isotone axis, these patches repeat identically, and represent a further constraint - each tube must have around its circumference a whole number of copies of this patch.

Considering Figs. 6–13 again, we also see a blue straight line. This line represents what would be the zigzag chiral direction, and so the angle between this and the dashed green line represents the chiral vector. We have seen already that the dashed green line represents the isotone axis of the layout, and so we can see that each layout also maps to a cylindrical hexagonal lattice structure with a specific chiral vector.

The specific chiral angles of these common isomorphic layouts (in degrees to two significant digits) is calculated using equation (4), and are shown in Table 2.

Layout	(D, L)	Chiral angle
Jankó	(1,1)	30.00°
Harmonic Table	(4,3)	25.29°
Gerhard	(3,1)	13.90°
Park	(3,2)	23.41°
Wicki-Hayden	(5,2)	16.10°
Bajan	(2,1)	19.10°
B-system	(2,1)	19.10°
C-system	(2,1)	19.10°

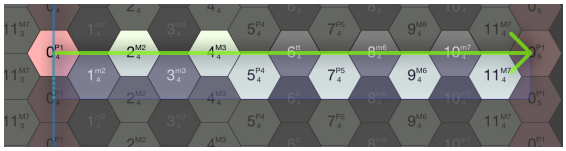
Table 2: The chiral vector of typical isomorphic layout.

6.2 Edge cases: Zigzag and Armchair

There are two special cases mentioned in [10]. The first one is where $L = 0$, which only happens for intervals 0,1,1 in *LGD* notation. This case results in a **Zigzag** type lattice (1,0). The second case is where $D=L$, which happens for intervals of 1,2,1 in *LGD* notation. This case makes the **Armchair** type lattice (1,1). The samples of those two special cases are shown in Fig. 14, and both can be seen to be instances somewhat similar to a Jankó layout.

7. BENEFITS OF A 3-D CYLINDRICAL HEXAGONAL LATTICE

We now have two new parameters that can be used to describe isomorphisms that have been curled into cylinders: The chiral angle, and the diameter. Between these two, it will be possible to explore the musical and ergonomic characteristics of different cylindrical isomorphisms, construct them into physical instruments, give them to musicians to play with, and characterize them based on playability, learnability, and expression. This exploration will be undertaken in future work, but we can begin with a theoretical discussion of some of the different playing modes and characteristics.



(a) **Zigzag** (1,0)



(b) **Armchair** (1,1)

Figure 14: Two special cases exist in the lattices

Presented here are three potential benefits of using a cylindrical hexagonal isomorphic lattice.

7.1 Boundary conditions and note reachability

One of the primary features of any arrangement of note actuators on a musical instrument is to make notes reachable. Adding additional manuals to an organ or additional strings to a bass guitar, for example, serve two purposes: to extend the range of the instrument, but also to make more notes available with less hand travel. On a traditional piano keyboard, only a little more than an octave of notes is available in any one hand position, and the ability to accurately move your hand to a new position is a critical stage in studying the piano.

Isomorphic layouts have the potential to be more compact than existing instruments, making more notes available in a single hand position and making all notes a smaller distance from the centre of the layout. However, any attempt to construct a reconfigurable hexagonal instrument that can present different isomorphisms comes to a challenge: each isomorphism potentially has a different *boundary*, that is, the overall shape of the entire layout showing all notes. Figure 15 shows the boundaries of two similar layouts.

It would be difficult to create a reconfigurable musical instrument that could represent both of these layouts to their top and bottom boundaries for two reasons. First, the angle of the boundaries is different; and second, the orientation of the hexagons is different: Wicki-hayden uses a "horizontal" layout where adjacent hexagons share a vertical face, while the Harmonic Table layout uses a "vertical" layout. Indeed, both layouts represent infinite duplications of notes to the left and right, at different angles, which would add to the challenge of manufacturing such an instrument.

Considering the parallelograms shown in Figs. 6 through 13, and extending these by repeating along the isotone axis and extending along the pitch axis, we see that each of the popular layouts will have a very different boundary shape. These boundary shapes are compared in Fig. 16. This is also related to the *shear*, a characteristic of an isomorphic layout, described in [11].

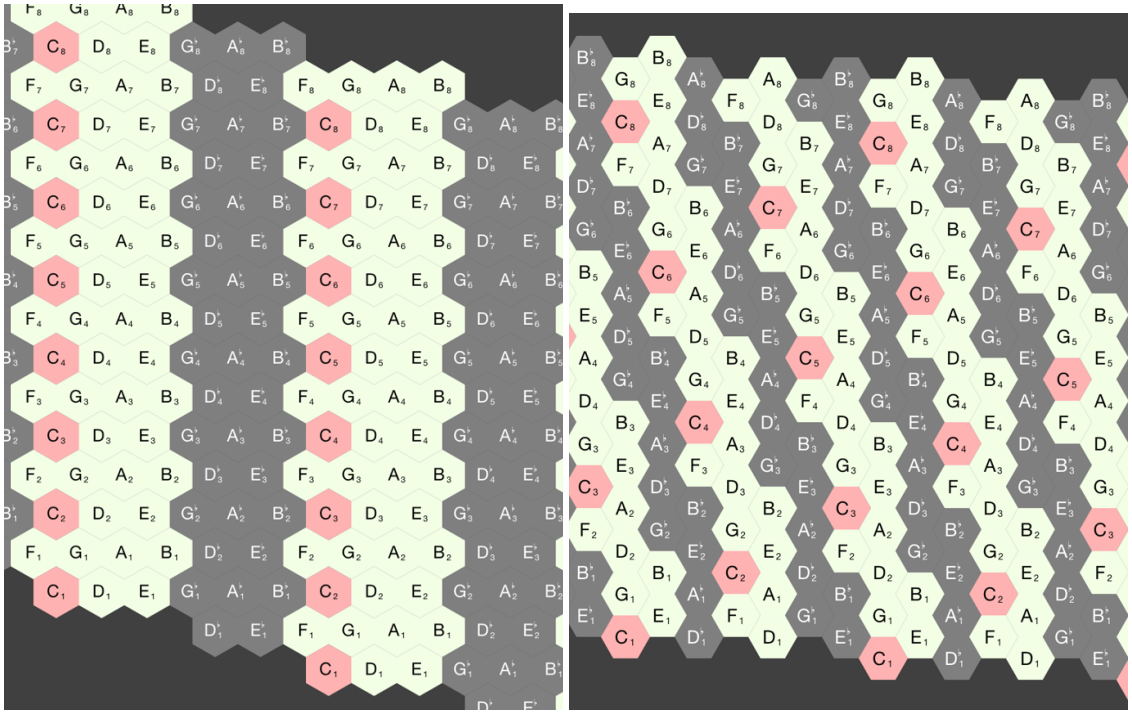


Figure 15: 8-octave boundaries of two popular isomorphisms. Left: Wicki-hayden. Right: Harmonic Table

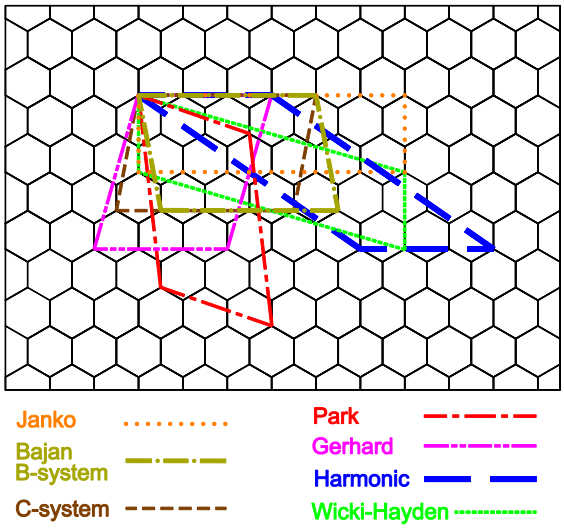


Figure 16: Parallelograms of typical isomorphic layouts

Layout	Diameter
Jankó	$\frac{3a}{\pi}$
Harmonic Table	$\frac{\sqrt{111}a}{\pi}$
Gerhard	$\frac{\sqrt{39}a}{\pi}$
Park	$\frac{\sqrt{57}a}{\pi}$
Wicki-Hayden	$\frac{\sqrt{117}a}{\pi}$
Bajan	$\frac{\sqrt{27}a}{\pi}$
B-system	$\frac{\sqrt{27}a}{\pi}$
C-system	$\frac{\sqrt{27}a}{\pi}$

Table 3: Diameters of eight typical isomorphic layouts, where a is the length of one side of a hexagon.

7.2 Wrapping infinite repetitions into tubes

Considering again the boundary shape of each isomorphism, it should be clear that the previous discussion on nanotube mapping and chiral angle can be simplified by considering an infinite sheet of repetitions of notes, and rolling that sheet in such a way that the repetitions coincide around the circumference of a tube. It should also be clear that the diameter of these tubes will be constrained to a whole number multiple of the distance between identical notes in the same octave. Table 3 shows the diameter of the tube corresponding to each of the layouts under discussion, calculated using equation (6).

When considering the construction of a physical instrument, given that there are different tube sizes required, there are two options: allow the size of the instrument to change; or allow the size of the buttons to change. Both present significant technical challenges that are not addressed in this paper and left for future work.

To map a specific isomorphic layout onto a tube with a given diameter, the length of the side of the hexagonal tiles (a) must be changed. As an example, consider the situation where two different isomorphisms are to be mapped onto a tube of a given size. The ratio of size of two hexagonal buttons can then be calculated from Table 3. Mapping *Gerhard* and *Wicki-Hayden* on the same tube, we must set:

$$\frac{\sqrt{39}a1}{\pi} = \frac{\sqrt{117}a2}{\pi}$$

Which also assumes that the both cylinders are using the same number of copies of the base set of notes around the



Figure 17: Tube size varied by the number of duplicates. (from left) 4 copies, 3, 2, and 1.

circumference. Simplifying, we get:

$$\frac{a1}{a2} = \frac{\sqrt{3}}{1}$$

which means the size of buttons in *Gerhard* layout is $\sqrt{3}$ times bigger than that in *Wicki-Hayden* layout, given the same tube diameter.

7.3 Tube Size and Note Duplication

As already discussed, the size of the tube for any given isomorphism will depend on the number of copies of the base parallelogram that are included around the circumference of the tube. This choice is aesthetic and can be used to influence playability, interaction, note availability, button size, and other factors.

Figure 17 presents a set of possible tubes from the same isomorphic layout, in this case the Gerhard layout. The only difference between tubes is the number of duplicates that go around the circumference of the tube. If a single copy is used, the tube is quite narrow and each note appears exactly once on the entire structure. Adding more duplicates makes the tube larger, but does not change the shape of any harmonic constructs on the tube.

8. PLAYABILITY MODES

Fingering on a curved keyboard can be a solution for some particular isomorphic layouts which were considered having "fingering difficulties" on 2-d planar keyboard, but this will require further study to conclusively prove. One can imagine a controller constructed with the ability to "roll" across a table or surface (Fig. 18), allowing different notes to become available at different times. With the appropriate layout, this could be an additional compositional or performance function, modulating key or tonality or adjusting other musical parameters.

It is also possible to imagine a larger cylinder with keys tiled on the inside of the surface instead of the outside. This could produce a compelling stage presence with players performing inside the lattice, and playing on the inner surface. The inside and outside tiling are shown in Fig. 19.

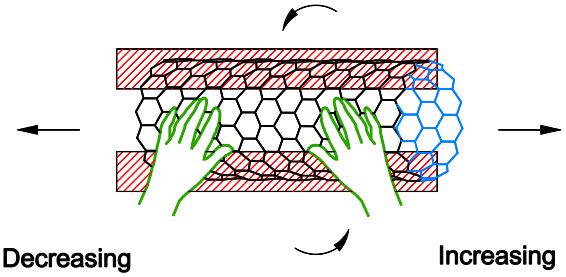


Figure 18: An appropriate area along either decreasing or increasing octave direction

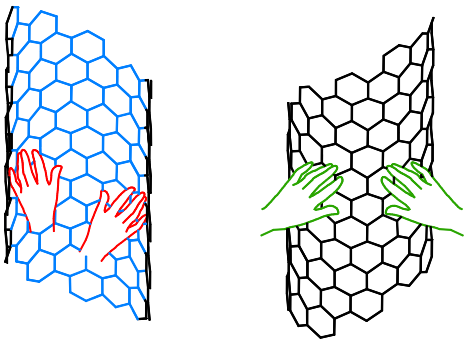


Figure 19: Playing on inner (left) or outer surface

9. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a discussion on turning the cyclic nature of tonnetz-style isomorphic discrete note layouts into a true cylindrical cycle, by setting the isotone axis of an isomorphism equal to the chiral vector direction of a nanotube. By choosing the intervals on the isomorphic axes, and by changing the number of duplicates and the size of buttons on each cylindrical hexagonal lattice, it is possible to create a wide variety of tubes of different sizes and structures, each of which maintains the strong constraints of isomorphic note arrangements while offering the possibility of new playing interfaces, compositional structures, and learning tools.

Future work on this topic will begin with brute-force generating a set of tubes for all possible isomorphisms, based on the completeness work in [10]. With this, we can explore the similarities and differences between tube layouts, as well as the musicality, playability, and interaction modes of these tubes. Next, we plan to choose some of the tubes with the greatest potential for new modes of interaction and physically construct new controllers based on this theory, and provide these to musicians, composers, and students, to explore and study. We will also formally study the musical and educational benefits of these tube structures. A long-term goal is to explore the possibility of creating a single reconfigurable tube for which the diameter and chiral angle can be modified in real time.

10. REFERENCES

[1] A. Milne, W. Sethares, and J. Plamondon, "Tuning continua and keyboard layouts," vol. 2, pp. 1–19, 03 2008, 1.

[2] L. Euler, "De harmoniae veris principiis per speculum musicum repraesentatis," in *in Novi commentarii academiae scientiarum Petropolitanae*, St. Petersburg, 1774, p. 330353.

[3] H. Riemann, "Ideen zu einer lehre von den tonvorstellungen, jahrbuch der bibliothek," pp. 21–22, 19141915.

[4] P. von Jank, "Neuerung an der unter no25282 patentirten kalviatur," vol. 25282, 1885.

[5] 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.

[6] B. Park and D. Gerhard, "Rainboard and musix: Building dynamic isomorphic interfaces," in *Proceedings of 13th International Conference on New Interfaces for Musical Expression*, 05 2013.

[7] C. Bhattacharya and R. W. Hall, "Geometrical representations of north indian thaats and raags," in *In Bridges: Mathematical Connections in Art, Music, and Science*, Pecs, Hungary, 2010.

[8] P. Schwerdtfeger, L. N. Wirz, and J. Avery, "The topology of fullerenes," in *WIREs Comput Mol Sci*, ser. 96–145, 2015.

[9] L.-C. Qin, "Determination of the chiral indice (n,m) of carbon nanotubes by election diffraction," in *Phys. Chem. Chem. Phys.*, 2007, vol. 9, pp. 31–48.

[10] B. Park and D. Gerhard, "Discrete isomorphic completeness and a unified isomorphic layout format," in *Proceedings of the Sound and Music Computing Conference*, Stockholm, Sweden, 2013.

[11] A. Prechtl, A.J.Milne, S. Holland, R.Laney, and D.B.Shape, "A midi sequencer that widens access to the compositional possibilities of novel tunings," vol. 36, no. 1, pp. 42–54, 2012.

[12] B. Park, D. Gerhard, and M. Potts. Musix. [Online]. Available: http://shiverware.com

IRISH TRADITIONAL ETHNOMUSICOLOGY ANALYSIS USING DECISION TREES AND HIGH LEVEL SYMBOLIC FEATURES

Mario L. G. Martins
Computer Music Technology Laboratory
Federal University of Technology of Parana
Av. Alberto Carazzai, 1.640 - 86300-000
Cornélio Procópio, PR, Brazil
mario@agenciaja.com

Carlos N. Silla Jr.
Computer Music Technology Laboratory
Federal University of Technology of Parana
Av. Alberto Carazzai, 1.640 - 86300-000
Cornélio Procópio, PR, Brazil
carlos.sillajr@gmail.com

ABSTRACT

In this paper we investigate the suitability of decision tree classifiers to assist the task of massive computational ethnomusicology analysis. In our experiments we have employed a dataset of 10,200 traditional Irish tunes. In order to extract features from the Irish tunes, we have converted them into MIDI files and then extracted high level features from them. In our experiments with the traditional Irish tunes, we have verified that decision tree classifiers might be used for this task.

1. INTRODUCTION

Within the Music Information Retrieval (MIR) community there is a consensus that MIR-based methods and tools might be used to assist musicologists with the task of analyzing large music collections [1–6]. This general problem is known as Computational Ethnomusicology, but it was only in the last few years that this problem started receiving more attention by the MIR researchers. In [5] the authors clarify that the term musicology is normally used by music scholars to refer to the study of European and European-derived art music traditions and for this reason the term ethnomusicology is normally used to refer to the study of art music traditions in other cultures. However, when dealing with Computational Ethnomusicology, the term ethnomusicology should be considered as the study of all the world's music [5].

There are several approaches that can be used to aid Computational Ethnomusicology [7–9]. However one important issue that should not be overlooked, when developing or using existing MIR technology to assist with Computational Ethnomusicology, is the comprehensibility of the results provided by the approach. In other data mining application domains, such as Medicine and Finance the issue of comprehensibility is high valued [10]. That is because the users of the system need to understand the reasoning of the algorithm for making that decision / suggestion. Although there are several ways that knowledge can be represented and used with different algorithms, the use of Rules is the most common one as it is easily interpretable even by non-expert users. This is one of the reasons why some of the recent research in Computational Ethnomusicology has used association rule mining algorithms [11–13]. The association rule mining algorithms generate a list of rules about facts that often happen together in a dataset. This type of algorithm allows the user to have a series of specific rules that might contain novel knowledge for the musicologist, but they do not allow the user to have a global feeling for the data.

The main contribution of this work is to explore the suitability of a decision tree classifier [14] to assist the task of Computational Ethnomusicology. Note that contrary to the association rule mining algorithms used in prior Computational Ethnomusicology research, the decision tree classifier provides additional benefits beyond just creating a rule-based representation. The first advantage is that the user can explore the tree visually. The second advantage is that the tree representation provides information about the interaction between the attributes and their values used to make the decisions. If a musicologist can understand the reasoning behind the attributes, then they can further ask for other types of information to also be made available in the data or suggest novel attributes because of the confusion being made by the classifier. However, for this to be possible, it is necessary to use high level symbolic features. For this reason in this work we employ the jSymbolic framework [15] that provides us with several high level symbolic features that can be extracted from a MIDI file.

The remainder of this paper is organized as follows: Section 2 presents a brief introduction to some of the ethnomusicology aspects of the Irish Traditional Music. Section 3 presents the experimental settings used in this work. Section 4 presents the computational experiments. Section 5 presents the conclusions and future research directions.

2. IRISH TRADITIONAL MUSIC

In this section we briefly explain some of the interesting points of the Traditional Irish Music Ethnomusicology. One definition used in this work is that as long as a tune is part of the Irish repertoire of Irish Traditional musicians, then it is considered to be part of Irish Traditional Music, regardless of the fact that it may have originated in other cultures. Therefore the Traditional Irish Musician reper-

Copyright: ©2015 First author 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.

toire also assumes that other folk songs and tunes, such as Polish and English songs, can also be considered as being part of Irish tradition. Furthermore, is remarkable how much these tunes change according to the region where the tunes are played and this can make them look like totally different tunes [16].

2.1 ABC Notation

The ABC notation is a music notation format that is expressed in plain text format. It was designed primarily for folk and traditional music of Western Europe origin (such as Traditional Irish music). It's history walks beside the growth of the Internet, where the ABC notation has become very popular. There are now lots of tunes in ABC format, from a variety of online collections. There are also several music notation software tools that are able to read the ABC notation, convert it to the standard music notation format or play it directly to the the speakers of a computer.

One of the most important aims of the ABC notation is that it can also be easily read by humans, as opposed to other computer-based music notations. Therefore, with a little practice, it is possible to play a tune directly from the ABC notation. Figure 1 presents a tune in ABC notation. Figure 2 presents a short excerpt of the same tune in standard music notation.

Cooley's REEL

```
X: 1
T: Cooley's
R: reel
M: 4/4
L: 1/8
K: Edor
|:D2|EBBA B2 EB|B2 AB dBAG|FDAD BDAD|FDAD dAFD|
EBBA B2 EB|B2 AB defg|afec dBAF|DEFD E2:|
|:gf|eB B2 efge|eB B2 gedB|A2 FA DAFA|A2 FA defg|
eB B2 eBgB|eB B2 defg|afec dBAF|DEFD E2:|
```

Figure 1. Example of ABC Notation

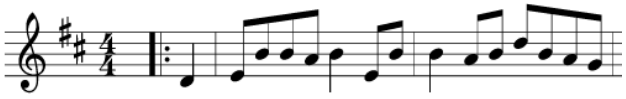


Figure 2. Cooley's (Reel)

2.2 The Session Website

The Session website (http://thesession.org/) is a non-profit endeavor as an online community dedicated to Irish traditional music. It was created in 1999, and up to this day it is maintained by its collaborators. The website contains over 13,000 Irish tunes in the ABC notation with embedded plug-ins which allow users to download the MIDI File or the Music Sheet of a given tune.

2.3 Irish Music Genres

2.3.1 Reel

One of the oldest Irish dances, reels are performed in 4/4 time. They are believed to have been played in Ireland for the first time in the late 1700s. The reel is performed either solo or in a group, in several contexts, like competitions, exhibitions or socially [17]. One of the characteristics of the reel are two groups of four notes each, adding up to an eighth-note bar. Within each group there are two heavy-light pairs [16]. Figure 2 presents a short excerpt from one of the most popular reel tunes according the thessesion.org website.

2.3.2 Barndance

A traditional genre generally performed to 4/4 rhythm, but, related to marching practice, danced to 6/8 time in North County Antrim. The barndances were most popular as social dance up to the 1950s. Its name comes from the practice of dancing in 'barns' (large sheds) which was common prior to the provision of social and meeting halls, and this assigns to this genre a rural association [17]. Figure 3 presents a short excerpt from the most popular barndance tune according the thessesion.org website.

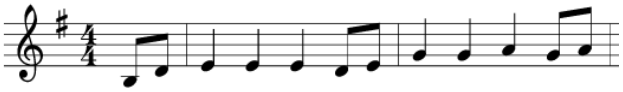


Figure 3. The Star of the Country Down (Barndance)

2.3.3 Hornpipe

Performed in 4/4 time with a characteristic dotted rhythm and with accents occurring on beats one and three. Historically, the hornpipe comes to Ireland via England, at the end of the eighteenth century, performed by professional dancers between acts in plays, and has maritime associations. This heritage gave a exhibitionistic face to Irish hornpipes [17]. It should be noted that not all hornpipes are notated with dotted rhythms and are usually played with swing even though the sheet music is not dotted. Figure 4 presents a short excerpt from the most popular hornpipe tune according the thessesion.org website.

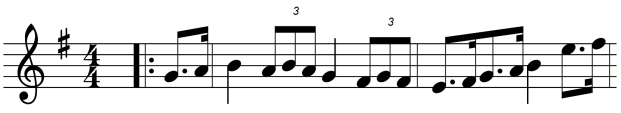


Figure 4. The Rights Of Man (Hornpipe)

2.3.4 Jig

These tunes are noted in 6/8 time, generally performed in competitive and exhibition dance contexts, choreographed by both males and females. The characteristic pattern of the jig is two groups of three quavers. These tunes are structured in two eight-bar sections, each section repeated twice (AABB) [17]. Figure 5 presents a short excerpt from

the most popular jig tune according the thessesion.org website.

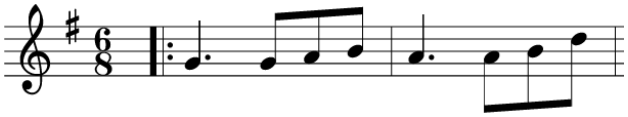


Figure 5. The Kesh (Jig)

2.3.5 Mazurka

A dance-form in 3/4 time which within the context of Irish musicians, was popularized to the greatest extent in Donegal, where it arrived in the first half of the nineteenth century. However, mazurka emerged in the Polish province of Mazovia, in the 1500s [17]. It is distinguished from its cousin waltz [16] by its unique emphasis on the second, rather than the more expected first, of the three beats. While the rhythm of the Donegal mazurkas conforms completely to the definitive Polish pattern, the phrasing structure of these correspond to that used in other Irish rhythms, and shows no connection with the mazurkas of eastern Europe [17]. Figure 6 presents a short excerpt from the most popular mazurka tune according the thessesion.org website.

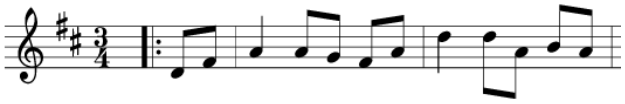


Figure 6. Sonny's (Mazurka)

2.3.6 Polka

A popular dance form notated in 2/4 time, popularly performed with march-like rhythms. It was developed in Bohemia in the early eighteenth century and arrived in Ireland in the late 1800s. It is most commonly associated with the counties of Cork, Kerry and Limerick, but polkas have spread them throughout Ireland [17]. Figure 7 presents a short excerpt from the most popular polka tune according the thessesion.org website.

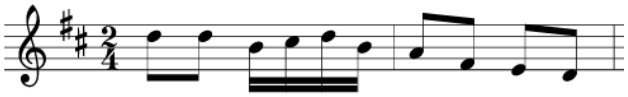


Figure 7. Ryan's (Polka)

2.3.7 Slide

A tune type associated with the jig, slides are performed faster and in 12/8 time. The predominant rhythm involves the alternation of crotchets and quavers creating the feeling of long and short. Slides are essentially dance music and the long-short rhythm of the tune is echoed by the movements of the dancers [17]. Figure 8 presents a short excerpt from the most popular slide tune according the thessesion.org website.

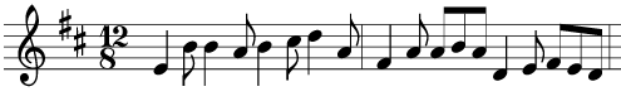


Figure 8. The Road To Lisdoonvarna (Slide)

2.3.8 Slip Jig

This different type of jig is performed in 9/8 time, generally used for group dances. Unlike the other jig types, musically the slip jig is in single form; its two-part, eight-bar music structure is not repeated. Its characteristic rhythmic pattern is three groups of three quavers. It is often referred to as "the queen of step dances" to indicate the required gracefulness of the dance [17]. Figure 9 presents a short excerpt from the most popular slip jig tune according the thessesion.org website.

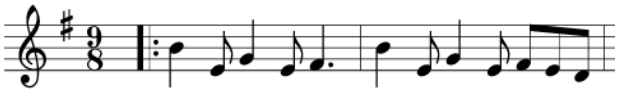


Figure 9. The Butterfly (Slip Jig)

2.3.9 Strathspey

These tunes, notated in 4/4 time, originated in Scotland in the middle of the eighteenth century. They arrived in Ireland in the late nineteenth century, in Donegal, although it never functioned for the dance there. The strathspey is in common time with each beat of a bar being accented. The tune type is particularly noted for its dotted rhythms, especially the "Scots snap", where the short note precedes the long note [17]. Figure 10 presents a short excerpt from a popular strathspey tune according the thessesion.org website.

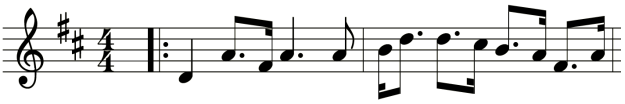


Figure 10. Calum's Road (Strathspey)

2.3.10 Three-two

Three-twos are a different march-like form performed in 3/2 time and may have origins in the the application of triple meter to the hornpipe form [18]. Figure 11 presents a short excerpt from the most popular three-two tune according the thessesion.org website.

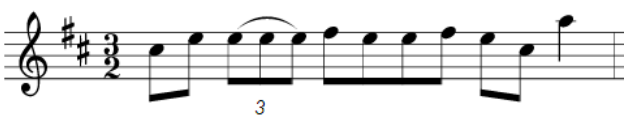


Figure 11. An Drochaid Chliuteach (Three-Two)

2.3.11 Waltz

The waltz is a dance-form in 3/4 time, particularly distinctive for the strong accent given to the first beat in each bar. Its origins are uncertain, but it may dated from the fourteenth century, unrelated to the European minuet. The tunes are generally sung or played on fiddle, and have agricultural associations [17]. Figure 12 presents a short excerpt from the most popular waltz tune according the thessesion.org website.

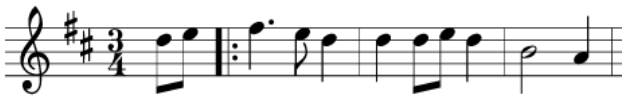


Figure 12. Si Bheag Si Mhor (Waltz)

3. EXPERIMENTAL SETTINGS

3.1 Database Construction

In order to perform our experiments we have created a novel Irish Traditional Music dataset. In order to create the dataset we have automatically retrieved all the tunes available from thesession.org website. However, the website only had the tunes in the ABC format, and in order to extract symbolic features from the tunes we needed to convert this data to the MIDI format. Therefore, the creation of the dataset was composed of three main stages: (1) ABC download; (2) ABC to MIDI conversion; (3) MIDI (Symbolic) Feature Extraction. Table 1 shows the final result of this process.

3.1.1 ABC Download

In order to automatically retrieve all the tunes from thesseion.org website we developed an algorithm that uses the PHP function cURL in a loop. The cURL function returns a string with the retrieve HTML webpage for a given URL. For each tune webpage we have used a set of regular expressions to find and extract the ABC notation within a given webpage. After the extraction of the tune in ABC format we place it in a specific folder according to the Traditional Irish genre. With this procedure we managed to retrieve 11,980 tunes in the ABC format.

3.1.2 ABC to MIDI Conversion

As mentioned earlier in order to extract symbolic features to create the dataset, we need the tunes to be in the MIDI format. Therefore, in order to convert the downloaded ABC tunes to the MIDI format we have employed an online converter. Note that the number of successfully converted ABC to MIDI tunes was 10,200 MIDI files.

3.1.3 MIDI (Symbolic) Feature Extraction

After the tunes have been converted into the MIDI format, we have used the jSymbolic [1] software [15] to extract high level symbolic features from the MIDI files. This is particularly important, as high level symbolic features can be interpreted by musicologists. With the jSymbolic software we extracted a total of 1,022 high-level symbolic features that fall into the broad categories of texture, rhythm, dynamics, pitch statistics, melody and instrumentation.

[1] Available at: http://jmir.sourceforge.net/jSymbolic.html

Tune type	Number of tunes	Relative Number
barndances	298	2,92%
hornpipes	843	8,26%
jigs	2,666	26,14%
mazurkas	116	1,14%
polkas	695	6,81%
reels	3,864	37,88%
slides	228	2,24%
slip jigs	380	3,73%
strathspey	329	3,32%
three-twos	78	0,76%
waltz	703	6,89%

Table 1. Number of Irish Traditional Tunes

Tune type	Precision	Number of leaves
barndance	26,1%	55
hornpipe	84,7%	38
jig	100%	1
mazurka	49,5%	9
polka	100%	1
reel	94,2%	35
slide	100%	1
slip jig	100%	1
strathspey	60%	39
three-two	100%	1
waltz	91,3%	15

Table 2. Precision by genre

4. RESULTS

In this section we are interested in answering the following questions by using controlled experiments: How well can a Decision Tree Classifier predict the genre labels of the 11 Irish music genres? Can the generated decision tree be used as a tool to assist Computational Ethnomusicology?

4.1 Irish Music Genre Classification

In order to perform the experiments reported in this section we have used the J48 Decision tree classifier implementation of the WEKA data mining toolkit [19]. The experiments were performed using stratified ten-fold cross-validation.

The experimental results of the classification experiment are presented in Table 2. The analysis of the results presented in Table 2 shows that overall, the precision across all Traditional Irish genres was high (92,3%). This result means that 9,418 tunes were correctly classified. However, It should be noted that in some particular Irish Traditional genres, like barndance, the obtained results are much lower

a	b	c	d	e	f	g	h	i	j	k	classified as
64	51	0	0	0	160	0	0	23	0	0	a = barndance
38	739	0	0	0	9	0	0	57	0	0	b = hornpipe
0	0	2666	0	0	0	0	0	0	0	0	c = jig
0	0	0	54	0	0	0	0	0	0	62	d = mazurka
0	0	0	0	695	0	0	0	0	0	0	e = polka
118	20	0	0	0	3684	0	0	42	0	0	f = reel
0	0	0	0	0	0	228	0	0	0	0	g = slide
0	0	0	0	0	0	0	380	0	0	0	h = slip jig
25	63	0	0	0	58	0	0	183	0	0	i = strathspey
0	0	0	0	0	0	0	0	0	78	0	j = three two
0	0	0	55	0	0	0	0	0	0	647	k = waltz

Table 3. Confusion Matrix

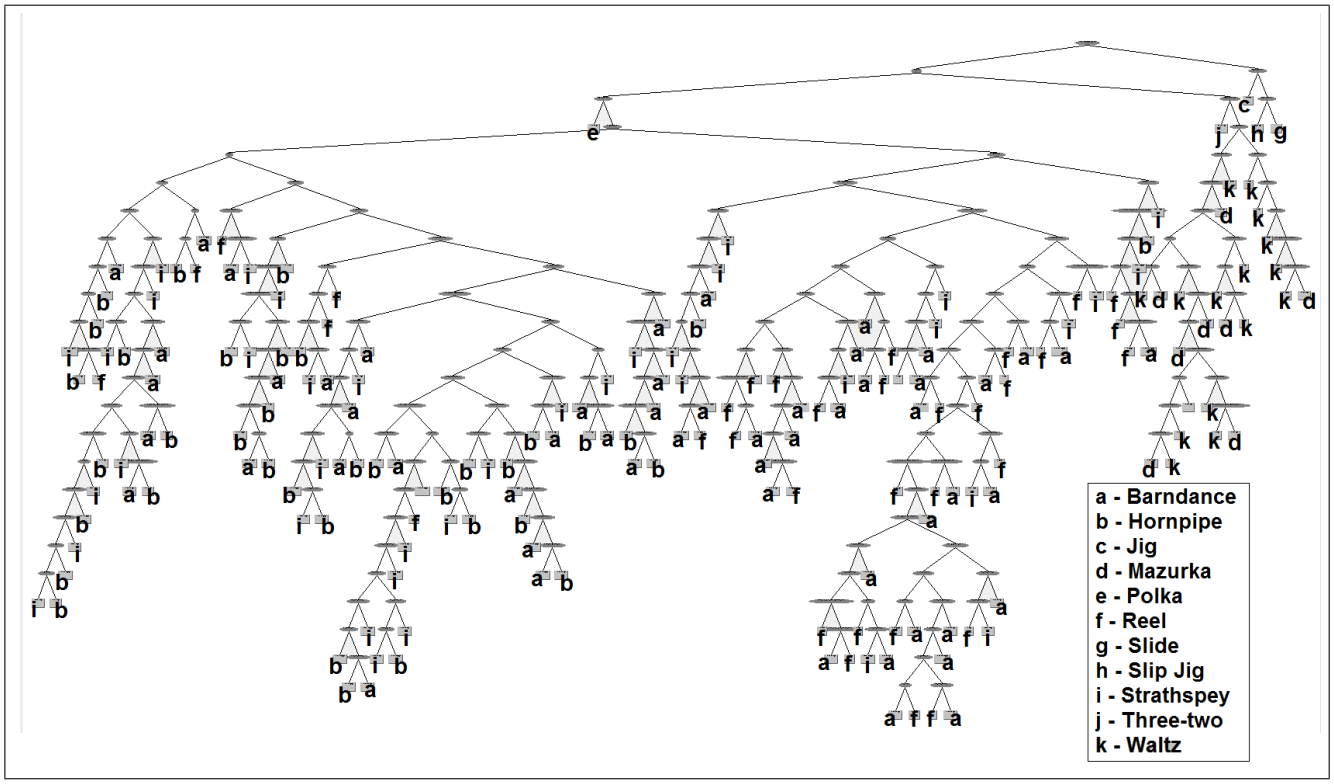


Figure 13. General vision of the decision tree.

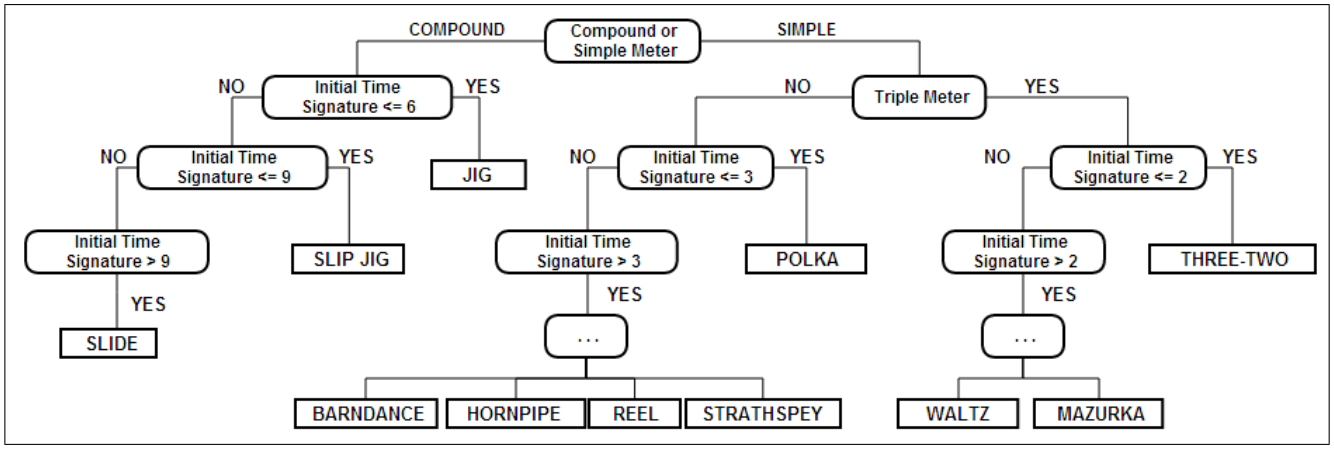


Figure 14. Three first levels of the generated tree.

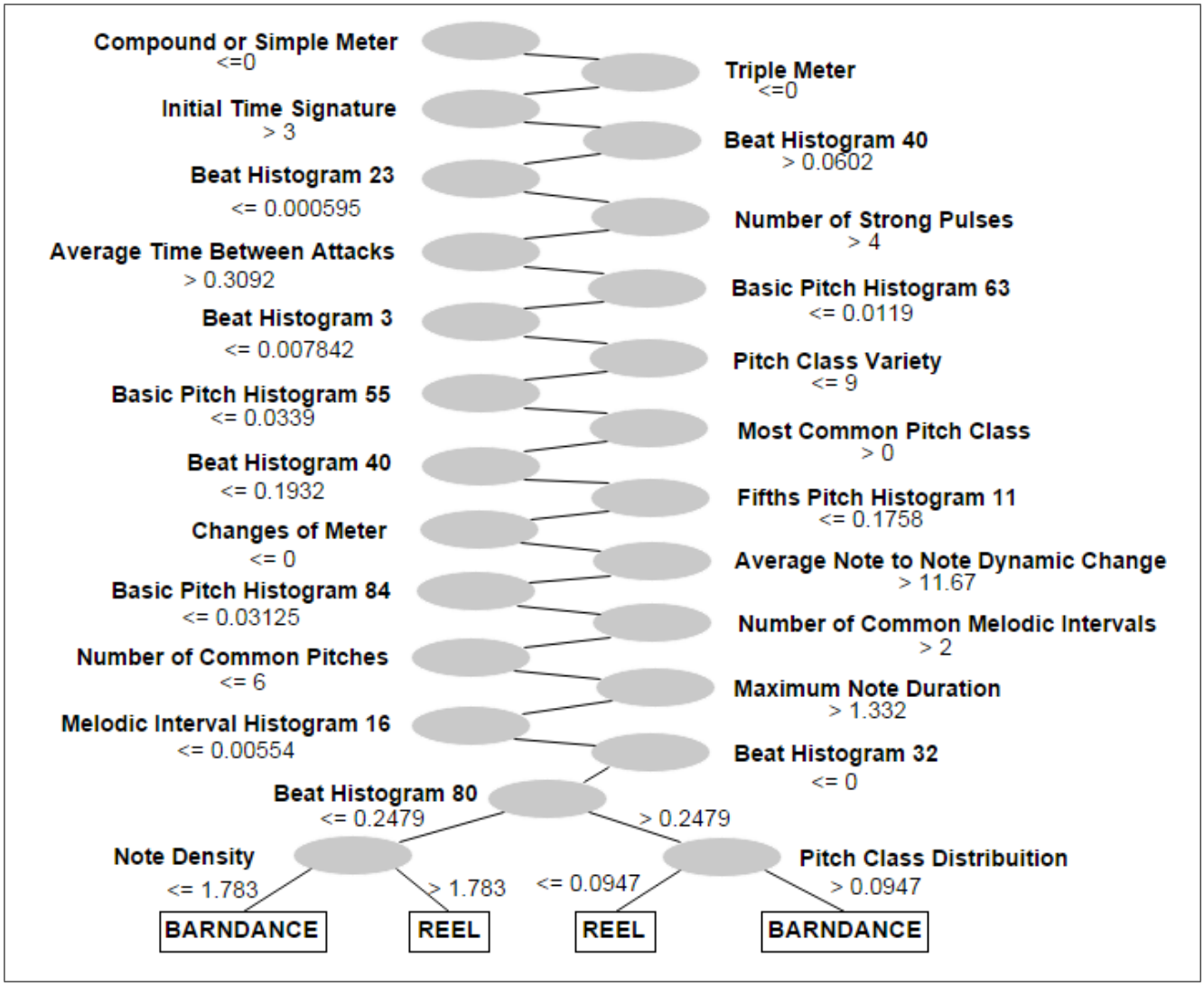


Figure 15. Deepest path of the generated decision tree

than this overall result. In order to understand the confusions being made by the decision tree classifier, let us analyse the confusion matrix presented in Table 3.

The analysis of the confusion matrix presented in Table 3 shows that the low precision for the tunes of the barndances Irish Traditional genre corresponds to the same misconception made by naive listeners due the musical similarity of the genres. In the case of the barndances, the genre that had the lowest precision, the tunes were mostly classified as reels. This misclassification might be due to the fact that both genres has the same Initial Time Signature, i.e. 4/4. The same type of error happens with mazurkas, that are usually confused with its cousin, the waltz [16].

4.2 Analysis of the Generated Tree

One of the advantages of using a decision tree classifier is that it allows us to visually inspect the generated tree and that it also outputs the classification rules. The decision tree algorithm used in this work had as the input a feature vector of 1,022 features (described in [20]) for each tune. The final generated decision tree employed 105 attributes (out of the 1,022) to create a decision tree with 393 rules and 197 leaves. Figure 13 presents a general vision of the tree, where the letters represents the genres at the leaves.

The first levels of the generated tree are presented in detail in Figure 14. The analysis of just these first three levels of the tree shows some interesting insights. First, it shows what an individual who is not trained in Irish Traditional music perceives while listening to the different Irish Traditional Tunes. A naive listener may perceive that a given tune is is in a 3/2 time signature (although not necessarily naming it is a 3/2 time signature). However this naive listener will have trouble pin-pointing whether this song is a waltz or a mazurka. That is of course, only if this listener knows Mazurkas exist, otherwise the listener will classify it as a Waltz.

Second, one important aspect of a decision tree is that it visually shows the importance of each and how they were used (i.e. with which values and/or decision splits) to make a decision. By using decision trees in combination with high level symbolic features, this information can aid musicologists to validate the automated approaches and/or look closely at the data to verify if a mistake was really made. It might even be possible for musicologists to

request the creation of new high level symbolic features from the Music Information Retrieval research community based on their knowledge if they verify that there is important information missing. Due to space limitations it is not possible to plot in detail the complete generated decision tree, however in Figure 15 we present the deepest path of the generated decision tree in detail. The analysis of Figure 15 shows that the distinction between two of Irish music genres involves several different aspects of the tunes.

Third, in this experiments we have only looked at one aspect of interest by musicologists, i.e. the classification of genres (and more importantly what are the different properties that distinguishes them). We argue that this same approach (using high level symbolic features with a decision tree classifier) might be used to assist with other musicological tasks such as auto tagging and music discovery, among others.

5. CONCLUSIONS

In this work we have showed that a decision tree classifier might be used as a Computation Ethnomusicology tool to assist musicologists. In order to perform our experiments, we have created a novel dataset with 10,200 Irish Traditional Tunes obtained from the www.thessesion.org website. The tunes are available on the website using the ABC notation and therefore we converted all the tunes into MIDI files. We then used the jSymbolic feature extractor to obtain high level symbolic features from the MIDI files. With the high level symbolic features extracted from the MIDI files, we have trained a decision tree classifier, which correctly classified 9,418 tunes.

After the classification, the decision tree classifier has the advantage of producing a graphical model (a decision tree) and also a set of rules. By analyzing the decision tree, it became clear that on the first three levels of the tree, the high level symbolic features of Compound or Simple Meter, Triple Meter and Initial Time Signature were shown to correspond to the perception made by a naive listeners, i.e. the distinction between the different Irish Traditional genres, starts by using the rhythm information. This particular result shows that decision trees might be used to aid musicologists.

As future research we plan to perform experiments with other rule-generating methods on different datasets and to use the methodology proposed in this paper to other computational ethnomusicology tasks.

Acknowledgments

We thank the anonymous reviewers for their very valuable feedback.

6. REFERENCES

[1] K. Hartmann, D. Buchhner, A. Berndt, and A. Nunberger, "Interactive data mining & machine learning techniques for musicology," in *Proc. of the 3rd Conf. on Interdisciplinary Musicology*, 2007.

[2] O. Cornelis, M. Lesaffre, D. Moelants, and M. Leman, "Access to ethnic music: Advances and perspectives in content-based music information retrieval," *Signal Processing*, vol. 90, pp. 1008–1031, 2010.

[3] P. v. Kranenburg, J. Garbers, A. Volk, F. Wiering, L. P. Grijp, and R. C. Veltkamp, "Collaboration perspectives for folk song research and music information retrieval: The indispensable role of computational musicology," *Journal of Interdisciplinary Music Studies*, pp. 17–43, 2010.

[4] K. Neubarth, M. Bergeron, and D. Conklin, "Associations between musicology and music information retrieval," in *Proc. of the 12th Int. Conf. on Music Information Retrieval*, 2011, pp. 429–434.

[5] G. Tzanetakis, A. Kapur, W. A. Schloss, and M. Wright, "Computational ethnomusicology," *Journal of Interdisciplinary Music Studies*, pp. 1–24, 2007.

[6] S. Oramas and O. Cornelis, "Past, present and future in ethnomusicology: the computational challenge," in *Proc. of the 13th Int. Conf. on Music Information Retrieval*, 2012.

[7] W. Chai and B. Vercoe, "Folk music classification using hidden markov models," in *Proc. of International Conference on Artificial Intelligence*, 2001.

[8] C. Guastavino, F. Gomez, G. Toussaint, F. Marandola, and E. Gomez, "Measuring similarity between flamenco rhythmic patterns," in *Journal of New Music Research*, 2009, pp. 129–138.

[9] R. Hillewaere, B. Manderick, and D. Conklin, "String methods for folk tune genre classification," in *Proc. of the 13th Int. Conf. on Music Information Retrieval*, 2012, pp. 217–222.

[10] A. A. Freitas, "Comprehensible classification models: a position paper," *ACM SIGKDD Explorations*, vol. 15, no. 1, pp. 1–10, 2013.

[11] J. Taminau, R. Hillewaere, S. Meganck, D. Conklin, A. Nowe, and B. Manderick, "Descriptive subgroup mining of folk music," in *International Workshop on Machine Learning and Music*, 2009, pp. 1–6.

[12] K. Neubarth, I. Goienetxea, C. G. Johnson, and D. Conklin, "Association mining of folk music genres and toponyms," in *Proc. of the 13th Int. Conf. on Music Information Retrieval*, 2012, pp. 7–12.

[13] K. Neubarth, C. G. Johnson, and D. Conklin, "Discovery of mediating association rules for folk music analysis," in *International Workshop on Machine Learning and Music*, 2013.

[14] J. R. Quinlan, "Induction of decision trees," *Machine Learning 1*, pp. 81–106, 1986.

[15] C. McKay and I. Fujinaga, "jsymbolic: A feature extractor for midi files," in *Proceedings of the International Computer Music Conference*, 2006, pp. 302–305.

[16] A. Ng, "irishtune.info," http://www.irishtune.info.

[17] F. Vallely, Ed., *Companion to Irish Traditional Music*. NYU Press, 1999.

[18] J. Keith, "thesession.org," http://www.thessesion.org.

[19] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005.

[20] C. McKay, "Automatic genre classification of midi recordings," Ph.D. dissertation, McGill University, 2004.

NAVIGATING THE MIX-SPACE: THEORETICAL AND PRACTICAL LEVEL-BALANCING TECHNIQUE IN MULTITRACK MUSIC MIXTURES

Alex Wilson
Acoustics Research Centre
School of Computing, Science and Engineering
University of Salford
a.wilson1@edu.salford.ac.uk

Bruno M. Fazenda
Acoustics Research Centre
School of Computing, Science and Engineering
University of Salford

ABSTRACT

The mixing of audio signals has been at the foundation of audio production since the advent of electrical recording in the 1920's, yet the mathematical and psychological bases for this activity are relatively under-studied. This paper investigates how the process of mixing music is conducted. We introduce a method of transformation from a "*gain-space*" to a "*mix-space*", using a novel representation of the individual track gains. An experiment is conducted in order to obtain time-series data of mix engineers exploration of this space as they adjust levels within a multitrack session to create their desired mixture. It is observed that, while the exploration of the space is influenced by the initial configuration of track gains, there is agreement between individuals on the appropriate gain settings required to create a balanced mixture. Implications for the design of intelligent music production systems are discussed.

1. INTRODUCTION

The task of the mix engineer can be seen as one of solving an optimisation problem [1], with potentially thousands of variables once one considers the individual level, pan position, equalisation, dynamic range processing, reverberation and other parameters, applied in any order, to many individual audio components.

The objective function to be optimised varies depending on implementation. Conceptually, one should maximise '*Quality*', an often-debated concept in the case of music production. In this context, borrowing from ISO 9000 [2], we can consider '*Quality*' to be the degree to which the inherent characteristics of a mix fulfil certain requirements. These requirements may be defined by the mix engineer, the artist, the producer or some other interested party. In a commercial sense, we consider the requirement to be that the mix is enjoyed by a large amount of people.

This paper considers how the mix process could be represented in a highly simplified case, investigates how high-quality outcomes are achieved by human mixers and offers insights into how such results could be achieved by intelligent music production systems.

Copyright: ©2015 Alex Wilson 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.

2. BACKGROUND

For many decades the mixing console has retained a recognisable form, based on a number of replicated channel strips. Audio signals are routed to individual channels where typical processing includes volume control, pan control and basic equalisation. Channels can be grouped together so that the entire group can be processed further, allowing for complex cross-channel interactions.

One of the most fundamental and important tasks in music mixing is the choice of relative volume levels of instruments, known as level-balancing. Due to its ubiquity and relative simplicity, level-balancing using fader control is a common approach to the study of mixing. It has been indicated that balance preferences can be specific to genre [3] and, for expert mixers, can be highly consistent [4].

As research in the area has continued, a variety of assumptions regarding mixing behaviours have been put forward and tested. A number of automated fader control systems have used the assumption that equal perceptual loudness of tracks leads to greater inter-channel intelligibility [5, 6]. This particular practice was investigated in a study of "best-practice" concepts [7], which included panning bass-heavy content centrally, setting the vocal level slightly louder than the rest of the music or the use of certain instrument-specific reverberation parameters. A number of these practices were tested using subjective evaluation and the equal-loudness condition did not necessarily lead to preferred mixes [7].

Much of these "best-practice" techniques may be anecdotal, based on the experience of a small number of professionals who have each produced a large number of mixes (see [8, 9] for reviews). Due to the proliferation of the Digital Audio Workstation (DAW) and the sharing of software and audio via the internet, it has now become possible to reverse this paradigm, and study the actions of a large number of mixers on a small number of music productions. This allows both quantitative and qualitative study of mixing practice, meaning the dimensions of mixing and the variation along these dimensions can be investigated.

To date, there have been few quantitative studies of complete mixing behaviour, as lack of suitable datasets can be problematic. One such study focussed on how a collection of students mixed a number of multitrack audio sessions [10]. It was shown that, among low-level features of the resultant audio mixes, most features exhibited less variance across mixers than across songs.

3. THEORY

When considering a realistic mixing task the number of variables becomes very large. An equaliser alone may have dozens of parameters, such as the center frequency, gain, bandwidth and filter type of a number of independent bands, leading to a large number of combinations. There are methods to reduce the number of variables in these situations. In [11], the combination of track gains and simple equalisation variables was reduced to a 2D map by means of a self-organising map, where the simple equalisation parameter was the first principal component of a larger EQ system, showing further dimensionality reduction. While these approaches can create approximations of the mix-space, the true representation is difficult to conceive for all but the most simple mixing tasks.

3.1 Defining the "mix-space"

We introduce a new definition for *"mix-space"*. Fig. 1 shows a trivial example of just two tracks. When mixing, the gains of the two tracks, g_1 and g_2, are adjusted. Here it can be seen that, using polar coordinates, the angle ϕ provides most information about the mix, as it is the proportional blend of g_1 and g_2. Any other point on the line at angle ϕ would represent the same balance of instruments, thus r is a scaling factor, corresponding to the combined mix volume. As the gains are normalised to [0,1], ϕ is bound from 0 to $\pi/2$ radians.

For a system of n audio signals, $x_1(t), \ldots, x_n(t)$, we can define an n-dimensional *gain-space* with time-varying gains $g_1(t), \ldots, g_n(t)$. As the n gains are adjusted this *gain-space* is explored. Consider the case when all *n* gains are increased or decreased by an equal amount. While there is a clear displacement in the gain-space, there is no change to the overall mix, only a change in volume. Acknowledging this, and by extending the concept shown in Fig. 1, the hyperspherical co-ordinates of a point in the gain-space are used to transform to the mix-space. This co-ordinate system, written as $(r, \phi_1, \phi_2, \ldots, \phi_{n-1})$, is defined by Eqn. 1.

$$r = \sqrt{{g_n}^2 + {g_{n-1}}^2 + \cdots + {g_2}^2 + {g_1}^2} \tag{1a}$$

$$\phi_1 = \arccos \frac{g_1}{\sqrt{{g_n}^2 + {g_{n-1}}^2 + \cdots + {g_1}^2}} \tag{1b}$$

$$\phi_2 = \arccos \frac{g_2}{\sqrt{{g_n}^2 + {g_{n-1}}^2 + \cdots + {g_1}^2}} \tag{1c}$$

$$\vdots$$

$$\phi_{n-2} = \arccos \frac{g_{n-2}}{\sqrt{g_n^2 + {g_{n-1}}^2 + {g_{n-2}}^2}} \tag{1d}$$

$$\phi_{n-1} = \begin{cases} \arccos \frac{g_{n-1}}{\sqrt{g_n^2 + {g_{n-1}}^2}} & g_n \geq 0 \\ 2\pi - \arccos \frac{g_{n-1}}{\sqrt{g_n^2 + {g_{n-1}}^2}} & g_n < 0 \end{cases} \tag{1e}$$

Consider a system of four tracks, as shown in Fig. 2. Here, ϕ_3 denotes the balance of the drum and bass tracks, to form the rhythmic foundation of the mix. ϕ_2 describes the projection of this balance onto the guitar dimension,

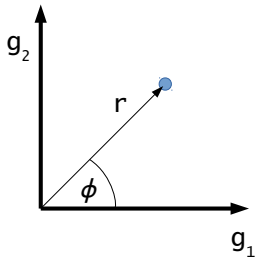


Figure 1: The point represents a balance of two instruments, controlled by gains g_1 and g_2. Any other point on the line at angle ϕ would represent the same balance of instruments, thus r is a scaling factor.

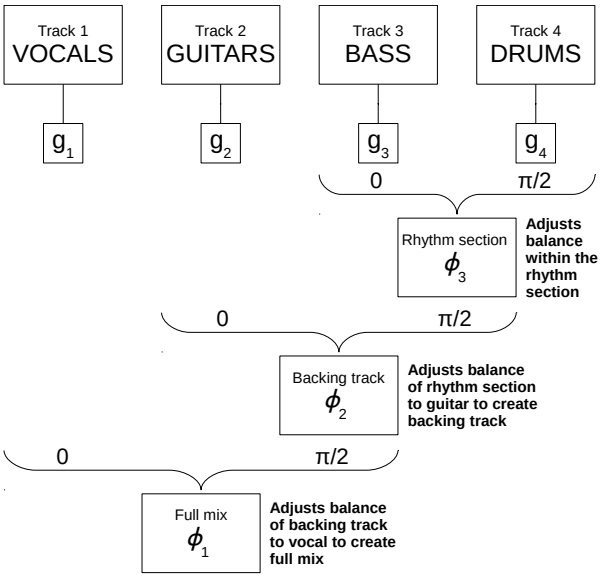


Figure 2: Schematic representation of a four-track mixing task and the semantic description of the three ϕ terms.

and thus, the complete musical backing track. ϕ_1 then describes the balance between this backing track and the vocal. Using this notation, ϕ_1 has been studied in isolation in previous studies [3, 4]. For a system with four tracks only three ϕ terms must be determined to construct the mix-space. Convention typically dictates that ϕ_{n-1} describes an equatorial plane and ranges over $[0, 2\pi)$ and that all other angles range from $[0, \pi]$, however since all gains are positive, each angle ranges over $[0, \pi/2]$, as in Fig. 1.

Since *r* is a scaling factor, when the values of all ϕ terms are held constant, there is a constant difference in the relative gains of each track, when expressed in decibels. This can be illustrated by converting ϕ terms back to gain terms, which can be achieved using Eqn. 2.

$$g_1 = r\cos(\phi_1) \tag{2a}$$

$$g_2 = r\sin(\phi_1)\cos(\phi_2) \tag{2b}$$

$$g_3 = r\sin(\phi_1)\sin(\phi_2)\cos(\phi_3) \tag{2c}$$

$$\vdots$$

$$g_{n-1} = r\sin(\phi_1)\cdots\sin(\phi_{n-2})\cos(\phi_{n-1}) \tag{2d}$$

$$g_n = r\sin(\phi_1)\cdots\sin(\phi_{n-2})\sin(\phi_{n-1}) \tag{2e}$$

3.2 Characteristics of the mix-space

With a mix-space having been defined, what characteristics does the space have? How does the act of mixing explore this space? We now discuss three scenarios - beginning at a '*source*', exploring the '*mix-space*' and arriving at a '*sink*'

3.2.1 The 'source'

In a real-world context, when a mixer downloads a multitrack session and first loads the files into a DAW, each mixer will initially hear the same mix, a linear sum of the raw tracks [1] . While each of these raw tracks can be presented in various ways if we presume each track is recorded with high signal-to-noise ratio (as would have been more important when using analogue equipment) then, with all faders set to 0dB, the perceived loudness of those tracks with reduced dynamic range (such as synthesisers, electric bass and distorted electric guitars) would be higher than that of more dynamic instruments.

Much like the final mixes, this initial 'mix' can be represented as a point in some high-dimensional, or feature-reduced, space. It is rather unlikely that a mixer would open the session, hear this mix and consider it ideal, therefore, changes will most likely be made in order to move away from this location in the space. For this reason, this position in the mix-space is referred to as a '*source*'.

In practice, the session, as it has been received by the mix engineer, may be an "unmixed sum" or may be a rough mix, as assembled by the producer or recording engineer. In a real-world scenario, the work may be received as a DAW session, where tracks have been roughly mixed. Alternatively, where multitrack content is made available online, such as in mix competitions, the unprocessed audio tracks are usually provided without a DAW session file. The latter approach is assumed in this study, in order for mix engineers to have full creative control over the mixing process. If mixers were to make unique changes to the initial configuration then that source can be considered to be radiating omni-directionally in the mix-space. However, it is possible that, for a given session, there may be some changes which will seem apparent to most mixers, for example, a single instrument which is louder than all others requiring attenuation. For such sessions, the source may be unidirectional, or if a number of likely outcomes exist, there may exist a number of paths from the source.

3.2.2 Navigating the mix-space

The path from the *source* to the final mix could be represented as a series of vectors in the *mix-space*, henceforth named '*mix-velocity*', and defined in Eqn. 3, for the three dimensions shown in Fig. 2.

$$u_t = \phi_{(1,t)} - \phi_{(1,t-1)} \tag{3a}$$

$$v_t = \phi_{(2,t)} - \phi_{(2,t-1)} \tag{3b}$$

$$w_t = \phi_{(3,t)} - \phi_{(3,t-1)} \tag{3c}$$

If all mixers begin at the same *source* then a number of questions can be raised in relation to movement through the *mix-space*.

- Moving away from the *source*, at what point do mix engineers diverge, if at all?
- How do mix engineers arrive at their final mixes? What paths through the *mix-space* do they take?
- Do mix engineers eventually converge towards an ideal mix?

3.2.3 The 'sink'

Complementary to the concept of a *source* in the *mix-space*, a '*sink*' would represent a configuration of the input tracks which produces a high-quality mix that is apparent to a sizeable portion of mix engineers and to which they would mix towards. As the concept of quality in mixes is still relatively unknown there are a number of open questions in the field which can be addressed using this framework.

- Is there a single sink, i.e. one ideal mix for each multitrack session? In this case the highest mix-quality would be achieved at this point.
- Are there multiple sinks, i.e. given enough available mixes, are these mixes clustered such that one can observe a number of possible alternate mixes of a given multitrack session? These multiple sinks would represent mixes that are all of high mix-quality but audibly different.

4. EXPERIMENT

To the authors' knowledge, there is a lack of appropriate data available to directly test the theory presented in Section 3. In order to examine how mix engineers navigate the mix-space a simple experiment was conducted. In this instance the mixing exercise is to balance the level of four tracks, using only a volume fader for each track. Importantly, the participants will all begin with a predetermined balance, in order to examine the source directivity. This experiment aims to answer the following research questions:

Q1. Can the source be considered omni-directional or are there distinct paths away from the source?

Q2. Is there an ideal balance (single sink)?

Q3. Are there a number of optimal balances (multiple sinks)?

Q4. What are the ideal level balances between instruments?

[1] Here it is significant that a DAW typically defaults to faders at 0dB, while a separate mixing console may default to all faders at -∞dB. This allows an experimenter to ensure that all mixers begin by hearing the same 'mix'. This has been referred to in previous studies as an 'unmixed sum' or a 'linear sum'. While the term 'unmixed' can be misleading, it does reflect the fact that the artistic process of mixing has not yet begun.

Previous studies have indicated that perceptions of quality and preference in music mixtures are related to subjective and objective measures of the signal, with distortion, punch, clarity, harshness and fullness being particularly important [12, 13]. By using only track gain and no panning, equalisation or dynamics processing, most of these parameters can be controlled.

4.1 Stimuli

The multitrack audio sessions used in this experiment have been made available under a creative commons license [2] [3]. These files are also indexed in a number of databases of multitrack audio content [4] [5] Three songs were used for this experiment, which consisted of vocals, guitar, bass and drums, as per Fig. 2, and as such the interpretations of ϕ_n from here on are those in Fig. 2.

The four tracks used from "*Borrowed Heart*" are raw tracks, where no additional processing has been performed apart from that which was applied when the tracks were recorded [6]. The tracks from "*Sister Cities*" also represent the four main instruments but were processed using equalisation and dynamic range compression. These can be referred to as 'stems', as the 11 drum tracks have been mixed down, the two bass tracks (a DI signal and amplifier signal) have been mixed together, the guitar track is a blend of a close and distant microphone signals and the vocal has undergone parallel compression, equalisation and subtle amounts of modulation and delay. In the case of "*Heartbeats*", the tracks used are complete 'mix stems', in that the song was mixed and bounced down to four tracks consisting of 'all vocals', 'all music' (guitars and synthesisers), 'all bass' and 'all drums'. For testing, the audio was further prepared as follows:

- 30-second sections were chosen, so that participants would be able to create a static mix, where the desired final gains for each track are not time-varying.
- Within each song, each 30-second track was normalised according to loudness. In this case, loudness is defined by BS.1770-3, with modifications to increase the measurements suitability to single instruments, rather than full-bandwidth mixes [14]. This allows the relative loudness of instruments to be determined directly from the mix-space coordinates.
- For each song, two source positions were selected. The ϕ terms were selected using a random number generator, with two constraints: to ensure the two sources are sufficiently different, the pair of sources must be separated by unit Euclidean distance in the mix-space and to ensure the sources are not mixes where any track is muted, the values were chosen from the range $\pi/8$ to $3\pi/8$ (see Fig. 2).

[2] http://weathervanemusic.org/shakingthrough
[3] http://www.cambridge-mt.com/ms-mtk.htm
[4] http://multitrack.eecs.qmul.ac.uk/
[5] http://medleydb.weebly.com/
[6] https://s3.amazonaws.com/tracksheets/Hezekiah+Jones+-+Tracksheet.xlsx

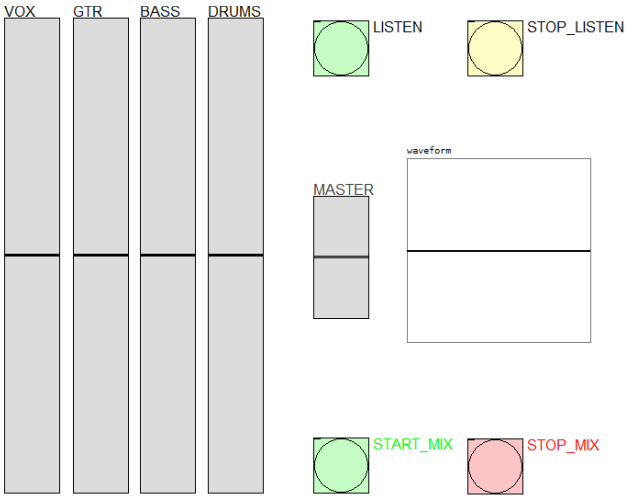


Figure 3: GUI of mixing test. The faders are unmarked and all begin at the same central value, which prevents participants from relying on fader position to dictate their mix.

4.2 Test panel

In total, 8 participants (2 female, 6 male) took part in the mixing experiment. As staff and students within Acoustics, Digital Media and Audio Engineering at University of Salford, each of these participants had prior experience of mixing audio signals. The mean age of participants was 25 years and none reported hearing difficulties.

4.3 Procedure

Rather than use loudspeakers in a typical control room, the test set-up used a more neutral reproduction. The experiment was conducted in a semi-anechoic chamber at University of Salford, where the background noise level was negligible. Audio was reproduced using a pair of Sennheiser HD800 headphones, connected to the test computer by a Focusrite 2i4 USB interface. Due to the nature of the task, each participant adjusted the playback volume as required. Reproduction was monaural, presented equally to both ears. While the choice between loudspeakers and headphones is often debated [15], in this case, particularly as reproduction was mono, headphones were considered to be the choice with greater potential for reproducibility.

The experimental interface was designed using Pure Data, an open source, visual programming language. The GUI used by participants is shown in Fig. 3. Each participant listens to the audio clip in full at least once, then the audio is looped while mixing takes place and fader movement is recorded. The participant then clicks 'stop mix' and the next session is loaded. For each session the user is asked to create their preferred mix by adjusting the faders.

An initial trial was provided in order for participants to become familiar with the test procedure, after which the six conditions (3 songs, 2 sources each) were presented in a randomised order. The mean test duration was 14.2 minutes, ranging from 11 to 17 minutes. The real-time audio output during mixing was recorded to .wav file at a sampling rate of 44,100Hz and a resolution of 16 bits. Fader positions were also recorded to .wav files using the same

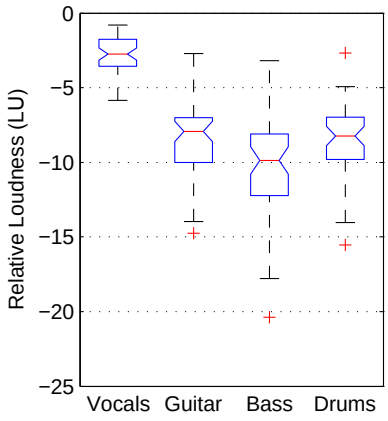


Figure 4: Normalised gain levels of each track, evaluated over all final mix positions.

format. As shown in Fig. 3, the true instrument levels were hidden from participants by displaying arbitrary fader controls. The range of the faders was limited to ± 20dB from the source, to prevent solo-ing any instrument, due to the uniqueness of the mix-space breaking down at boundaries.

5. RESULTS AND DISCUSSION

For each participant, song and source, the recorded time-series data was downsampled to an interval of 0.1 seconds, then transformed from gain to mix domains using Eqn. 1. From this data the vectors representing *mix-velocity*, described in Section 3.2.2, were obtained using Eqn. 3.

5.1 Instrument levels

Since the experiment is concerned with relative loudness levels between instruments and not the absolute gain values which were recorded, normalised gains can be calculated from Eqn. 2, with $r = 1$. When all songs, sources and participants are considered, the distribution of normalised gains at the final mix positions is shown in Fig. 4, expressed in LU. In Fig. 4 and 5 the boxplots show the median at the central position and the box covers the interquartile range. The whiskers extend to extreme points not considered outliers and outliers are marked with a cross. Two medians are significantly different at the 5% level if their notched intervals do not overlap. Fig. 4 shows good agreement with previous studies, particularly a level of $\approx -3LU$ for vocals [7, 10] and $\approx -10LU$ for bass (see Fig. 1 of [10]). Fig. 6 also shows the final positions of all mixes of each song, where mix '**1A**' is the mix produced by mixer 1, starting at source A, etc. This indicates a clustering of mixes based on the source position. Fig. 5d shows the box-plot of each ϕ value when data for all songs, sources and participants is combined. Since the audio tracks were loudness-normalised, the median value can be used to determine the preferred balance of tracks in terms of relative loudness, using Eqn 4. The results are shown in Table 1. Had the experiment been performed in a more conventional control room with studio monitors, less variance might have been observed [15].

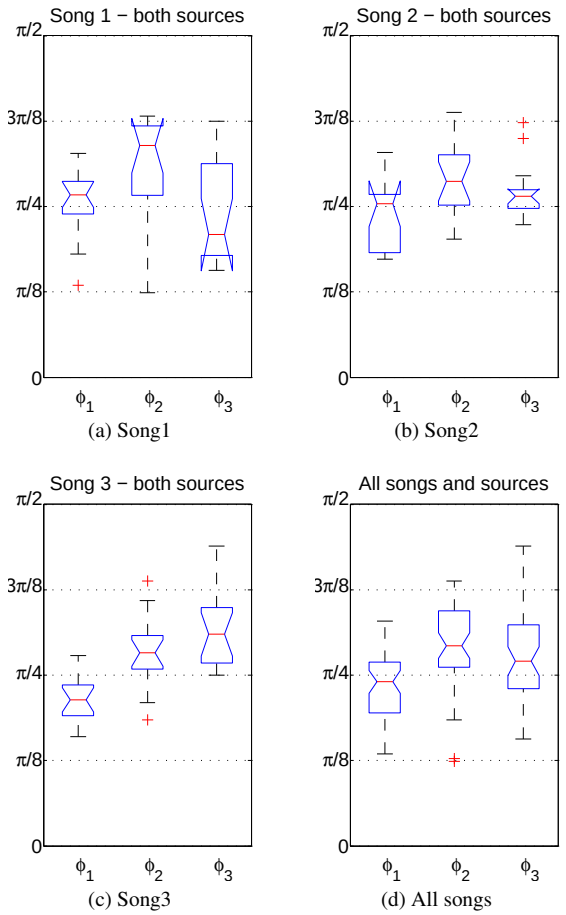


Figure 5: Boxplots showing the distribution of ϕ terms at final mix positions. While balances vary with song, vocal/backing balance and guitar/rhythm balance are more consistent than the bass/drums balance.

$$\text{vocals/backing} = 20 \times \log_{10}\left({}^{\cos(\phi_1)}/_{\sin(\phi_1)}\right) \quad (4a)$$

$$\text{guitar/rhythm} = 20 \times \log_{10}\left({}^{\cos(\phi_2)}/_{\sin(\phi_2)}\right) \quad (4b)$$

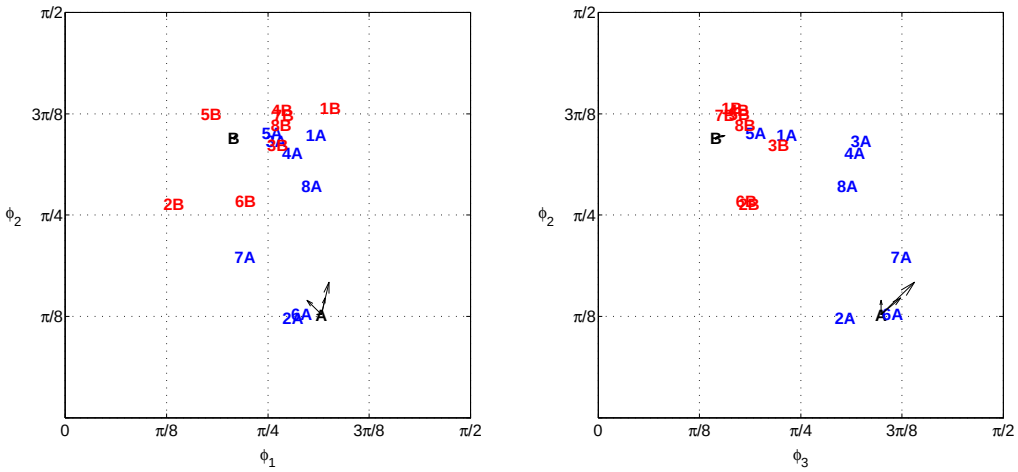
$$\text{bass/drums} = 20 \times \log_{10}\left({}^{\cos(\phi_3)}/_{\sin(\phi_3)}\right) \quad (4c)$$

Balance	Song 1	Song 2	Song 3	All
vocals/backing	-0.95	-0.23	+1.98	+0.54
guitar/rhythm	-5.15	-2.04	-1.78	-2.38
bass/drums	+2.27	-0.83	-3.35	-1.12

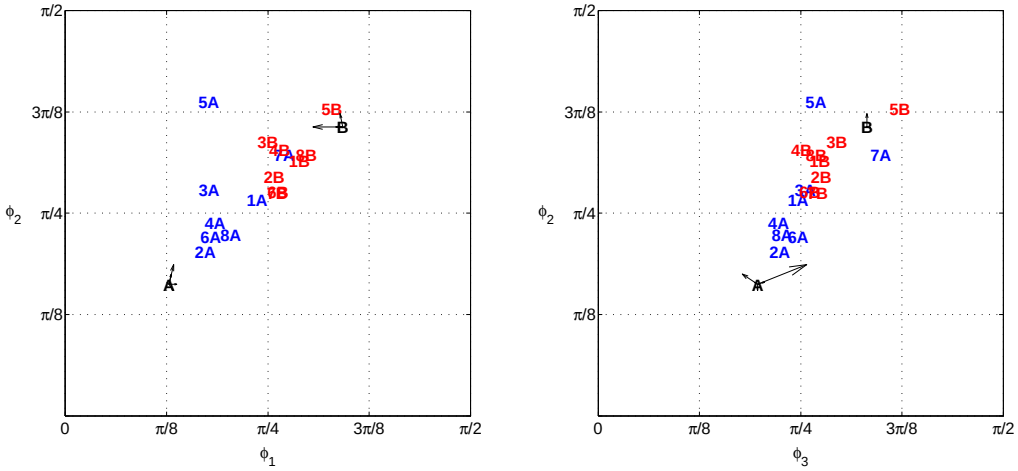
Table 1: Median level-balances (in loudness units) from Fig. 5, between sets of instruments defined by Fig. 2.

5.2 Source-directivity

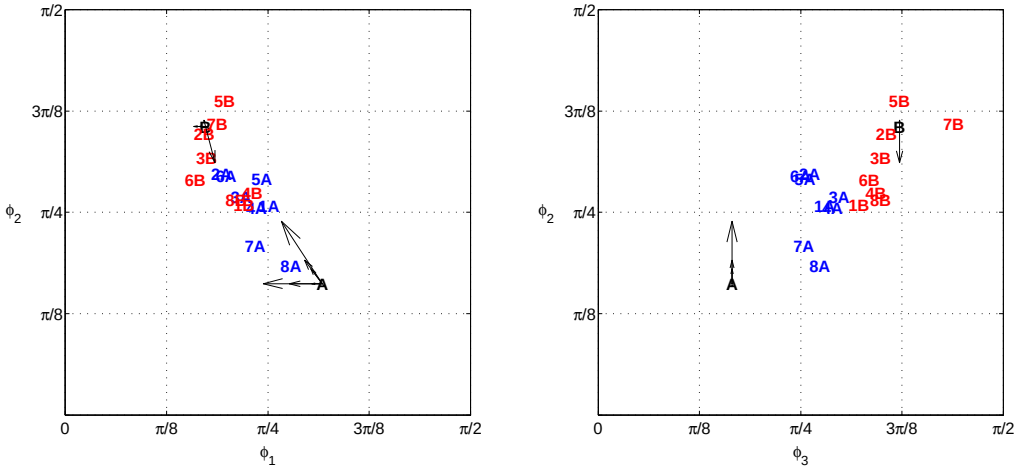
Movement away from the source is characterised by the first non-zero element of the mix-velocity triple u, v, w (see Eqn. 3). The displacement and direction of this move is used to investigate the source directivity. Fig. 6 shows



(a) Song 1 - the central cluster of mixes contains mixes originating at both sources.



(b) Song 2 - **7A** is the only mix in this study which has more nearest neighbours from the other source.



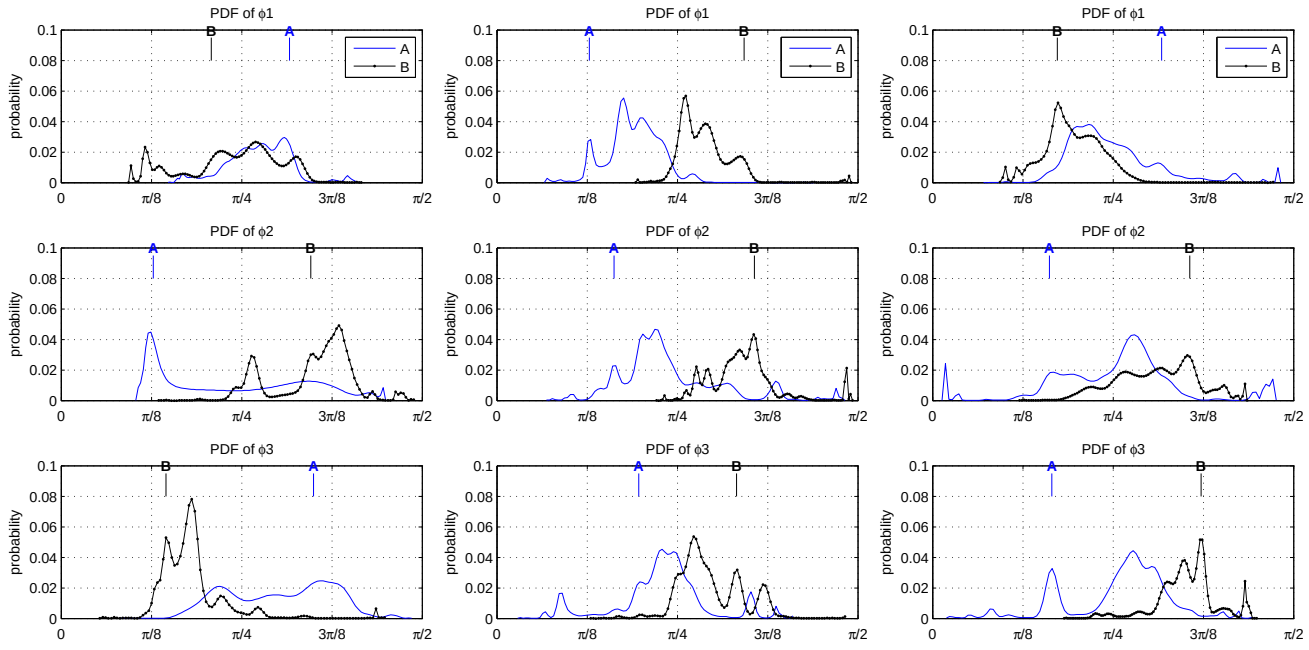
(c) Song 3 - distinct cluster of mixes formed of those which started from source A

Figure 6: Positions of sources and final mixes in the mix-space. Source-directivity is indicated by added vectors.

the source positions within the mix-space, marked '**A**' and '**B**'. The initial vectors are also shown, indicating the direction and step size of the first changes to the mix. None of the sources can be considered omnidirectional, as certain mix-decisions are more likely than others. This directivity indicates that the source position has an immediate influence on mixing decisions.

5.3 Mix-space navigation

Fig. 7 shows the probability density function (PDF) of $\phi_{n,t}$ when averaged over the eight mixers depicted in Fig. 6. The function is estimated using Kernel Density Estimation, using 100 points between the lower and upper bounds of each variable. This plot displays the mix configurations



(a) Song1 (b) Song2 (c) Song3

Figure 7: Estimated probability density functions of ϕ terms, for each of the three songs, averaged over all mixers. Sources positions are highlighted with **A** and **B**. As the functions often differ it can be seen that exploration of the mix-space is dependant on initial conditions.

which the participants spent most time listening to and it is seen that all distributions are multi-modal. There are peaks close to the initial positions, the final positions and other interim positions that were evaluated during the mixing process. There are a number of different approaches to multitrack mixing of pop and rock music, one of which is to start with one instrument (such as drums or vocals) and build the mix around this by introducing additional elements. Some participants were observed mixing in this fashion, shown in Fig. 7, where peaks at extreme values of ϕ_n show that instruments were attenuated as much as the constraints of the experiment would allow.

For Song 1, ϕ_1 is well balanced and centered close to $\pi/4$. This indicates that mixers tended to listen in states where the relative loudness of the vocal and backing track were similar. A similar pattern is observed for Song 2, where ϕ_3, shows that the level of drum and bass tend to be adjusted such that the tracks have similar loudness (Table 1 shows the median loudness difference within final mixes was $<$1dB). The distributions of ϕ_2 indicates that the guitar was often set to be of lower loudness than the rhythm section, as also shown in Table 1.

There are notable differences due to the source. The distributions for Song 2 suggest that exploration depended on the initial source configuration, with Source A leading to louder vocals and louder guitar than Source B. However, for Song 2, the distributions of ϕ terms are similar for both source positions, simply offset. This suggests that, while different regions of the mix-space were explored, they were explored in a similar fashion.

Overall, for Song 3, the distributions in Fig. 7, the median balances in Fig. 5c and the clustering of final positions shown in Fig. 6c indicate that mixers were more consistent with this song than others. This may be due to the tracks representing processed stems of a full mix, where the inter-channel balances in these stems, subject to dynamic range compression as well as the relative level of reverberation and other effects, may have provided clues as to how the groups were balanced in that final mix from which stems were obtained. This further suggests that the more prior work that has been put into the mix, the less likely subsequent mixers are to explore the entire mix-space.

Since this experiment gathered data for only three songs, the results should be considered as specific rather than general. It is not known at this time how many songs would need to be studied to be able to generalise to mixing as a whole, however, these three songs are considered to be typical, due to their conventional instrumentation.

5.4 Application of results

In automatic fader control, rather than aiming for equal loudness across all instruments, the preferred balances between semantic pairings of instruments, shown in Fig. 5d, could be used as the target for optimisation. This would require the unsupervised clustering of audio tracks into semantically-linked instrument groups, a task which is currently an active area of research [16–18].

Intelligent mixing systems aim to generate audio mixtures based on some desired criteria, ideally '*Quality*'. With a defined *mix-space* it is possible to utilise a number of dynamic techniques in generating mixes. The results of the

experiment outlined in this paper could be used to train an intelligent mixing system to produce a number of alternate mixes which the user could select from, in order to further train the system. Further information regarding mixing style can be found from the data. For example, the probability density function of *mix-velocity* could differentiate between mixers who mixed using either careful adjustment of the faders towards a clear goal or by alternating large displacements with fine-tuning. Knowing the distribution of step size used by human mixers will aid optimisation of search strategies in intelligent mixing systems.

6. CONCLUSIONS

For a level-balancing task, a mix-space has been defined using the gains of each track. A number of features of the space have been presented and an experiment was performed in order to investigate how mix engineers explore this space for a four track mixture of modern popular music.

From these early results it has been observed that each source has a directivity that is not equal in all directions, i.e. that not all possible first decisions in the mix process are equally likely. For each song there are varying degrees of clustering of final mixes and it is seen that the final mix is dependant on the initial conditions. The exploration of the space is also dependant on the initial conditions. This experiment has indicated a certain level of agreement between participants regarding the ideal balances between groups of instruments, although this varies according to the song in question.

Ultimately, the theory presented here could be expanded to include other mix parameters. Since panning, equalisation and dynamic range compression/expansion are each an extension to the track gain (either channel-dependant, frequency-dependant or signal-dependant), it should be possible to add these parameters to the existing framework.

7. REFERENCES

[1] M. Terrell, A. Simpson, and M. Sandler, "The Mathematics of Mixing," *Journal of the Audio Engineering Society*, vol. 62, no. 1, 2014.

[2] "ISO 9000:2005 Quality management systems – Fundamentals and vocabulary," 2009, http://www.iso.org/iso/catalogue_detail?csnumber=42180.

[3] R. King, B. Leonard, and G. Sikora, "Consistency of balance preferences in three musical genres," in *Audio Engineering Society Convention 133*, San Francisco, USA, October 2012.

[4] ——, "Variance in level preference of balance engineers: A study of mixing preference and variance over time," in *Audio Engineering Society Convention 129*. San Francisco, USA: Audio Engineering Society, Nov 2010.

[5] 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. WASPAA'09.* IEEE, 2009, pp. 1–4.

[6] S. Mansbridge, S. Finn, and J. D. Reiss, "Implementation and evaluation of autonomous multi-track fader control," in *Audio Engineering Society Convention 132*, Budapest, Hungary, April 2012.

[7] P. Pestana and J. D. Reiss, "Intelligent Audio Production Strategies Informed by Best Practices," in *AES 53rd International Conference: Semantic Audio*, London, UK, January 2014, pp. 1–9.

[8] J. Reiss and B. De Man, "A semantic approach to autonomous mixing," *Journal on the Art of Record Production*, vol. Issue 8, Dec. 2013.

[9] E. Deruty, F. Pachet, and P. Roy, "Human-Made Rock Mixes Feature Tight Relations Between Spectrum and Loudness," *Journal of the Audio Engineering Society*, vol. 62, no. 10, pp. 643–653, 2014.

[10] B. De Man, B. Leonard, R. King, and J. Reiss, "An analysis and evaluation of audio features for multitrack music mixtures," in *ISMIR*, Taipei, Taiwan, October 2014, pp. 137–142.

[11] M. Cartwright, B. Pardo, and J. Reiss, "Mixploration: rethinking the audio mixer interface," in *International Conference on Intelligent User Interfaces*, Haifa, Israel, February 2014.

[12] A. Wilson and B. Fazenda, "Perception & evaluation of audio quality in music production," in *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland, 2013, pp. 1–6.

[13] ——, "Characterisation of distortion profiles in relation to audio quality," in *Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx-14)*, Erlangen, Germany, 2014, pp. 1–6.

[14] P. D. Pestana, J. D. Reiss, and A. Barbosa, "Loudness measurement of multitrack audio content using modifications of itu-r bs. 1770," in *Audio Engineering Society Convention 134*, Rome, Italy, May 2013.

[15] R. L. King, B. Leonard, and G. Sikora, "Loudspeakers and headphones: The effects of playback systems on listening test subjects," in *Proc. of the 2013 Int. Congress on Acoustics*, Montréal, Canada, June 2013.

[16] S. Essid, G. Richard, and B. David, "Musical instrument recognition by pairwise classification strategies," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 1401–1412, 2006.

[17] V. Arora and L. Behera, "Musical source clustering and identification in polyphonic audio," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 6, pp. 1003–1012, Jun. 2014.

[18] J. Scott and Y. E. Kim, "Instrument identification informed multi-track mixing." in *ISMIR*, Curitiba, Brazil, October 2013, pp. 305–310.

Smooth Granular Sound Texture Synthesis by Control of Timbral Similarity

Diemo Schwarz, Sean O'Leary
Ircam–CNRS–UPMC
firstname.secondname@ircam.fr

ABSTRACT

Granular methods to synthesise environmental sound textures (e.g. rain, wind, fire, traffic, crowds) preserve the richness and nuances of actual recordings, but need a preselection of timbrally stable source excerpts to avoid unnaturally-sounding jumps in sound character. To overcome this limitation, we add a description of the timbral content of each sound grain to choose successive grains from similar regions of the timbre space. We define two different timbre similarity measures, one based on perceptual sound descriptors, and one based on MFCCs. A listening test compared these two distances to an unconstrained random grain choice as baseline and showed that the descriptor-based distance was rated as most natural, the MFCC based distance generally as less natural, and the random selection always worst.

1. INTRODUCTION

The synthesis of credible environmental sound textures such as wind, rain, fire, crowds, traffic noise, is a crucial component for many applications in computer games, installations, audiovisual production, cinema. Often, sound textures are part of the soundscape of a long scene, and in interactive applications such as games and installations, the length of the scene is not determined in advance. Therefore, it is advantageous to be able to play a given texture for an arbitrary amount of time, but simple looping would introduce repetition that is easy to pick out. Using very long loops, or layering several loops can avoid this problem (and is the way sound designers currently do this), but this stipulates that a long enough recording of a stable environmental texture is available, and uses up a lot of media and memory space.

We present here a method to extend an environmental sound texture recording for an arbitrary amount of time, without the need for the source recording to be of a stable and uniform timbre or density. This means, a sound designer can use a recording that fits the scene in atmosphere, but without needing to isolate a stable and sufficiently long loop, since our method will ensure smooth timbral transitions, while still varying the texture to avoid repetition effects.

Copyright: ©2015 Diemo Schwarz 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.

Our method is based on randomised granular playback with control of the similarity between grains using two different timbral distance measure that are compared in an evaluation: a timbral distance based on audio descriptors, and an MFCC-based distance. We also compare to purely randomised playback as a baseline.

2. PREVIOUS AND RELATED WORK

The method presented here situates itself in the granular synthesis-based approaches to sound textures, as opposed to ones based on signal or physical models. These methods need a recording as source material from which sound grains are picked and played back. Granular playback takes advantage of the richness of actual recorded sound, in contrast to other methods based on pure synthesis [4], see the state-of-the-art overview on sound texture synthesis [19] for further discussion and a general introduction of sound textures. Fröjd and Horner [5] use purely randomised playback of long grains (around one second), with half-grain crossfade, and slight randomisation of playback parameters (detuning, amplification) to avoid repetition. O'Leary and Roebel's *Montage approach* [14, 15] exchanges grains by timbral similarity to avoid repetition, while following template sequences from the original, and introduce a spectral crossfade minimising phase distortion.

Specifically, the present research draws on previous work on corpus-based sound texture synthesis [20, 21], that can also be seen as content-aware granular synthesis, and extends the work of Fröjd and Horner [5] by the explicit modeling and control of timbral similarity on randomised granular playback. Other methods to extend a given texture are based on modeling of higher-order statistical properties [1, 9, 11, 12]. All these latter methods need a source recording with stable and uniform texture content while our proposed method can work with more varied textures by being aware of the timbral content of all grains.

Other methods for sound textures go further by modeling and recreating the typical transitions occuring in the source texture by wavelet- or Markov-trees [3, 7, 8].

A recent approach by Heittola et al. [6] quite similar to ours is aimed at full soundscape synthesis to recreate the acoustic environment of a specific location for digital maps. There, the timbral similarity is calculated on MFCCs and their deltas averaged over four second grains. The resulting similarity matrix serves to coalesce adjacent grains into longer segments, and to cluster these in order to control the smoothness of transitions.

3. TEXTURE SYNTHESIS

Our method is derived from corpus-based concatenative synthesis (CBCS) [18], where grains are played back from a corpus of segmented and descriptor-annotated sounds. Usually, CBCS is used to control the timbral evolution of the synthesised sound while still using original recordings as the sound source. This can be applied to texture synthesis to match the sound to the evolution of a given scene [20], see also the example video of interactive wind texture synthesis in a 2D descriptor space [1], when the descriptor target is given directly by the sound designer, or by the game engine. However, in the application described here, we don't want to control the timbral output directly, but have the system synthesise a varying texture without audible repetions nor artefacts such as abrupt timbral or loudness changes. To this end, we use a timbral distance measure d between the last played grain and all other grains as candidates, and randomly select a successor grain from the timbrally closest grains, thus generating a random walk through the timbral space of the recording, that never takes too far a step, but that potentially still traverses the whole space of expression of the recording.

The algorithm proceeds as follows:

1. We construct a corpus of one or more recordings, segment it into grains (here of length 800 ms without overlap), and analyse each grain i for its timbral characteristics in a feature vector u_i.

   In our experiments we used two variants of annotation giving rise to two different distance measures:

   (a) An analysis of the 7 audio descriptors validated by [20], extracted with the IRCAMDESCRIPTOR library [16]: The mean of the instantaneous descriptors *Loudness, FundamentalFrequency, Noisiness, SpectralCentroid, SpectralSpread, SpectralSlope* over all frames of size 23 ms.

   (b) An analysis of the timbral shape in terms of the mean of the mel-frequency cepstral coefficients (MFCCs) over the segment.

2. For synthesis, we start with a seed grain q, selected randomly or given manually to start off with a certain timbral content.

3. When a grain is triggered, $c = 5$ successor grains are searched by a $(c+1)$-nearest neighbour search, i.e., given the current grain's descriptor values u_q as query point, the kD-tree [2, 22] finds in logarithmic time the c candidate grains with descriptor values closest to the query (and the query grain q itself, since it has a distance of zero). The distance function is a weighted Euclidean distance, with weights given by the inverse standard deviation to normalise the search space. Multiplying the weights allows us further to give more importance to certain descriptors, or to exclude them from influencing the search.

4. The succesor grain s is chosen randomly from the c candidate grains. If s is within one second of q, a new grain s is picked from the candidates, to avoid picking grains too close to each other.

5. To avoid too regular triggering of new grains, the duration and time of the next grain are randomly drawn within a 600–1000 ms range, and a random start offset of +/- 200 ms is applied to each grain.

6. Played grains are overlapped by 200 ms, and an equal-power sinusoidal cross-fade is applied during the overlap.

7. While the desired length of the output texture is not reached, the chosen grain s becomes the query grain q, and the algorithm continues at step 3.

3.1 Implementation

The prototype system is implemented in Max/MSP using the MuBu (Multi-Buffer) extension library [17], with the integration of the batch analysis module pipo.ircamdescriptor [2].

4. RESULTS AND EVALUATION

The method presented here is evaluated in an ongoing listening test accessible online [3]. At the time of writing, 31 subjects took the test.

The test consists of a questionnaire with 7 second extracts of 7 sound examples listed in table 1. This small test database contains sounds from [3] that are widely used in evaluation of sound textures [5, 10] and thus partially allows comparison of the results. Other sounds were contributed by [20] and by the partners of the PHYSIS project [4]. All sounds were chosen for their properties of being a non-uniform environmental sound texture, i.e. containing some variation in texture and timbre, but not clearly distinguishable short sound events. An exception is the Baby Crying sound, that is here as an extreme counterexample, since it contains very different and well-separated cries.

Sound Example	Description
Lapping Waves	long-term structure
Desert Wind	wind with occasional gusts
Stadium Crowd	atmosphere, occasional cheering and honking
Water Faucet	various speeds of water flow
Formula One	not actually a texture, containing structured variation
Traffic Jam	motor sounds, honking, some shouts
Baby Crying	not actually a texture, containing large variation

Table 1. List of Test Sound Database and description

[1] http://imtr.ircam.fr/imtr/Sound_Texture_Synthesis

[2] http://forumnet.ircam.fr/product/max-sound-box

[3] http://ismm.ircam.fr/sound-texture-transition-control-evaluation

[4] http://sites.google.com/site/physisproject

	orig	**descr**	**mfcc**	**random**
Lapping Waves	85.09 (± 20.70)	73.04 (± 19.76)	71.82 (± 23.58)	46.01 (± 25.16)
Desert Wind	92.46 (± 08.70)	59.90 (± 22.28)	61.97 (± 26.19)	23.24 (± 23.11)
Stadium Crowd	91.65 (± 13.36)	56.22 (± 29.05)	23.03 (± 18.52)	25.83 (± 20.18)
Water Faucet	86.82 (± 16.93)	55.38 (± 24.01)	25.34 (± 18.11)	14.18 (± 15.11)
Formula One	95.15 (± 08.57)	29.55 (± 20.62)	17.43 (± 19.61)	12.85 (± 15.71)
Traffic Jam	77.36 (± 26.44)	59.01 (± 31.47)	56.23 (± 29.07)	52.97 (± 27.07)
Baby	95.43 (± 09.45)	17.98 (± 15.15)	13.07 (± 15.38)	15.89 (± 21.02)
Total	89.14 (± 17.30)	50.16 (± 29.76)	38.41 (± 31.37)	27.28 (± 26.15)

Table 3. Naturalness rating mean and standard deviation over all subjects.

For each example, the original, and 4 test stimuli of 7 s length are presented. The stimuli contain in randomised order the 3 syntheses (by descriptor distance, MFCC distance, random), and the original as hidden reference. For each stimulus, the subject is asked to rate the aspect of *Naturalness* on a scale of 0–100, with labels given in table 2. Note that the question of *Sound Quality* does not make sense for this evaluation since no signal processing other than long cross-fades is applied, and therefore the percieved sound quality is the same for all stimuli.

We linearly scaled the collected naturalness ratings individually for each subject (over all sounds) to a range of 0 to 100. The rationale is that the relative ratings of overly enthousiastic or overly critical subjects are thus made comparable with the rest of the subjects. We can see in figures 3 and 4 that only a few subjects (notably 1, 12, 14, 27) exhibit very narrow rating ranges.

The collected data is summarised in figure 1. We can see that the descriptor-based similarity measure generally obtains better ratings than the MFCC based one, that the random grain choice is rated worst, and that the originals are rated very high, with only a few outliers.

To test if the observed differences of means are significant or simply due to chance, further statistical analysis has been carried out using the ANOVA (analysis of variance) method with Bonferroni correction. Here the null hypothesis H_0 is that means are equal, and differences are due to chance, and the alternative hypothesis H_A is that the means are not equal. The p-values and significance levels [5] for each pair of comparisons are given in tables 4 and 5 for the raw and scaled ratings, respectively. The scaling seems to augment the contrast of the results, leading to a rise in significance level for a few pairs in the ANOVA results.

ANOVA confirms that globally the descriptor-based similarity is preferred over MFCC, and both are preferred over the random method. However, the detailed analysis shows

Score	**Label**
0-19	Very unnatural: repetitions, jumps, cuts.
20-39	Somewhat unnatural
40-59	Somewhat natural
60-79	Very natural
80-100	As natural as original

Table 2. Naturalness rating scale

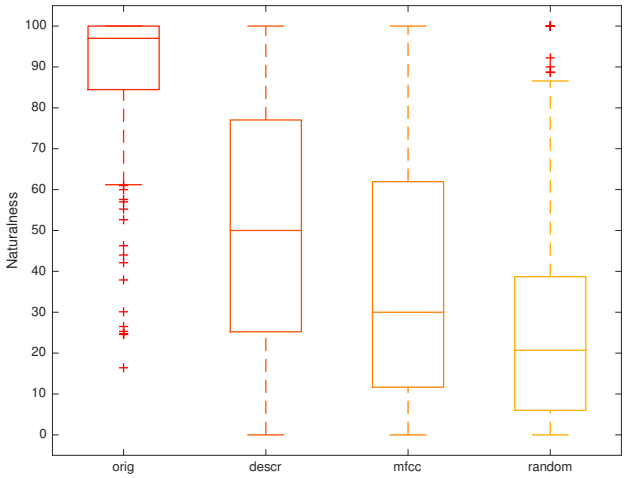


Figure 1. Box plot of the scaled naturalness ratings per type of stimulus, showing the median (middle line), quartile range (box), min/max (whiskers), and outliers (crosses).

that only for *Stadium Crowd* and *Water Faucet*, and to a lesser degree for *Formula One*, descriptor and MFCC-based distance are rated significantly different. We hypothesize that especially these sounds benefit greatly from the more detailed descriptors *Loudness* and *FundamentalFrequency*, as they contain sequences of pitched foreground events. For all sounds but *Traffic Jam* and *Baby Crying* the descriptor-based distance is significantly rated better than the random method, while for the MFCC-based distance this is only so for *Lapping Waves* and *Desert Wind*. Another remark is that for *Lapping Waves* and *Traffic Jam* the original is not rated significantly different from the descriptor-based method, and for the former sound this also applies to the MFCC-based method.

5. CONCLUSIONS AND FUTURE WORK

A possible explanation for the general superiority of the descriptor-based similarity measure over the MFCC based one is that the perceptual descriptors better capture certain aspects of the sound character of environmental textures beyond pure spectral shape (that is represented by MFCCs). We can hypothesise that some of this information is related to pitch content, as expressed by the *FundamentalFrequency* and *Noisiness* descriptors. More re-

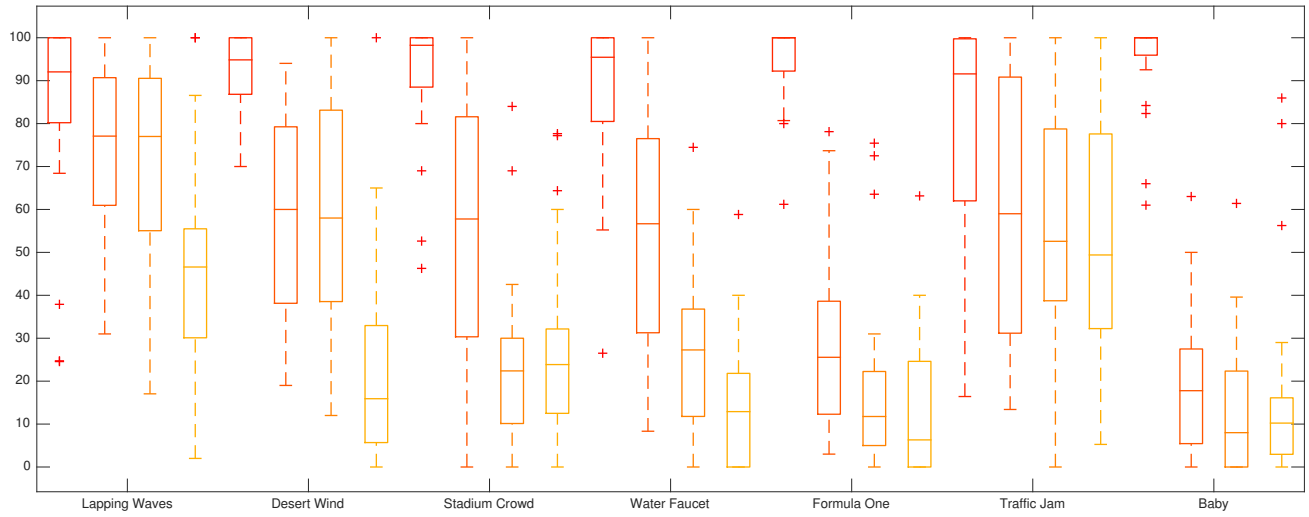


Figure 2. Box plot of the scaled naturalness ratings per source sound and type of stimulus (orig, descr, mfcc, random).

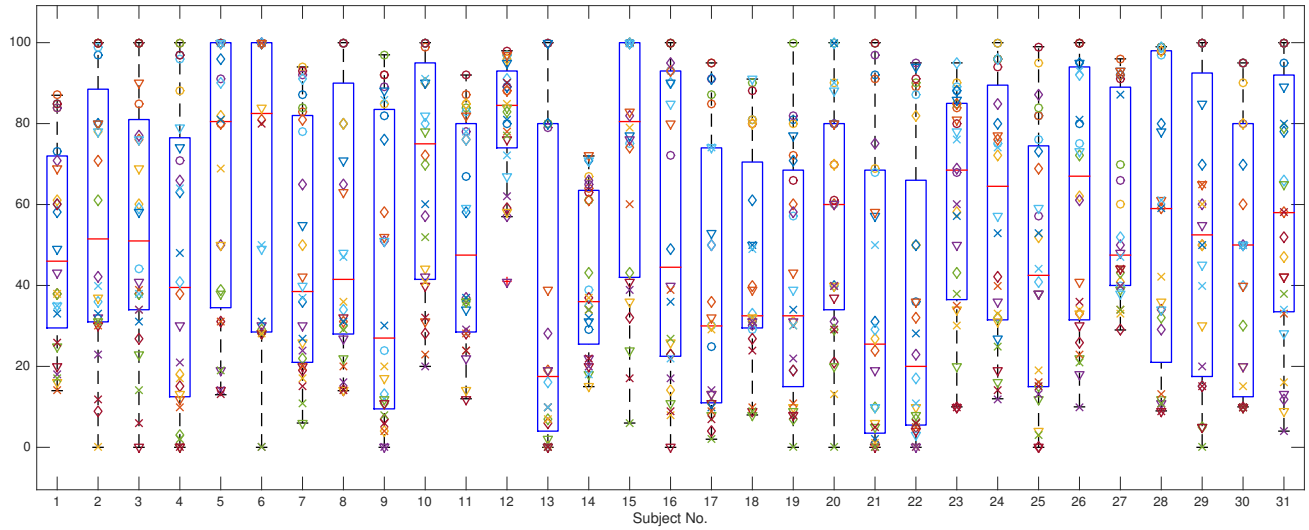


Figure 3. Box and dot plot of the per-subject naturalness rating prior to scaling (○ original, ◇ descr, ▽ mfcc, × random).

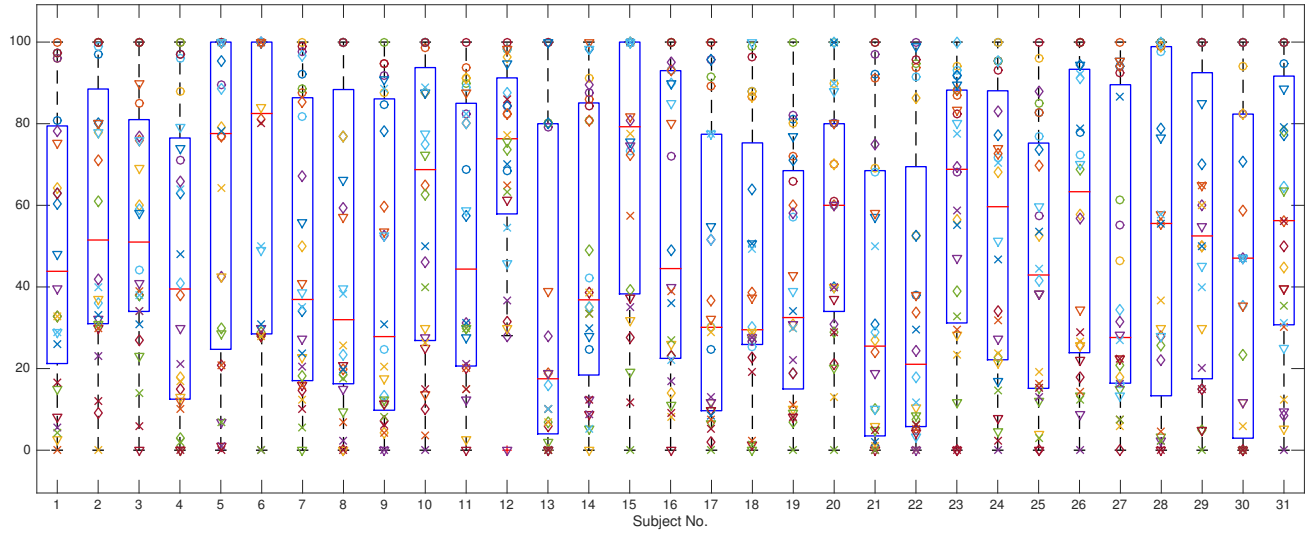


Figure 4. Box and dot plot of the per-subject naturalness rating after scaling (○ original, ◇ descr, ▽ mfcc, × random).

	orig descr	**orig mfcc**	**orig random**	**descr mfcc**	**descr random**	**mfcc random**
Lapping Waves	0.1772	0.2037	0.0000 ****	1.0000	0.0003 ***	0.0002 ***
Desert Wind	0.0000 ****	0.0000 ****	0.0000 ****	1.0000	0.0000 ****	0.0000 ****
Stadium Crowd	0.0000 ****	0.0000 ****	0.0000 ****	0.0000 ****	0.0000 ****	1.0000
Water Faucet	0.0000 ****	0.0000 ****	0.0000 ****	0.0000 ****	0.0000 ****	0.1179
Formula One	0.0000 ****	0.0000 ****	0.0000 ****	0.1576	0.0106 *	1.0000
Traffic Jam	0.0833	0.0338 *	0.0115 *	1.0000	1.0000	1.0000
Baby	0.0000 ****	0.0000 ****	0.0000 ****	1.0000	1.0000	1.0000
total	0.0000 ****	0.0000 ****	0.0000 ****	0.0002 ***	0.0000 ****	0.0002 ***

Table 4. P-values and significance class [5] for each pair of differences of means on unscaled naturalness ratings.

	orig descr	**orig mfcc**	**orig random**	**descr mfcc**	**descr random**	**mfcc random**
Lapping Waves	0.2360	0.1409	0.0000 ****	1.0000	0.0000 ****	0.0001 ***
Desert Wind	0.0000 ****	0.0000 ****	0.0000 ****	1.0000	0.0000 ****	0.0000 ****
Stadium Crowd	0.0000 ****	0.0000 ****	0.0000 ****	0.0000 ****	0.0000 ****	1.0000
Water Faucet	0.0000 ****	0.0000 ****	0.0000 ****	0.0000 ****	0.0000 ****	0.1408
Formula One	0.0000 ****	0.0000 ****	0.0000 ****	0.0365 *	0.0012 **	1.0000
Traffic Jam	0.0858	0.0297 *	0.0075 **	1.0000	1.0000	1.0000
Baby	0.0000 ****	0.0000 ****	0.0000 ****	1.0000	1.0000	1.0000
total	0.0000 ****	0.0000 ****	0.0000 ****	0.0000 ****	0.0000 ****	0.0001 ****

Table 5. P-values and significance class [5] for each pair of differences of means on scaled naturalness ratings.

search is necessary to test this hypothesis and to link it with recent findings about fundamental mechanisms of sound texture perception [13].

While the presented method is not a sequence model that tries to model and generate the temporality of variation given an environmental recording, it can regenerate at least some of the naturally occuring variation in texture recordings. This has the two advantages of having a more varied output for background atmosphere sounds that uses the whole range of sound occoring in a source recording, and that the sound designer does not have to limit herself to stable textural recordings, or has to hunt down long-enough stretches of stable texture in longer recordings.

Although this is not the topic of this article, synthesis can be started off at specific-sounding grains in the recording as seeds, in order to start the texture with a given atmosphere (e.g. start with calm wind to not startle the listener at the beginning of a new scene with a gust of wind). This could, for instance, be achieved using a scatterplot interface that allows to visualise the timbral space in 2D as popularised by the CataRT software [6]. With a little future work, the texture could then be made to move towards another type of sound by specifying its feature vector and favouring transitions that move towards that point in the descriptor space. More future work should check the influence and possible automatic estimation of the neighbourhood parameter (the number of candidates c).

To conclude, we hope that this method can improve the workflow of sound designers for interactive or post-production applications, and further augment the advantages that procedural audio has to offer over fixed media in order to foster uptake by the industry.

[5] The significance level depending on the p-value is habitually represented by a number of stars as follows:

Level	*	**	***	****
$p \leq$	0.05	0.01	0.001	0.0001

[6] http://ismm.ircam.fr/catart

Acknowledgments

The work presented here is partially funded by the French *Agence Nationale de la Recherche* (ANR) within the project *PHYSIS*, ANR-12-CORD-0006. The authors wish to thank Wei-Hsiang Liao for his groundwork on the online evaluation questionnaire, Axel Röbel, the *PHYSIS* project partners, and the Analysis–Synthesis and ISMM teams at Ircam.

References

[1] Joan Bruna and Stéphane Mallat. Audio Texture Synthesis with Scattering Moments. page 5, November 2013. URL http://arxiv.org/abs/1311.0407.

[2] Wim D'haes, Dirk van Dyck, and Xavier Rodet. PCA-based branch and bound search algorithms for computing K nearest neighbors. *Pattern Recognition Letters*, 24(9–10):1437–1451, 2003.

[3] Shlomo Dubnov, Ziz Bar-Joseph, Ran El-Yaniv, Danny Lischinski, and Michael Werman. Synthesis

of audio sound textures by learning and resampling of wavelet trees. *IEEE Computer Graphics and Applications*, 22(4):38–48, 2002.

[4] Andy Farnell. *Designing Sound*. MIT Press, October 2010. ISBN 9780262014410. URL http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=12282.

[5] M. Fröjd and A. Horner. Sound texture synthesis using an overlap-add/granular synthesis approach. *Journal of the Audio Engineering Society*, 57 (1/2):29–37, 2009. URL http://www.aes.org/e-lib/browse.cfm?elib=14805.

[6] Toni Heittola, Annamaria Mesaros, Dani Korpi, Antti Eronen, and Tuomas Virtanen. Method for creating location-specific audio textures. *EURASIP Journal on Audio, Speech and Music Processing*, 2014.

[7] Stefan Kersten and Hendrik Purwins. Sound texture synthesis with hidden markov tree models in the wavelet domain. In *Proceedings of the International Conference on Sound and Music Computing (SMC)*, Barcelona, Spain, July 2010.

[8] Anil Kokaram and Deirdre O'Regan. Wavelet based high resolution sound texture synthesis. In *Proceedings of the Audio Engineering Society Conference*, 6 2007. URL http://www.aes.org/e-lib/browse.cfm?elib=13952.

[9] Wei-Hsiang Liao, Axel Roebel, and Wen-Yu Su. On the modeling of sound textures based on the STFT representation. In *16th Int. Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland, Septembre 2013. URL http://architexte.ircam.fr/textes/Liao13a/.

[10] L. Lu, L. Wenyin, and H.J. Zhang. Audio textures: Theory and applications. *IEEE Transactions on Speech and Audio Processing*, 12(2):156–167, 2004. ISSN 1063-6676. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1284343.

[11] J.H. McDermott, A.J. Oxenham, and E.P. Simoncelli. Sound texture synthesis via filter statistics. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, October 18–21 2009.

[12] Josh H McDermott and Eero P Simoncelli. Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis. *Neuron*, 71(5):926–40, September 2011. ISSN 1097-4199. doi: 10.1016/j.neuron.2011.06.032. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4143345&tool=pmcentrez&rendertype=abstract.

[13] Josh H McDermott, Michael Schemitsch, and Eero P Simoncelli. Summary statistics in auditory perception. *Nature neuroscience*, 16(4):493–8, April 2013. ISSN 1546-1726. doi: 10.1038/nn.3347. URL http://www.ncbi.nlm.nih.gov/pubmed/23434915.

[14] Sean O'Leary and Axel Roebel. A two level montage approach to sound texture synthesis with treatment of unique events. In *DAFx*, Germany, Septembre 2014. URL http://architexte.ircam.fr/textes/OLeary14b/.

[15] Sean O'Leary and Axel Roebel. A montage approach to sound texture synthesis. In *EUSIPCO*, Lisabon, Portugal, Septembre 2014. URL http://architexte.ircam.fr/textes/OLeary14a/.

[16] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the Cuidado project. Technical Report version 1.0, Ircam – Centre Pompidou, Paris, France, April 2004. URL http://www.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf.

[17] Norbert Schnell, Axel Röbel, Diemo Schwarz, Geoffroy Peeters, and Ricardo Borghesi. MuBu & friends – assembling tools for content based real-time interactive audio processing in Max/MSP. In *Proceedings of the International Computer Music Conference (ICMC)*, Montreal, Canada, August 2009.

[18] Diemo Schwarz. Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine*, 24(2):92–104, March 2007. Special Section: Signal Processing for Sound Synthesis.

[19] Diemo Schwarz. State of the art in sound texture synthesis. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)*, Paris, France, September 2011.

[20] Diemo Schwarz and Baptiste Caramiaux. *Interactive Sound Texture Synthesis through Semi-Automatic User Annotations*. Lecture Notes in Computer Science. Springer International Publishing, 2014. doi: 10.1007/978-3-319-12976-1-23.

[21] Diemo Schwarz and Norbert Schnell. Descriptor-based sound texture sampling. In *Proceedings of the International Conference on Sound and Music Computing (SMC)*, pages 510–515, Barcelona, Spain, July 2010.

[22] Diemo Schwarz, Norbert Schnell, and Sebastien Gulluni. Scalability in content-based navigation of sound databases. In *Proceedings of the International Computer Music Conference (ICMC)*, Montreal, QC, Canada, August 2009.

EMBODIED AUDITORY DISPLAY AFFORDANCES

Stephen Roddy
Trinity College Dublin, Dublin 2, Ireland
roddyst@tcd.ie

Dermot Furlong
Trinity College Dublin, Dublin 2, Ireland
dfurlong@tcd.ie

ABSTRACT

The current paper takes a critical look at the current state of Auditory Display. It isolates naive realism and cognitivist thinking as limiting factors to the development of the field. An extension of Gibson's theory of affordances into the territory of Embodied Cognition is suggested. The proposed extension relies heavily on Conceptual Metaphor Theory and Embodied Schemata. This is hoped to provide a framework in which to address the problematic areas of theory, meaning and lack of cognitive research in Auditory Display. Finally the current research's development of a set of embodied auditory models intended to offer greater lucidity and reasonability in Auditory Display systems through the exploitation of embodied affordances, is discussed.

1. CURRENT STATE OF FIELD

The field of Auditory Display (AD) is in crisis. After a strong and promising start with the establishment of the International Community for Auditory Display and many years of groundbreaking research, the program struggles for momentum. Data sonification has not reached the same level of innovation or mainstream acceptance, as visualization. This is often dismissed as a symptom of a visually biased Western culture. The AD corpus is littered with open questions and dead ends that require innovative solutions if this discipline is to continue to evolve past its current state. Walker and Nees [1] call for an all-encompassing theoretical framework in which to position AD research. Neuhoff and Heller [2] suggest that future research needs to leverage intuitive mental models in the design of AD technologies. Gossman [3] suggests an embodied cognition approach towards AD design and Walker and Kramer [4] propose focusing on general cognitive processing as a key concern in the development of the area. These examples are reflective of the general thinking across the community on the future of AD. This trend tends to reference the need for a theoretical framework more cognitively based research and a deeper understanding of the place of meaning in AD technologies.

2. AUDITORY DISPLAY THEORY

The status of AD as a collaborative research program at the intersection of science, technology, cognitive science and the arts is often considered a disadvantage across the literature [1]. This perception drives the search for some solid theoretical foundations upon which to lie AD. There are general theoretical arguments to be made against this approach. Gardner [5] points out how valuable and positive inter-disciplinary collaboration has been to the establishment and development of cognitive science. Modern cognitive science exists at a junction between Psychology, Philosophy, Artificial Intelligence, Linguistics, Anthropology and Neuroscience. It has been this mixture of different kinds of theory and practice that have led to the success of cognitive science as a research program. To date AD has benefited from contributions across a large spectrum of research areas, to move away from this knowledge sharing community would be a mistake. The general spirit of foundationalism (whereby one endeavours towards a single unified theory) in science, academics and the arts has been heavily critiqued by thinkers such as Dewey, Rorty, and Popper [6, 7, 8]. They have shown that the pragmatic interrelation of diverse and sometimes opposed theories presents a more useful context in which to pursue scientific, academic and artistic goals. Severing AD from its rich and innovative background context necessarily reduces the potential for a cross-pollination of new ideas. This limits the relevancy of AD to outside research fields and erodes the potential for benefit from future developments within these fields. This may not be the best approach for overcoming the current stagnation and could have dire consequences for the innovation and development of the discipline in the future. In light of these considerations, the march towards a monolithic theory of AD seems misgiven. Rather multiple theories of AD, both competitive and complementary, should be encouraged. This in turn may give rise to a form of Adhocracy whereby AD theories can be pragmatically applied to solve a specific problem.

3. TOWARDS A PRAGMATIC DESIGN THEORY

As previously discussed AD exists in a state of flux at the intersection of myriad other complimentary research fields including but not limited to Cognitive Science, Computer Science, Music, HCI, Sound Design, and Psychology. The fractured and dynamic nature of the research program is simultaneously its strength and its weakness. Eldridge [9], while recognizing the need for theoretical underpinnings for AD has suggested the development of generic principles of AD design informed by perceptual and psychoacoustic research rather than the development of a single unified theory. It is conjectured

Copyright: © 2015 Stephen Roddy et al. 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.

here that such an approach is valid in attempting to address stagnation within the field. Where the possibility of a fundamental theory is a meta-issue that applies to the overall state of AD, the most pressing internal issue within AD is the question of meaning. The creation of meaningful and intuitive sonifications that rely more on users innate cognitive capacities than previous learning is a key issue in sonification. The question of meaning is extensively dealt with in the literature (e.g. [1, 2, 3,9, 10, 11]) and will be explored in detail later in this paper. A third problem area within AD is the deficit of cognitively based research. This is highlighted repeatedly across the literature (e.g. [2, 12, 4]). This will be explored in greater detail later in this paper also.

In summary three main problem areas in the study of AD are:

1- The need for theory.
2- The question of meaning.
3- The need for more cognitively oriented research.

4. GIBSONS AFFORDANCES

The ecological approach to perception pioneered by J.J. Gibson [13] puts forward a model of perception where the possible number of actions an organism can execute within its environment is limited by the affordances (opportunities for action) the environment offers to organism. This theory has been widely employed across a number of prominent research projects in AD [4, 14, 15, 16, 17, 18, 19]. This approach is bound up with the philosophical notion of naive realism; the idea that the human senses offer direct perception of an objective world. This is a notion that fails to take into account the features of the human body within which perception takes place, and also the cognitive processing that is by now known to be an inherent facet of perception. It is here speculated that this approach has contributed to the stagnation of AD by downplaying the role of both cognition and embodiment and misinterpreting the status of the auditory domain by treating sonic phenomena as wholly objective events to which human perception grants unmediated access. For auditory display to truly embrace modern (second generation) cognitive science, a fundamental shift in how researchers think about sound is required. It is here argued that the ecological approach provides only half of the picture when it comes to auditory phenomena. In order to remedy this, AD research must begin to acknowledge the role of cognition in parsing the auditory environment and its affordances. At present this is not the case. By embracing the notion of Enaction (cognition as guided action) [20] a wider view comes into focus, where the environment and affordances are organised and explained not only in physical terms but in terms of common cognitive capacities uncovered by second generation cognitive science. In this way affordances are shaped by the interplay of cognition and the physical world. This grants the notions of an embodied auditory environment that can be extended to AD design allowing for the exploitation of entirely new cognitively based affordances as well as their physical counterparts.

5. COGNITIVISM

Where AD and auditory research more generally doesn't directly embrace Gibson's ecological perspective or naive realism it tends towards a cognitivist model of the mind. This model defines the mind as a computer. The objective world is represented to this computer, via perception, in an array of arbitrary symbols. Thinking is the manipulation of such symbols [20]. Cognitivism acted as the dominant conceptualization of the human mind until the emergence of second-generation cognitive science in the early 1980's. As a field AD has yet to part from this model and truly embrace the implications of modern cognitive science. According to Harnad's Symbol Grounding Problem [21] cognitivism cannot explain meaning (or the process of meaning-making). This is because it defines cognition as the relation of arbitrary symbols to a corresponding objective reality, entirely omitting the role of meaning. Any serious attempt to address the question of meaning in the field of AD must at the very least acknowledge its existence, if not offer a satisfying account of its genesis in relation to auditory perception and cognition. It could be argued that the stagnation experienced by the field may at least be partially resultant from the research and developments of AD solutions that adhere to the by now defunct notion of the cognitivist mind. It is here argued that in the development of theory and design guidelines for the field of AD, careful consideration needs to be paid to the implications of ones theoretical assumptions on the status of cognition, perception and critically, meaning. The assumption of the cognitivist mind must be corrected if we are to avoid the symbol grounding problem. If not, a true account of both meaning and cognition will be forever beyond the reach of AD. With the application of rigour and a serious reconsideration of the role of the mind in AD, new solutions and pragmatic theories can be devised.

6. TOWARDS EMBODIED AFFORDANCES

Embodied affordances are here described as affordances offered by the interplay of cognition and the environment where cognition is defined in terms of second generation cognitive science, more typically known as Embodied Cognition (EC). This theoretical school (EC) may help to inform solutions to the three main problem areas in the study of AD visited earlier. When extended to the auditory domain, the recognition of embodied aspect of affordances allows for design solutions that are better rooted in our cognitive capacities and more implicitly meaningful as a result. Such an approach could potentially address all 3 problem areas in AD simultaneously. This topic shall be discussed in greater detail after a brief account of EC as it currently stands in the field of AD.

7. EMBODIED COGNITION

An alternative theory of mind put forward in the formative years of second generation cognitive science, which has since found scientific validity and wide application [22], is Embodied Cognition. Embodied cognition (EC)

presents a new paradigm for thinking about the perception and cognition of both music and sound [23, 24]. It provides an answer to the symbol grounding problem recognizing meaning-making as a key cognitive task and is said to offer a more accurate view of cognition than cognitivism [25, 26, 27, 22]. Rather than remaining of purely theoretical worth it also provides a comprehensive account of human cognitive competencies, the mental faculties by which a mind cognizes, understands, imagines and reasons. EC researchers concern themselves with topics such as affective/kineasthetic dynamics, conceptual metaphor theory, sensorimotor mimesis, embodied schemata and conceptual blending. Each of these cognitive capacities arises from activity in the sensory-motor system. As a result, they are thought of as being embodied and have been show to organize our cognitive experience in terms of our embodiment. They are intimately bound with the process of meaning making in audition [24] and the other sense domains [16]. The EC framework provides theoretical guidelines for cognitively based research into the question of meaning in auditory display. In so doing it provides a framework in which we can address the three problem areas in AD as they are presented in this document. It has been decided to focus on Lakoff and Johnson's [25, 26] conceptual metaphor theory, and embodied schema as a means to extending Gibson's theory of affordances. These explicit mechanisms have been chosen as not only are they theoretically relevant to meaning-making, but a strong body of empirical research documents their operation [22]. Before outlining a theory of embodied affordances the contribution of EC to AD must be considered.

8. EMBODIED INTERACTION

To date, notions from EC theory have been applied to the field of AD in number of successful ways. Much of this application has been on the side of interaction. AD technologies and systems have been developed on the basis of Dourish's theory of embodied interaction [28]. In embodied interaction, the physical world is treated as the medium for interaction with digital technology. The primary focus of the theory is conversion of action into meaning. It bridges the gap between social and tangible computing and allows for the generation and sharing of meaning through interaction with tangibles. This allows the designer to leverage EC theory in the design of interaction within AD technologies. This theory has provided the basis for research into continuous sonic interaction as an embodied interface [29, 30], sonic interaction design [31] and embodied interaction in auditory display [32, 33]. In order to move beyond interaction design in AD, and to tackle the three questions stated above a broader understanding of EC as it applies to AD is required. Theory based purely on embodied interaction does not allow for applications in the exclusively auditory sub-set of AD.

9. COGNITIVE CAPACITIES

Rather than being simply a theoretical research program, the EC literature offers us detailed descriptions of the cognitive capacities and their workings. Some of these previously listed capacities will now be considered in relation to AD. Conceptual Metaphor Theory (CMT) and Embodied Schema theory are two complementary and scientifically valid streams of thought within EC research [25, 26, 22, 35, 36]. CMT illustrates a cognition in which concepts are composed of cross-connections from source to target domains. A source domain is an intelligible human experience and the target domain is the conceptual space. Meaning is connotative and results from the correlations between these domains referred to in EC as cross-domain mappings. A concept then is a conglomerate of other inter-related aspects of human experience, be they concepts, memories or perceptions themselves. At their root, concepts are grounded in the human experience of embodiment within physical and socio-cultural environments. Certain repetitive patterns or schemata emerge from this embodiment. These schemata provide the logical rules by which concepts and their transformations are governed. Cognition, reasoning and perception are organized in terms of these schemas [37]. According to the invariance principle in CMT target domains retain the image schematic structure from the source domains that inform them [26]. It is the invariance of embodied schematic structure across all domains of human experience that enables the mind to understand abstract and seemingly arbitrary concepts. The embodied mind lends concepts intelligible structure, by projecting logical organization into a conceptual domain from domains of repetitive embodied experience. In working from the point of conceptual structure back to embodied experience, the embodied mind achieves understanding and applies meaning to a concept. The mechanism here simply relies on mapping logical structure in terms of embodied schemata from abstract to more familiar domains where all mapping paths find an ultimate grounding in embodied experience. In EC to reason is to reason about one thing in terms of another where both elements share embodied schematic organization. Understanding and meaning-making work similarly. For example we can reason about balancing a balance sheet in the same way we would balance a see-saw. The common schema here is the twin-pan balance schema. We can understand what it means for one's heart to be inside our chest from our experiences of placing clothes in a drawer. The common schema here is the container schema. The mind can make a musical experience meaningful by relating galloping bass line to the galloping of horses. This notion is richer and contains many embodied schemata (e.g. Source-path-goal and force schemata.) Metaphor has long been acknowledged in the field of linguistics [25]. CMT differs by extending the notion of metaphor so that it is situated as the mechanism by which thought is possible. As discussed above to think is to think in terms of something other than the original thought. In another example above a musical bass line is described as galloping. This represents a metaphorical projection where music is reasoned about and understood in terms of the movement of a horse or similar animal. CMT claims (and offers empirical support) that this isn't just a linguistic mechanism but describes how people think, imagine and reason. The mind actually reasons about the bass line in terms of prior experiences of gal-

loping. The mind takes the schemata present in ones experiences of galloping and uses them to make sense of a bass line.

10. EMBODIED COGNITION IN AUDITORY DISPLAY

The worth of embodied cognition in furthering the field of auditory display has been acknowledged to a degree by researchers working in the area. Embodied schemata and CMT are beginning to find application in the field. They've been used as a framework for designing auditory feedback systems that a user can better understand and reason about (using a balance schema) [38]. Embodied music cognition has been considered as a framework [39, 23] along with more purist approaches that focus on the work of Lakoff and Johnson as well as Varela, Thompson and Rosch's [20] enaction [19, 35]. CMT and embodied schemata theory are being employed in auditory display as a design framework for salient feedback in auditory display environments [38] where an embodied schema is leveraged as a model by which users can reason about and understand the auditory display. This is being employed in the context of interaction. Springboard is one such AD system, which is specifically designed for human cognitive capacities. This approach has been repeated in the field. CMT and embodied schema provide an excellent design framework for intuitive understanding especially in the realm of audio [36, 38, 40, 41, 42, 43, 44]. It is hoped that this approach can be extended to the exclusive auditory domain in the future. CMT and embodied schemata also prove to be well suited to the design of more meaningful auditory displays, where audio signals are of a higher level of salience [26]. Antle et al. [45] have developed AD design guidelines grounded in CMT and embodied schemata theory, which support a listener reasoning and understanding in an interaction context. The application of metaphors in sonification mapping has been a topic of interest for numerous researchers [46, 47, 48]. Design patterns, have been suggested to guide the development of solutions to commonly encountered sonification problems by reusing previously effective sonification metaphors and strategies [49, 50, 51]. There is an excellent body of research testifying to the usefulness of embodied cognition as a framework for the design of auditory display. The projects reported on here are not only concerned with embodied interaction and embodied interfaces, but also integrate AD into the embodied interaction process. The current research maintains an interest in how human bodily nature provides certain cognitive affordances, which may be leveraged for the interpretation of elements of an auditory display. As such, it is necessary to consider some of the cognitive capacities discussed earlier in order to establish how exactly a project of this type may be executed. The applications of EC considered thus far share a common oversight. EC solves the symbol-grounding problem by rooting meaning in embodied experience, and this fact has not been exploited to offer satisfying answers to the question of meaning in AD. It is here proposed that by focusing on the role of cognitive competencies in meaning making, a cohesive account of embodied affordances can act as a design framework. It is here argued that such a framework would be well suited to developing solutions to the three problem areas in AD. Lawrence Zbikowski [24] demonstrates how CMT and embodied schematic transformations drive the process of meaning making in audition at an extremely low level. The very mechanisms by which the mind organizes auditory perceptions make such perceptions meaningful. On the level at which auditory perceptions come to conscious awareness they are instantly meaningful. Zbikowski grounds this meaning making empirically offering it as an explanation for musical experiences. This is an important fact that the current research aims to exploit.

11. EMBODIED AFFORDANCES

Gibson's naive realism argues that we perceive the world directly. This does away with the notion of cognitivism, where the world is represented to cognition in symbolic form via the senses. Concurrently, naive realism fails to offer any satisfactory account of meaning. Rather than answering the symbol-grounding problem, it simply ignores it. As Varela et al. [20] demonstrate that our experience of color in a visual scene arises from the interplay of embodiment, cognitive competencies and the environment. Smallman and John [52] demonstrate the inadequacies of naive realism as a guiding framework for the development of visual displays due to its severe underestimation of both the difficulty and accuracy of visual perception. It is here argued that the same is true for auditory perception. A final argument against naive realism is its disqualification of cognitive capacities in acts of perception. As mentioned earlier it fails to account for meaning by sidestepping the symbol-grounding problem. At the same time, it prevents any cognitively based account of meaning arising through its dismissal of the role of embodied cognitive capacities in perception. In order for AD to tackle the questions of meaning and cognition and to generate sound and useful theory based on these enterprises it must overcome the pitfalls of both naive realism and cognitivism. By reconsidering Gibson's affordances in the light of EC (most specifically CMT and embodied schemata) we are presented with the notion of embodied affordances that overcome the disconnects of both naive realism and the symbol grounding problem thrown up by cognitivism. An embodied affordance can be thought of as any affordance that is open to a user as a result of the users cognitive meaning-making capacities. The notion of the embodied affordance opens up new areas in which to design AD solutions. In order to appeal to these embodied affordances we require AD tools that can better interface with a users cognitive capacities. The current study is concerned primarily with developing such tools. It focuses solely on the auditory portion of auditory display. It does not deal directly with the question of interaction; rather the focus is on designing sonic models to aid understanding and reasonability in AD. This is an attempt at contributing to the three problems areas in AD discussed in this paper. It is hoped that the wealth of EC research in AD has been faithfully represented here and that the need for an exclusive focus on meaning, cognition and the

purely auditory element of AD has been communicated. It is expected that this project will offer tools for designing more intuitive and user-centric AD systems. The development of a design framework based on embodied affordances will expand the breadth of cognitively based research in the area as well as promoting a non-foundational theory that can be pragmatically applied across the field.

12. DESIGNING FOR EMBODIED AFFORDANCES

In order to capitalize on the role of human cognitive capacities in meaning making for the enrichment of the AD field the current project focuses on the modeling of internal logical structures of select embodied schemata and the codifying of these models in the auditory domain. These models make subconscious embodied schemata conscious through metaphorical projection of embodied schemata into the auditory domain. A user can understand and reason about these auditory signals in terms of those schemata by which they are organized. It is hoped that these embodied auditory models will offer the user new cognitive affordances by which they can understand and reason about an AD. A framework of theoretical design guidelines will be drawn up from the development and testing of these auditory models. These models are based on CMT and embodied-schemata principles and represent a first attempt to design elements of AD to offer the user embodied affordances. The first model under development is the twin-pan balance model where dual data inputs are mapped to individual sound objects at equidistant location across both X and Y axes in and auditory space. Changes in the magnitude of the two data inputs may map to salient audio dimensions such as pitch or timbre. Such a model can communicate relational changes between two variables. For example X is larger than Y or Y is decreasing while X increases. Each embodied schemata and each configuration of multiple schemata is conductive to a different form of reasoning and imparts a different meaning. Where the twin-pan balance schema is useful for conveying relationships between two values, the source-path-goal schema is more useful in conveying temporal changes in a single variable. By extending a schema into the auditory domain in this way, an unconscious reasoning strategy is made conscious and offered to the user as a tool by which to more clearly understand and reason about an auditory phenomenon. It is intended to develop a set of such auditory models for deployment in AD systems while also documenting their development and implementation in order to inform a design framework. The proposed theory of embodied auditory models that leverage embodied affordances differs from past AD research in its focus on meaning making. Such a theory is also freed from strict ties to interaction. Embodied affordance design guidelines, should be chiefly concerned with meaning, understanding and reasonability in AD due to theoretical the underpinnings. This allows innovative new solutions to the three problems areas in AD selected at the start of this paper.

13. APPLICATION OF FRAMEWORK

Some experimental evidence is presented here in order to support the idea that embodied schemata can be modeled in the auditory domain using sound synthesis techniques. An embodied schema is gestalt-like pattern. In order to realize that pattern it must be applied to a specific sonic domain. The domain chosen for this experiment is that of vocal synthesis. Being gestalt-like patterns, individual elements of the schema are defined relative to one another, rather than in relation to some external benchmark.
A good analogy for explaining this dynamic is presented by alphabetic letters. For example the letter 'H' is a pattern where two equal length parallel lines are bisected by a third perpendicular line. 'H' can be presented using many different fonts e.g. '*H'*&'**H'** but as long as the internal logic of the pattern is maintained a reader will recognize it as 'H'. Embodied schema too can be presented in many different kinds of sound, but as long as the basic pattern is in place the schema will be recognized. In defining an embodied schema then, each individual element must be described in relation to the other elements of the schema rather than in reference to external measures.

13.1 Hypothesis

This experiment was intended to test the hypothesis that amplitude, frequency profile, pitch level, vowel profile, envelope attack speed, reverb level, compression level and stereo image width could be used, to make a sound seem either Big or Small thus modeling the Big-Small schema discussed by Johnson [36] in the auditory domain. The Big-Small schemata is basic to a listener's embodied experiences of any sound that communicates a sense of size relative to the object or process that created it, the space in which the sound is located, or the sound in itself.

13.2 Design and Materials

The experiment had a 2x4 design with repeated measures on both factors.
The sounds for this experiment were created using additive synthesis techniques to which different degrees of processing mentioned previously. 4 stimuli were given a noisy timbre and 4 a clear vocal like timbre. Both sets of stimuli were then assigned a set of cues to create a Smallest, Smaller, Bigger and Biggest version for each timbre type.
The two stimuli with the Smallest cues are given a low amplitude, a boost in the higher end of the frequency range, a high pitch level, the vowel formant profile for an 'I', a quick amplitude envelope attack speed, a small amount of reverb, little compression and a narrow centrally panned stereo image.
The two stimuli with the Smaller cues have higher values than those of the smallest cues.
The two stimuli with the Biggest cues are given a high amplitude, a boost in the lower end of the frequency range, a low pitch level, the vowel formant profile for an 'A', a slow amplitude envelope attack speed, a large amount of reverb, much compression and a wide diffuse stereo image.

The two stimuli with the Bigger cues have lower values than those of the smallest cues.

13.3 Experimental Procedure

Listeners are presented with each stimulus once and asked to rate the stimulus on a 5-point Likert scale from Very Small to Very Big afterwards. Participants are allowed to listen to the stimuli as many times as needed to help rate them.

13.4 Results

A repeated measures ANOVA was performed on the size schema measures with the design 2(tone: clear vs. noisey) x 4 (size: Biggest Cues vs. Bigger Cues vs. Smaller Cues vs. Smallest Cues) with repeated measures on both factors. There was no main effect of tone $F<1$, but a significant effect of size $F(3, 414)= 19.77$, $p=.000$, $\eta p^2= .125$, and the two variables did not interact $F(3, 414)= 1.56$, $p=.2$.

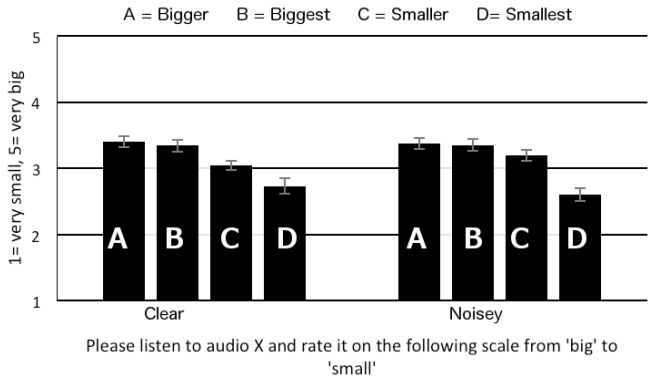


Figure 1. Experimental Results

13.5 Discussion

The results showed that participants accurately identified the stimuli with the smallest and smaller cues but identified stimuli with the bigger cues as slightly larger than those with the biggest cues across clear and noisey timbres. This indicates that the synthesis parameters used to model the Big-Small schema in the auditory domain are effective and that although listeners found it easy to distinguish between sounds modeled after the big and small poles of the schema they found it harder to distinguish between two individual sounds towards the larger end of the schema. This lack of distinction between like sounds may indicate that stimuli should be exaggerated to better enhance distinction. Regardless, the dimensions tested proved useful for modeling the Big-Small schema in the auditory domain. Table 1 presents the parameters required to model both poles of the Big-Small schemata. As discussed in section 13 each of the parameters are defined in relation to one another.

Parameters	Big	Small
Amplitude	Lower	Higher
Energy Profile	LF Energy	HF Energy
Pitch Level	Low	High
Attack Speed	Slow	Fast
Vowel Profile	"a"	"i"
Reverb Amount	Most	Least
Dynamics Range	Small	Large
Stereo Image	Wide	Narrow

Table 1. Parameters for Big-Small Schema in Vocal Synthesis.

14. EXPANDING THE FRAMEWORK

The aesthetic merits of the framework have received some acknowledgement through the well-received performances of data-driven pieces composed within this framework at national (Ireland) and international level. These pieces are intended to evoke a qualitative understanding of the human cost associated with Ireland's recent economic crash. A broader review of the aesthetic and philosophical factors associated with this approach to sonification and the constraints they impose upon individual technical implementations is available elsewhere [53, 54, 55]. It is hoped that in the future, this framework (and those similar) will become more commonplace in the world of AD and can continue to open up new avenues for designing meaning rich data sonifications that speak to a listeners perceptual and cognitive faculties in the same language in which they understand both themselves and their world.

15. REFERENCES

[1] B.N. Walker and M.A. Nees. "Theory of Sonification" in *The Sonification Handbook*. 1st ed., vol.1. T. Hermann, A. Hunt & J.G. Neuhoff Eds. Berlin: Logos Publishing House, 2011, pp.9-39

[2] J.G. Neuhoff and L.M. Heller. "One small step: Sound Sources and Events as the Basis for Auditory Graphs" in Proc. of ICAD, 2005, pp.1-3.

[3] J. Gossman. "From metaphor to medium: Sonification as extension of our body" in *Proc.* ICAD, 2010.

[4] B.N Walker, and G. Kramer. "Ecological psychoacoustics and auditory displays: Hearing, grouping, and meaning making". Ecological Psychoacoustics, pp150-175, 2004.

[5] H. Gardner. *The mind's new science: A history of the cognitive revolution*. Basic books, 1987

[6] J. Dewey. *Art as experience.* Perigee Trade, 1934.

[7] R. Rorty. *Objectivity, relativism, and truth: philosophical papers (Vol. 1)*. Cambridge University Press, 1990.

[8] R. Swinburne. *Objective Knowledge: An Evolutionary Approach*. Philosophical Books, pp.17-20, 1973

[9] A. Eldridge. "Issues in Auditory Display". *Artificial Life*, vol. 12, pp.259-274, 2006

[10] S. Serafin, K. Franinovic, T. Hermann, G. Lemaitre, M. Rinott and D. Rocchesso,. 2011. "Sonic Interaction Design" in *The Sonification Handbook,* 1st ed., T. Hermann, A. Hunt and J.G. Neuhoff. : Logos Publishing House, 2011, pp.88-110.

[11] S. Barass. "Personify: a toolkit for perceptually meaningful sonification" in *Proc. ACMA,* 1995.

[12] J.G. Neuhoff. "Perception, Cognition and Action in Auditory Displays in *The Sonification Handbook*. 1st ed., vol.1. T. Hermann, A. Hunt & J.G. Neuhoff Eds. Berlin: Logos Publishing House, 2011, pp.63-81.

[13] J.J. Gibson. *The ecological approach to visual perception*. Psychology Press, 1986.

[14] P. Sanderson, J. Anderson and M. Watson. 2000. "Extending ecological interface design to auditory displays" in P*roc. Australasian Conference on Computer-Human Interaction*, 2000 pp.259-266.

[15] S. Saue, S. "A model for interaction in exploratory sonification displays*" in Proc. ICAD,* 2000.

[16] P. Sanderson and M. Watson. "From information content to auditory display with ecological interface design: Prospects and challenges" *in Proc. Human Factors and Ergonomics Society Annual Meeting*, 2005, pp.259- 263.

[17] T.C. Davies, C.M. Burns and S.D. Pinder. "Using ecological interface design to develop an auditory interface for visually impaired travellers" *in Proc. 18th Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments,* 2006, pp.309-312.

[18] M. Mustonen. "A review-based conceptual analysis of auditory signs and their design" *in Proc. ICAD*, 2008.

[19] E. Brazil and M. Fernstrom. "Investigating concurrent auditory icon recognition" *in Proc. of ICAD*, 2006 pp.51-58.

[20] F.J. Varela, E. Thompson and E. Rosch. *The embodied mind: Cognitive science and human experience*. MIT press, 1992

[21] S. Harnad. "The symbol grounding problem" *Physica D: Nonlinear Phenomena*, vol. 42 pp.335-346.

[22] G. Lakoff. " Explaining Embodied Cognition Results" *Topics in Cognitive Science* vol. 4 (4), pp.773-785, 2012.

[23] M. Leman. *Embodied music cognition and mediation technology*. Mit Press, 2007.

[24] L.B. Zbikowski. *Conceptualizing music: Cognitive structure, theory, and analysis,* US: Oxford University Press, 2005.

[25] G. Lakoff and M. Johnson. *Metaphors We Live By*. Chicago: Univ. of Chicago Press, 1980.

[26] G. Lakoff and M. Johnson. *Philosophy in the Flesh: The Embodied Mind and it's Challenges to Western Thought.* New York: Basic Books, 1999, pp.3-602.

[27] W. Freeman and R. Nunez. *Reclaiming Cognition: The Primacy of Action, Intention and Emotion*. USA: Imprint Academic, 2000, pp.1-262.

[28] P. Dourish. *Where the Action Is: The Foundations of Embodied Interaction*. USA: The MIT Press, 2001, pp.1-210.

[29] M. Rath and D. Rocchesso. "Continuous sonic feedback from a rolling ball" *in Proc. Multimedia, IEEE*, 2005, pp.60-69.

[30] D. Rocchesso, P. Polotti and S. Delle Monache. "Designing continuous sonic interaction" *International Journal of Design*, vol. 3 pp.13-25, 2009.

[31] A. Dewitt and R. Bresin. "Sound design for affective interaction" in *Proc. Affective Computing and Intelligent Interaction*, 2007, pp.523- 533.

[32] R. Wakkary, M. Hatala, R. Lovell, and M. Droumeva. "An ambient intelligence platform for physical play" in *Proc. ACM international conference on Multimedia,* 2005, pp.764-773.

[33] M. Droumeva and R. Wakkary. "Understanding aural fluency in auditory display design for ambient intelligent environments" in *Proc.* ICAD, 2008.

[34] M. Droumeva, S. De Castell and R. Wakkary. "Investigating Sound Intensity Gradients as Feedback for Embodied Learning" in *Proc.* of ICAD, 2007, pp.26-9.

[35] S.C. Peres and M.D. Byrne. "The Interactive Behavior Triad and Auditory Graphs" in *Proc.* ICAD, 2005.

[36] M. Johnson. *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason.* Chicago: University of Chicago Press, 1987.

[37] B. Hampe and J.E Grady. *From perception to meaning: embodied-schemas in cognitive linguistics.* Ber-

lin: Mouton de Gruyter, 2005.

[38] A.N. Antle, G. Corness and M. Droumeva. "What the body knows: Exploring the benefits of embodied metaphors in hybrid physical digital environments", *Interacting with Computers: Special Issue on Physicality*, pp.66-75, 2009.

[39] N. Diniz, P. Coussement, A. Deweppe, M. Demey and M. Leman. "An embodied music cognition approach to multilevel interactive sonification", *Journal on Multimodal User Interfaces*, pp.1-9, 2009.

[40] A.N. Antle, G. Corness and M. Droumeva."Human-Computer-Intuition? Exploring the cognitive basis for intuition in embodied interaction", *International Journal of Art and Technology,* vol. 2, pp.235-254, 2009.

[41] A. Macaranas, A. Antle and B.E. Riecke. "Bridging the gap: attribute and spatial metaphors for tangible interface design", in *Proc. International Conference on Tangible, Embedded and Embodied Interaction*, 2012, pp.161-168.

[42] J. Hurtienne and L. Blessing. "Design for Intuitive Use-Testing image schema theory for user interface design" in *Proc. International Conference on Engineering Design*, 2007 pp.1-12.

[43] J. Hurtienne. "Cognition in HCI: An ongoing story", *Human Technology*, vol. 5, pp.12-28, 2009.

[44] J. Hurtienne, K. Weber and L. Blessing. "Prior experience and intuitive use: embodied-schemas in user centred design", *Designing inclusive futures*, pp.107-116, 2008

[45] A.N. Antle, G. Corness, S. Bakker, M. Droumeva, E. Van Den Hoven and A. Bevans. "Designing to support reasoned imagination through embodied metaphor", *in Proc. Conference on Creativity and Cognition*, 2009, pp.275-284.

[46] K. Vogt and R. Höldrich, R. "A metaphoric sonification method-towards the acoustic standard model of particle physics" in *Proc. of ICAD*, 2010.

[47] N. Schaffert, K. Mattes, S. Barrass and A.O. Effenberg. "Exploring function and aesthetics in sonifications for elite sports" in *Proc. of ICoMCS2*, 2009, pp.83-86.

[48] S. Barrass" EarBenders: Using stories about listening to design auditory interfaces" in *Proc. APCHI*, 1996.

[49] S. Barrass, S. "Sonification design patterns" in *Proc. ICAD*, 2003.

[50] C. Frauenberger and T. Stockman. "Auditory display design—an investigation of a design pattern approach". *International Journal of Human-Computer Studies*, vol. 67, pp.907-922, 2009.

[51] M. Adcock and S. Barrass. "Cultivating Design Patterns for Auditory Display: in *Proc. ICAD*, 2004.

[52] H.S. Smallman and M. John. "Naive realism: Limits of realism as a display principle" in *Proc. Human Factors and Ergonomics Society Annual Meeting*, 2005, pp.1564-1568.

[53] S. Roddy and D. Furlong. 2014. "Embodied Aesthetics in Auditory Display". *Organised Sound*, vol. *19*(01), pp.70-77, 2014.

[54] S. Roddy and D. Furlong. 2013. "Embodied Cognition in Auditory Display" in *Proc*. *ICAD*, 2013.

[55] S. Roddy and D. Furlong. 2013. "Rethinking the Transmission Medium in Live Computer Music Performance". Available at: http://issta.ie/wp-content/uploads/ISSTC-2013-RODDY.pdf [June 2015].

MUSICALLY INFORMED SONIFICATION FOR SELF-DIRECTED CHRONIC PAIN PHYSICAL REHABILITATION

Joseph W. Newbold & Nadia Bianchi-Berthouze
UCLIC, UCL
joseph.newbold.14@ucl.ac.uk
n.berthouze@ucl.ac.uk

Nicolas E. Gold
Dept. of Computer Science, UCL
n.gold@ucl.ac.uk

Amanda Williams
Dept. of Clinical, Educational & Health Psychology, UCL
amanda.williams@ucl.ac.uk

ABSTRACT

Chronic pain is pain that persists past the expected time of healing. Unlike acute pain, chronic pain is often no longer a sign of damage and may never disappear. Remaining physically active is very important for people with chronic pain, but in the presence of such persistent pain it can be hard to maintain a good level of physical activity due to factors such as fear of pain or re-injury. This paper introduces a sonification methodology which makes use of characteristics and structural elements of Western tonal music to highlight and mark aspects of movement and breathing that are important to build confidence in peoples body capability in a way that is easy to attend to and devoid of pain. The design framework and initial conceptual design that uses musical elements such as melody, harmony, texture and rhythm for improving the efficiency of the sonification used to support physical activity for people with chronic pain is here presented and discussed. In particular, we discuss how such structured sonification can be used to facilitate movement and breathing during physical rehabilitation exercises that tend to cause anxiety in people with chronic pain. Experiments are currently being undertaken to investigate the use of these musical elements in sonification for chronic pain.

1. INTRODUCTION

Chronic pain (CP) affects millions of people worldwide [1]. CP is a persistent pain that remains after the expected time of healing [2]. According to the 2009 UK Chief Medical Officer's Report, "Each year, 5 million people in the United Kingdom develop chronic pain, but only two-thirds will recover" and "In England, there is currently only one pain specialist for every 32,000 people in pain" [3]. Due to this large number of cases, and the life changing effect CP has on people's lives, self-management is the main form of therapy. This work focuses specifically on Musculoskeletal chronic pain (MCP) and how sonified movement and breathing can be used to support them during physical activity. Remaining physically active is an important way for people with MCP to manage pain and maintain everyday functioning [4]. However in the presence of pain it can be difficult to maintain a good level of physical activity as people can become fearful of specific movements. As described by Leeuw *et.al*, this fear of movement is caused "when stimuli that are related to pain are perceived as a main threat" [5]. Anxiety differs from fear as it is defined as the *anticipation* of threat. Anxiety can cause people to become hyper-vigilant of movements they perceive as threatening [5].

It is suggested that a way to reduce fear and anxiety related to these movements is improving self-efficacy [6]. Self-efficacy is the confidence in one's own ability to achieve particular goals and evidence has shown that a person with higher self-efficacy is less likely to display movement avoidance behaviour [6]. This is why it is important that a person with MCP is able to explore and understand their capabilities. Unfortunately, anxiety and over-attention to pain often does not allow people to accurately perceive their capabilities. One way that self-efficacy can be improved is through the use of feedback that can be easily attended to by reducing anxiety and overriding the attention to pain [7]. In Singh *et.al* [7], the authors identify some of the important information of which people need to be aware in order to gain confidence and be able to perform physical rehabilitation. In particular, understanding the range of movement that they are capable of before they start adopting protective strategies (e.g. guarding) or before the pain increases, correct pacing, and the minimum amount of movement for days of increased pain. It was shown that using sonification to represent movement led to increased engagement and self-efficacy in people with MCP. Sonification was used as a method for feedback due to the omnidirectional natural of sound that does not require focusing on a screen. Sound feedback has been shown to initiate motor activity and facilitate motor learning in clinical [8] and educational settings [9]. Additionally sound can be used to convey multiple streams of data at one time in a way that is still understandable [10].

In this paper, we build on that work to provide a conceptual framework to sonify movement and breathing in MCP physical rehabilitation exercises that may induce anxiety. By providing people with MCP information about their body and movement, they can better understand their capabilities and build confidence in their movement. We propose a "musically-informed" sonification approach that makes use of music structure to represent critical information about movement and breathing. By thinking about

Copyright: ©2015 Joseph W. Newbold & Nadia Bianchi-Berthouze 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.

music as not as an artistic medium but as a way to organise sound, the characteristics and structural elements of Western tonal music can be used to organised sonification in a way that provides relevant information that it easy to attend to and process. We briefly introduce chronic pain and discuss the barriers to physical activity and review the literature on the use of sound in rehabilitation. We then present our musically-informed sonification framework grounded on aspects of physical activity that are important to facilitate physical rehabilitation. We then exemplify the use of the framework, by proposing an implementation of the sonification of anxiety-inducing exercises typically used in lower back MCP physical rehabilitation.

1.1 Sonic Feedback in Chronic Pain Rehabilitation

Sonification has many benefits over traditional methods for portraying data, for example, ears have a slower rate of fatigue than eyes, the omnidirectional nature of sound means that there is no need to focus on a screen, and the auditory system has the ability to perceive multiple streams of data at once or hone into to individual streams [10]. Music has seen application in CP by way of musical analgesics, where music can be used either instead of or in conjunction with traditional pain relief to reduce pain levels [11]. However as shown by Finlay, the effect of music as pain relief, is time limited and short term without cumulative effects [11]. While this may be seen as a disadvantage for its use in MCP management, it does highlight the fact that the use of sound (especially that which is musical in nature) as a feedback mechanism could be beneficial for people with MCP to undertake physical activity without pain being their main point of focus.

Nazemi *et.al* developed a series of "Soundwalks" in order to help people with CP manage anxiety [12]. These soundscapes were designed to be listened to by people with CP while in clinical waiting rooms to reduce their anxiety and help with the afterwards healthcare consultation. The soundscapes used binaural recordings of environmental sounds to emulate the feeling of going on a relaxing journey. Work has been done by Vidyarthi *et.al* to use sound to create an immersive experience for relaxation [13]. Their "Sonic Cradle" is situated in a darkened room in which the person controls different synthesised sounds through their breathing, slowly building a soundscape specific to them. This creates an experience that is not only designed for relaxation but draws on ideas from mindfulness meditation that directs focus on the person's self in that moment. It is suggested by Vidyarthi *et.al* that such a system could be a powerful tool to help people with CP and other chronic ailments manage anxiety.

The PhysioSonic project focuses on helping people undergoing physiotherapy to better understand their body and their movement through motion capture data sonification [8]. The system used two scenarios, one emulates a woodland scene where people would reach from the ground to the birds in the sky and one where their movement would drive the playback and manipulation of a musical sample/text. The majority of people showed improvement in both shoulder flexibility and reduced evasive movement

Singh *et.al* present the "going with the flow" system, that uses sound to provide information on the current position in a movement [7]. As the person stretches forward, a smartphone on their back is used to measure the current position within a pre-calibrated exercise space, a space calibrated for each individual for each exercise. As the person moves through this space, ascending then descending notes of a major scale are played to represent their current position in the movement. This use of structured sound to provide information on the structure of movement was found to have positive results on how people with MCP perceived their movement and it was reported in interviews that people found the sound feedback helped them "hear how they were doing" [7].

In this paper, we build on the positive results shown in Singh *et.al* [7] to develop a framework for using aspects of Western tonal music that are understood implicitly, to create sonifications that deliver feedback on people's movement and breathing.

1.2 Implicit Music Understanding

There are several aspects of music that people with no formal music training are able to discern. This is due to the implicit knowledge that is gained through exposure to music in people's day to day lives (the nature of the specific musical knowledge gained clearly depends on the culture in which a person is immersed day to day; for this work we assume Western tonal music). For the purposes of this framework we will focus on the aspects of melody, harmony, texture and rhythm to be used for the sonification of physical activity for people with MCP.

It has been shown that people are able to easily recognise familiar tunes [14]. Deutsch showed that people are able to identify a well-known tune, "Yankee Doodle", in three different octaves with 100% efficiency from hearing the first half alone [14]. This work illustrates how different melodies can be easily identified by non-musically trained individuals. The work done in the area of earcons also demonstrates how structured information can be given through the use of melodic phrasing [15].

Bigand demonstrates that the idea of musical stability, whether a piece of music creates tension and expectation (instability) or resolution (stability) can be identified by people with no formal music training [16].

Another aspect of how musical structure is perceived is demonstrated by the way in which people tend to synchronise their movement with music. Sensorimotor synchronisation (SMS) is the synchronisation of peoples' body movements with an external reference and can be thought of most simply in the form of tapping to a beat or dancing, see Repp [17] for a review. Aschersleben discusses how this SMS is affected by musical experience, with non-musicians showing a 10ms longer asynchrony than people who reported playing a musical instrument as a hobby [18]. Although this shows that musical training improves this ability, the phenomenon is still observed in non-musicians.

Given that these aspects of music that can be understood implicitly by people with no formal musical training, it seems reasonable that they could be used by a broad range

of people with MCP to provide more effective and engaging sonified movement feedback.

2. CONCEPTUAL DESIGN FOR MUSICALLY-INFORMED SONIFICATION

In section 1.1, it was shown that sound has been used to provide both relaxation and feedback for people with CP. By using the features of music outlined in section 1.2 this information display can be enhanced to provide more effective feedback to the people, regardless of musical training. This paper considers musically-informed sonification as a way of thinking about the space of exercise for people with MCP. We describe below four musical elements that can be used to represent important information during physical rehabilitation. These elements can be used as building blocks to define sonified exercise spaces.

In physical rehabilitation, a purely biomechanical approach has been used to model movement. For example, a physical exercise (e.g., sit-to-stand, reaching forward) can be divided into phases where different phases may be characterized by changes in the movement dynamics and/or by involvement of different body parts. However, in the case of MCP, psychological (rather than simply biomechanical) aspects need to be taken into account as they may bias the perception of movement and its relationship to pain. We hence describe physical activity taking into account these two perspectives and propose how they can be represented through a musically informed sonification. Table 1 summarises a preliminary framework for how these implicitly-understood aspects of music can be used to highlight important aspects of physical activity.

Musical parameter	MCP parameter
Melody	Identifiers for different movement types and phases
Harmonic and cadential structure	Markers for specific points in movements that give either motivation (unstable), conclusion(stable) or a choice for continuation (relatively stable)
Texture	Reward for movement outwith a defined exercise space
Rhythm	Pacing for exercises, synchronisation between breathing and movement

Table 1. This table outlines the preliminary mappings for this framework, showing how musical parameters can be used to highlight aspects of physical activity important to MCP

Melody - Movement phase When a movement triggers anxiety in people with MCP, their physiotherapist breaks it into phases and ask the person to gradually build through those phases as confidence is built [7]. For example, in sit-to-stand, a physiotherapist may ask a person to first gain confidence in bending their trunk forward to gain momentum before they start to exercise the standing phase. Movement phases could hence represent biomechanical phases of a movement, milestones people want to achieve as they go through a movement, or by phases of a movement that people tend to avoid. We suggest representing phases of a movement through melodic phrases. Since people seem to have an innate ability to recognise melodies implicitly, it can be used to convey information on the different phase of the movement being done. At the same time the pitch structure of the melody can provide information about the progress within the phase. A series of melodies can be used to build complex movements whose parts can be easily recognized and provide a perception of where the body is through the movement.

Harmony/Cadence - Structured Motivation Overdoing may lead to setback, at the same time underdoing may lead to further body debilitation. In order to maintain a good level of activity, people with MCP set targets they want to achieve and then gradually build on these targets. At the same time they may reset their targets during bad days (i.e., days when the pain is very strong) to ensure activity without overdoing. Overdoing may lead to setback, at the same time underdoing may lead to further body debilitation. We propose the use of harmonic structure and cadence to represent targets and intermediate milestones that indicate progress towards those targets. We identify three types of targets: (i) a milestone that needs to be achieved but that does not indicate the end of the movement; (ii) a minimum target that needs to be achieved and where a movement may end if desired (e.g., for bad days or for facilitating building); and (iii) a maximum target where the movement need to end to avoid overdoing. The use of these targets depends on the psychological and physical needs and capabilities of the person. As discussed in section 1.2 musical stability can be thought of as the likelihood of a musical piece ending or whether it must continue to resolution and this can be manipulated in part through the use of harmony. The level of musical stability can be used to create a degree of ambiguity as to how far a certain phrase or movement should go before ending. By exploring different levels of stability at key points in a movement, a person could receive additional motivation through the use of tension to continue that movement. At the same time and possibly more importantly, the end of the set exercise space could resolve onto a point of relative stability but not complete resolution allowing the person to continue if they wish, but also the music to have an acceptable ending if not. Cadence points can be used within a melodic phrase or to characterise the ending of a phrase or the transition to a separate phase.

Texture - unstructured motivation Some phases of a movement may not have specific end points and thus cadence and melody may not be suitable to model them. Another musical element that can be used to provide information on progress through movement without a specific ending is texture. Texture as defined by Cohen and Dubnov is "the way of distributing the sound (of defined or undefined pitches) in the dimensions of frequency, time, and intensity" [19]. In this paper we refer to a simplified idea of texture that remains harmonically static but increases in in-

tensity by increasing the number of notes in a given metrical timeframe, an increasing spread in the voicing of those notes and decreasing the duration of each note. This mapping would reward movement with a richer soundscape, but is not limited by a defined exercise space.

Rhythm/Tempo - Pacing Breathing is a very important part of movement. Deep breathing can lead to relaxation whereas shallow breathing indicates and leads to anxiety [7]. In addition, breathing (as a pleasurable sensation) can be used to help keeping the right pacing . Using the idea of sensorimotor synchronisation (SMS) and a simple breathing sonification, the pacing of peoples' breathing could be regulated using a reference rhythmic pattern at a set tempo. Through this mapping a person can refocus on a more positive aspect of activity which is then used to synchronise breathing and movement to a set pace.

3. AN EXAMPLE OF IMPLEMENTATION OF THE FRAMEWORK

Below we outline a series of examples of how the different aspects of the framework could be utilised by people with MCP to develop their own sonification, the individual composition of these sonifications can be left to the specific person with MCP and possibly with physiotherapist to apply this framework suits their individual needs and exercises. The below example demonstartes a possible application of this framework on a common exercise in the physical rehabilitation of chronic back pain : the stretch forward [7]. This exercise can be thought of as having three phases, an initial stretch to a minimum amount of stretch, a stretch to a target point and then a final phase that allows continuation, see Figure 1.A.

3.1 Melodic Design

As in Singh *et.al* [7], the ascending then descending scale can be taken as a base for the simple stretch forward movement (Figure 1.A, with the ascending section corresponding to the first half of the movement (phase 1) where the back is bent and the stretch begins, and the descending section corresponding to the final reach forward (phase 2). This melody can then be used to inform the design of other movements, for example the sit-to-stand movement (shown in Figure 1.B).

The sit-to-stand movement can be broken down into three phases with the bending of the trunk to gain momentum for standing up being the first phase (the one most avoided by people with MCP as perceived as inducing pain). However, by avoiding it, standing becomes harder and they may need to use strategies that in the longer term may indeed induce pain or may in itself reduce confidence in the ability of performing the movement. The same ascending melody used for the stretching forward could be used to sonify the first phase of the sit-to-stand. The sonification of the phase through melody may increase awareness of its performance or the lack of it. This breakdown of movement phases into simple melodies could be a useful way to build more complex movements and could enable collaborative exploration and learning of new movements between the

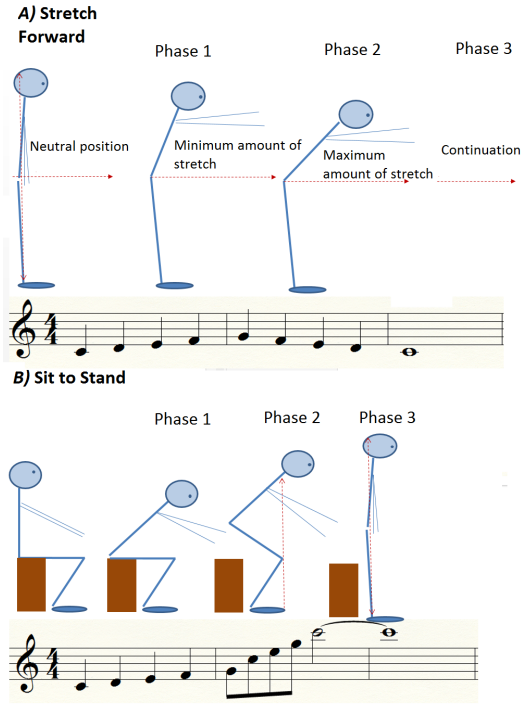


Figure 1. An example of how the melodic design could be used to highlight the similar sections of the two movements, showing a stretch forward and a sit to stand that uses the same melody for the initial stretch forward.

person with MCP and physiotherapist, perhaps in a similar fashion to that demonstrated by Smith and Claveau [9].

However this design requires defining the boundaries of a movement space to increase confidence in moving. As discussed, our framework allows us to define boundaries that may encourage or not trespassing them. Additionally the simplistic musical phrases provide little motivation and engagement and thus far focus only on the person's movement.

3.2 Stability Design

In the stretching forward exercise, the boundaries can be thought as the passing of the minimum amount of stretch desired and the maximum amount of stretch required. Through the use of harmony, we can now mark both a possible conclusion or continuation of a movement when a boundary is reached. For example, the minimum amount of stretch can be thought as a milestone to be reached but we may not want to encourage stopping at that point. Whereas, the maximum target point could be either designed to encourage stopping at that point or to encourage and reward further building on that target. As shown in Figure 2, the same ascending/descending pattern could be used as discussed above, but by more firmly establishing a harmonic context, cadences can be used as the markers of the anchor points and promote continuation or conclusion.

Figure 2.A shows this unstable design, where an imperfect cadence is used to mark the the minimum amount of stretching point as an intermediate milestone and encour-

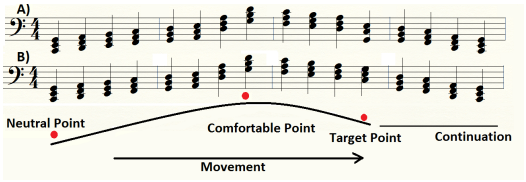


Figure 2. A) Unstable design, uses a second inversion tonic chord to promote an amount of instability at the maximum target point to allow continuation. B) Stable design, resolves to the tonic at the target point to indicate ending. In both designs an imperfect cadence is used at an intermediate point to provide motivation to continue towards resolution.

age continuation. This relatively unstable cadence offers some amount of resolution but also allows continuation. The same can be used for the maximum target point when building over it is encouraged (continuation phase). Figure 2.B shows a counter example for the maximum target point where a perfect cadence is used in place of the imperfect one, so at the anchor points the harmony is resolved and there is no motivation to continue.

It should be noted that in the case of the stretching forward exercise, these designs treat the maximum stretch position as the point of musical conclusion. In future work the return movement could be examined and the final neutral position would be the concluding point, with the cadences above acting as the "turning point" for the movement. This simple design exercise seeks to highlight the benefits that manipulating harmonic stability could provide to an exercise space, however they do not examine some of the difficulties that trying to isolate harmony from other musical factors such as voicing, rhythm and note duration would entail (indeed, greater success might be achieved by explicitly manipulating the combination of these musical parameters: a subject of our future work).

3.3 Texture Design

An alternative mapping could be from movement to texture. In this instance texture can be thought of as the richness of the sound. This mapping would allow the sound to retain its harmonic stability and would not necessitate any specific points be reached. For this implementation the amount of movement from the neutral position could be mapped to the amount of arpeggiation around a base note/chord, with increasing movement leading to a higher probability of a higher number of notes occurring in a given metrical timeframe. This mapping rewards movement no matter how far the person moves and simply decreases back to a single note as they return to the neutral point. This design therefore offers a much greater amount of freedom in the movement, with potentially no set boundaries for the exercise space. This is very important to encourage body exploration (for example) and to ensure that no message of "right" or "wrong " movement can be inferred. This is particularly important in the first phase of CP rehabilitation where the person is learning to gain confidence in moving and managing their anxiety [7]. However the obvious downfall of this design is the lack of structure which would make it difficult for people to gauge how far they have gone or if they have reached a certain point. Although over time judgement may be formed as to the meaning of a given level of arpeggiation, the precision of such judgements would be intrinsically low. This method also does not allow for the definition of multiple movements as it merely correlates the composite richness of texture with extent of movement. More complex movements might be modelled by mapping different movement types to the constituent components of that composite.

3.4 Rhythm Design

By using a sonification of people's breathing, this more pleasurable sensation can be made the focus of their activity and promote a more relaxed state [7]. Our initial sonification provides a high-pitched note for the inhalation and a lower-pitched note for the exhalation (V-I) with the volume of the notes mapped to the respiration amplitude, similar to the note sonification used by Watson and Sanderson's respiration sonification used to help anesthesiologists monitor the breathing of patients [20]. By providing a reference rhythm (in terms of the durations of the high and low pitches), the person with CP is encouraged to bring their breathing in time with reference pitches by listening to the sonification. This rhythm could then also be used to pace the movement, while also bring the breathing and movement in to synchrony. Although this design focuses on what may be a more pleasant aspect of movement it may facilitate better pacing.

4. CONCLUSIONS AND FUTURE WORK

We presented a framework where different musical elements are used to inform the sonification of different elements of an exercise space. These sonifications can be used to structure information that is easier to attend to and devoid of pain and hence enable confidence-building in movement. Each of the elements of the sonification spaces has its own benefit and through their combination different sonification spaces can be designed. By using a combination of the melodic and instability designs, cadences can be used to mark milestones for various movements. Movement beyond the target or comfort points can be rewarded through textural changes but without the need for a defined space (and thus a pre-defined harmonic structure). The breathing feedback could be used in unison and with a reference rhythm to provide pacing for both breathing and movement activities. It may even be possible to synchronise breathing and movement with the milestones, as shown in Figure 3.

In future work we will investigate further the effects of these individual musically-informed designs to evaluate how they can be used by people with MCP in their physical activity. We will explore the degree to which the sonifications are understood in the context of prior musical training. This will be done using a series of studies investigating each aspect of the proposed framework individually, evaluating both how the affect peoples physical activity and their

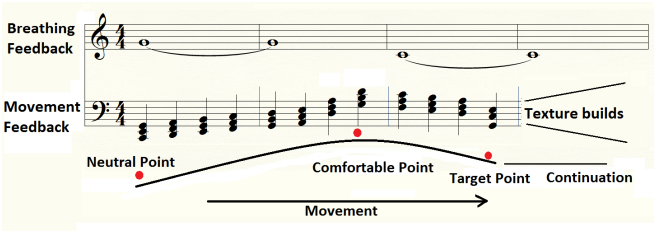


Figure 3. A combination of the designs for a musically informed sonification that uses parts of all four designs discussed in section 3.

perceptions of it, specifically their self-efficacy. It will then be explored how these aspects can be used together as described above as a single musically informed sonification system. This will determine the success of this framework, whether it can use this implicitly understood musical elements to highlight important aspects of feedback. Outside of MCP this style of framework could be used in other areas were it is important to promote awareness, wherein the specific mappings used may vary dependent on the feature to be highlighted, but the principle of using these musical elements to highlight important aspects of the feedback.

Acknowledgments

This work was supported by the UK Engineering and Physical Sciences Research Council Pain rehabilitation: E/Motion-based Automated Coaching project [grant number: EPSRC EP/G043507/1] `www.emo-pain.ac.uk` where audio/video examples of initial prototype designs will be posted.

5. REFERENCES

[1] IASP, "Unrelieved pain is a major global healthcare problem." [Online]. Available: http://www.efic.org/userfiles/PainGlobalHealthcareProblem.pdf

[2] D. C. Turk and T. E. Rudy, "IASP taxonomy of chronic pain syndromes: preliminary assessment of reliability," *Pain*, vol. 30, no. 2, pp. 177–189, Aug. 1987.

[3] L. Donaldson, "150 years of the Annual Report of the Chief Medical Officer: on the state of public health 2008," *London:Department of Health*, 2009.

[4] J. A. Hayden, M. W. van Tulder, A. Malmivaara, and B. W. Koes, "Exercise therapy for treatment of non-specific low back pain." *The Cochrane database of systematic reviews*, no. 3, p. CD000335, Jan. 2005.

[5] M. Leeuw, M. E. J. B. Goossens, S. J. Linton, G. Crombez, K. Boersma, and J. W. S. Vlaeyen, "The fear-avoidance model of musculoskeletal pain: Current state of scientific evidence," pp. 77–94, 2007.

[6] S. R. Woby, M. Urmston, and P. J. Watson, "Self-efficacy mediates the relation between pain-related fear and outcome in chronic low back pain patients." *European journal of pain (London, England)*, vol. 11, no. 7, pp. 711–8, Oct. 2007.

[7] A. Singh, A. Klapper, J. Jia, A. Fidalgo, A. Tajadura-Jiménez, N. Kanakam, N. Bianchi-Berthouze, and A. Williams, "Motivating People with Chronic Pain to Do Physical Activity: Opportunities for Technology Design," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 2803–2812.

[8] K. Vogt, D. Pirrò, I. Kobenz, R. Höldrich, and G. Eckel, "PhysioSonic - Evaluated movement sonification as auditory feedback in physiotherapy," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5954 LNCS, 2010, pp. 103–120.

[9] K. Smith and D. Claveau, "The Sonification and Learning of Human Motion," *20th International Conference on Auditory Display (ICAD 2014)*, 2014.

[10] R. McGee, "Auditory Displays and Sonification: Introduction and Overview," 2009.

[11] K. Finlay, "Music-induced analgesia in chronic pain: Efficacy and assessment through a primary-task paradigm," *Psychology of Music*, 2014.

[12] M. Nazemi, M. Mobini, T. Kinnear, and D. Gromala, "Soundscapes: A prescription for managing anxiety in a clinical setting," 2013.

[13] J. Vidyarthi, B. E. Riecke, and D. Gromala, "Sonic Cradle," in *Proceedings of the Designing Interactive Systems Conference on - DIS '12*. New York, New York, USA: ACM Press, Jun. 2012, p. 408.

[14] D. Deutsch, "Octave generalization and tune recognition," *Perception & Psychophysics*, vol. 11, no. 6, pp. 411–412, Nov. 1972.

[15] D. McGookin and S. Brewster, "Earcons," in *The Sonification Handbook*, T. Hermann, A. Hunt, and J. Neuhoff, Eds. Logos Verlag Berlin, 2011.

[16] E. Bigand, "Perceiving musical stability: The effect of tonal structure, rhythm, and musical expertise." *... of Experimental Psychology: Human Perception and ...*, 1997.

[17] B. H. Repp, "Sensorimotor synchronization: A review of the tapping literature," *Psychonomic Bulletin & Review*, vol. 12, no. 6, pp. 969–992, Dec. 2005.

[18] G. Aschersleben, "Temporal control of movements in sensorimotor synchronization." *Brain and cognition*, vol. 48, no. 1, pp. 66–79, Feb. 2002.

[19] D. Cohen and S. Dubnov, *Gestalt phenomena in musical texture*, 1997.

[20] M. Watson and P. Sanderson, "Sonification supports eyes-free respiratory monitoring and task time-sharing." *Human factors*, vol. 46, no. 3, pp. 497–517, 2004.

A DTW-BASED SCORE FOLLOWING METHOD FOR SCORE-INFORMED SOUND SOURCE SEPARATION

F.J. Rodriguez-Serrano
F.J. Canadas-Quesada
Universidad de Jaen
{fjrodrig,fcanadas}@ujaen.es

J. Menndez-Canal
R. Cortina
Universidad de Oviedo
raquel@uniovi.es
jonatanmenendez@gmail.com

A. Vidal
Universidad Politecnica de Valencia
avidal@dsic.upv.es

ABSTRACT

Along this work, a new online Dynamic Time Warping (DTW) based score alignment method is used over an online score-informed source separation system. The proposed alignment stage deals with the input signal and the score. It estimates the score position of each new audio frame in an online fashion by using only information from the beginning of the signal up to the present audio frame. Then, under the Non-negative Matrix Factorization (NMF) framework and previously learned instrument models the different instrument sources are separated. The instrument models are learned on training excerpts of the same kinds of instruments. Experiments are performed to evaluate the proposed system and its individual components. Results show that it outperforms a state-of-the-art comparison method.

1. INTRODUCTION

The goal of Sound Source Separation (SSS) is to segregate constituent sound sources from an audio signal mixture. The SSS task is of interest because a lot of direct user applications can be developed with it. Personalizing a live concert by letting the listener adjust the volume of individual instruments is one application of online SSS. Music education applications can be developed with both offline and online SSS methods.

There are some types of information that can be introduced in the SSS process. Spectral information can be introduced by using instrument models when the instruments are known in advance [1]. Also, musical score information can be used if the score and audio are well aligned [2].

The problem of audio-to-score alignment (or score matching) is the task of synchronizing an audio recording of a musical piece with the corresponding symbolic score. In offline alignment, the whole performance is accessible for the alignment process, i.e. it allows us to "look into the future" while establishing the matching. Online alignment, also known as score following, processes the data in realtime as the signal is acquired.

Copyright: ©2015 F.J. Rodriguez-Serrano 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.

In this paper we deal with the problem of online score-informed separation of harmonic musical sources from a single-channel recording. We combine a novel audio-score alignment model with the Multi-excitation per Instrument (MEI) NMF model proposed in [3] to build the whole informed separation system. In a previous work [4], this schedule is used with a different alignment method [2]. We then advance this system by using our own alignment method which has a reduced computational complexity and, in the future, could be implemented on devices with limited resources. Therefore, our final system takes a music score and pre-learned instrument models as prior information, aligns the score with the audio and separates the audio mixture, all completed in an online fashion.

In the experiments, we show that the proposed online algorithm separates sources almost as well as the offline version of the algorithm and the proposed alignment method outperform those published in [4].

1.1 Related Work

There are some online approaches for source separation under the NMF-SSS framework [5–7]. However, [5, 6] were only tested in speech enhancement applications as they were designed to adapt one source (speech or noise) but keeping the other source fixed during separation. In our proposal, multiple instruments are learned and separated. The method proposed in [7] is suitable for multichannel signals, but not for monaural ones. The mixing information (which represents the spatial information) is very important for their system. Also, random initialization of the model parameters without any extra information would make it difficult to discriminate between the different sources at monaural sources. To date, none of these approaches [5–7] have been applied to work in the score-informed source separation setting.

Audio-to-score alignment is traditionally performed in two steps: feature extraction and alignment. On the one hand, the features extracted from the audio signal characterize some specific information about the musical content. On the other hand, the alignment is performed by finding the best match between the feature sequence and the score. In fact, classical offline systems rely on cost measures between events in the score and in the performance. Two well known methods in speech recognition have been extensively used in the literature: statistical approaches (e.g. HMMs) [2], and DTW [8]. Although these techniques

achieve good results, their best performances are obtained in offline applications; the results decrease significantly for real-time audio-to-score alignment.

The present work is based on [4], where an online score-informed source separation system is proposed. This work uses a previously proposed alignment stage [5] to inform a SSS algorithm with adaptive instrument models. Here, we are going to use our own alignment method to supervise this separation process. The model adaptation stage proposed in [4] has been supressed because the aim of this work is to assess the proposed alignment method while the NMF-based source separation algorithm is used by the same way. Summarizing, only previously trained instrument models will be applied over the NMF-based SSS algorithm which will be score-informed by a new score alignment stage.

Besides [2], there are several other online polyphonic audio-score alignment methods [8, 9]. We compare the proposed alignment method, applied over a SSS system, versus the one proposed in [2] because it is designed to align multi-instrument polyphonic music audio with score information and it is has been tested on multi-instrumental polyphonic audio with clear objective measures. Despite the fact that the proposal of [8] is tested over piano music and multi instrument signals, its performance is not evaluated over multi instrument signals with objective results, due to the lack of reliable annotations. Other methods are only designed for, or tested on, [9] single-instrument polyphonic audio.

2. BACKGROUND

In this section the methods from the bibliography that the proposed system builds on are summarized. The aim of the proposed work is to introduce a DTW-based online Score Following method in a previous SSS algorithm. The SSS algorithm uses an NMF framework initialized with the score information for the activations and previously trained instrument models for the spectral patterns. This complete framework has been suitably modified to be able to segregate the individual signals in an online manner.

2.1 Instrument models

The instrument model used in the current work is *Multi-Excitation per Instrument* (MEI) model [3]. Let $X(t, f)$ be the true time-frequency representation of an audio mixture (e.g. a recording of several musical instruments), where t is time and f is a frequency of analysis. Let $\hat{X}(t, f)$ be an estimate of the true mixture. We define a spectral basis function $b(f)$ as a function that outputs the relative amplitudes of each frequency. We use spectral basis functions to represent the instantaneous timbre of sound sources, like musical instruments. If the timbre of an instrument is different in different situations (e.g. when a musical instrument plays different pitches) multiple spectral basis functions (one per pitch) can be associated with the instrument.

The MEI framework tries to decompose $\hat{X}(t, f)$ of the audio mixture into a linear combination of spectral basis function:

$$X(t,f) \approx \hat{X}(t,f) = \sum_{j=1}^{J} \sum_{n=1}^{N} g_{n,j}(t) b_{n,j}(f) \qquad (1)$$

where $b_{n,j}(f)$ is the n-th basis for the j-th instrument; $g_{n,j}(t)$ is its gain at frame t. When dealing with harmonic instrument sounds in this paper, each spectral basis function ideally corresponds to a pitch, and the gain represents the activation strength of the pitch.

The signal model used at the SSS stage, where only $g_{n,j}(t)$ are estimated, is the one described in (1). However, the way of getting the basis $b_{n,j}(f)$, which are not computed into the SSS stage, needs a more complex signal model which is summarized here, but it is fully explained in [3].

The multi-excitation model proposed by Carabias et al. in [3] is an extension of the regular excitation-filter model presented in [10]. This model defines the excitation spectrum as a linear combination of a few excitation basis vectors. Under the multi-excitation model, the excitation per pitch and instrument is defined as

$$b_{n,j}(f) = h_j(f) \sum_{m} \left(\sum_{i} w_{i,n,j} v_{i,m,j} \right) G(f - m f_0(n)) \qquad (2)$$

where $m = 1, ..., M$ is the index of the harmonics; and $i = 1, ..., I$ is the index of excitation basis vectors ($I << N$). $v_{i,m,j}$ is the m-th harmonic of the i-th excitation basis vector for instrument j; $w_{i,n,j}$ is the weight of the i-th excitation basis vector for pitch n and instrument j. $G(f - m f_0(n))$ is the window transform placed at the frequency of the m-th harmonic of the n-th pitch.

Given the MEI model, the magnitude spectra of the mixture signal can be decomposed by substituting Eq. (2) into Eq. (1):

$$\hat{X}(t,f) = \sum_{n,m,i,j} g_{n,j}(t) h_j(m f_0(n)) w_{i,n,j} v_{i,m,j} G(f - m f_0(n)) \qquad (3)$$

2.2 NMF parameter estimation

Once the signal model is presented, the way to compute the estimation of each parameter from both the instrument model and the SSS stage should be described. The same NMF framework used in [4] is applied. Given the model in Eq. (3), we want to estimate the parameters so that the reconstruction error between the observed spectrogram $X(t, f)$ and the modeled one $\hat{X}(t, f)$ is minimized. The β-divergence [11, 12] is used here as the cost function to define the reconstruction error, where β is in the range of $[0, 2]$. In [4], a study of the separation performance in function of the parameter β is shown, so that the same $\beta = 1.5$ is used here.

In [13], an iterative algorithm based on multiplicative update rules is proposed to obtain the model parameters that minimize the cost function. Under these rules, $D_\beta(X_t(f) \| \hat{X}_t(f))$ is non-increasing at each iteration and the non-negativity of the bases and the gains is ensured. The multiplicative update rule (see [13] for further details)

for each scalar parameter θ_l is given by expressing the partial derivatives of the $\nabla_{\theta_l} D_\beta$ as the quotient of two positive terms $\nabla^-_{\theta_l} D_\beta$ and $\nabla^+_{\theta_l} D_\beta$:

$$\theta_l \leftarrow \theta_l \frac{\nabla^-_{\theta_l} D_\beta(X(t,f)||\hat{X}(t,f))}{\nabla^+_{\theta_l} D_\beta(X(t,f)||\hat{X}(t,f))}. \quad (4)$$

assuming $\nabla_{\theta_l} D = \nabla^+_{\theta_l} D - \nabla^-_{\theta_l} D$. The main advantage of the multiplicative update rule in Eq. (4) is that non-negativity of the bases and the gains is ensured, resulting in a NMF algorithm.

2.3 Dynamic Time Warping for Score Alignment

In this paper we will use a low complexity signal decomposition method that obtains a distortion matrix for each combination of notes per frame. This matrix is directly used by an online Dynamic Time Warping (DTW) algorithm to obtain the best path without having any information from the future. Consequently, a brief review of DTW is presented below.

DTW is a technique for aligning time series or sequences which has been intensively applied by the speech recognition community [14, 15] and used in many fields such as data mining [16] and information retrieval. The series are represented by 2 vectors of features $U = u_1, ..., u_i, ...u_I$ and $V = v_1, ..., v_j, ..., v_J$ where i and j are the point indices in the time series. I and J represent the length of time series U and V, respectively. As a dynamic programming technique, it divides the problem into several subproblems, each of which contribute in calculating the distance (or cost function) cumulatively.

The first stage in the DTW algorithm is to fill a local distance matrix (a.k.a cost matrix) $\mathbf{D}$ as follows:

$$D(i,j) = \psi(u_i, v_j) \quad (5)$$

where matrix $\mathbf{D}$ has IxJ elements which represent the match cost between every two points in the time series. The cost function ψ could be any cost function that returns cost 0 for a perfect match, and a positive value otherwise (e.g. euclidean distance).

In the second stage (forward step), a warping matrix $\mathbf{C}$ is filled recursively as:

$$C(i,j) = \min \left\{ \begin{array}{c} C(i, j-c_j) + D(i,j) \\ C(i-c_i, j) + D(i,j) \\ C(i-c_i, j-c_j) + \sigma D(i,j) \end{array} \right\} \quad (6)$$

where c_i and c_j are step size at each dimension and range from 1 to α_i and 1 to α_j, respectively. α_i and α_j are the maximum step size at each dimension. Parameter σ controls the bias toward diagonal steps. $C(i,j)$ is the cost of the minimum cost path from $(1,1)$ to (i,j), and $C(1,1) = D(1,1)$.

Finally, in the last stage (traceback step), the minimum cost path $\mathbf{w} = w_1, ..., w_k, ..., w_K$ is obtained by tracing the recursion backwards from $C(I,J)$. Each w_k is an ordered pair (i_k, j_k) such that $(i,j) \in \mathbf{w}$ means that the points u_i and v_j are aligned. The cost of a path $C(\mathbf{w})$ is the sum of the local match costs of the path:

$$C(\mathbf{w}) = \sum_{k=1}^{K} C(i_k, j_k) \quad (7)$$

3. ALIGNMENT METHOD

3.1 States Definition

The aim of this section is to adequately organize the information given by the score to be used for alignment purposes.

First of all, the binary ground-truth transcription matrix $\mathbf{GT}(n,\tau)$ is inferred from the MIDI score, where τ is the time in frames referenced to the score (MIDI time) and n are the notes in MIDI scale. The score defines a consecutive sequence of M states. Each state m is defined by its combination of notes (for all instruments). There are only K unique combination of notes in a score where $K \leq M$ because some states are composed by the same combination of notes.

From the ground-truth transcription matrix $\mathbf{GT}(n,\tau)$, we obtain the following decomposition of binary matrixes

$$\mathbf{GT}(n,\tau) = \mathbf{Q}(n,k)\mathbf{R}(k,\tau) \quad (8)$$

where $\mathbf{Q}(n,k)$ is the notes-to-combination matrix, k the index of each unique combination of notes and $\mathbf{R}(k,\tau)$ represents the activation of each combination in MIDI time. This matrix contains the notes belonging to each combination but no information about MIDI time. Conversely, $\mathbf{R}(k,\tau)$ matrix retains the MIDI time activation per combination but no information about the notes active per combination.

3.2 Spectral Patterns Learning

When a signal frame is given to a DTW-based score follower, the first step should be the computation of a similarity measure between the current frame and the different combinations of notes defined by the score. Our approach is to compute a distortion between the frequency transform of the input and just one spectral pattern per combination of notes. A spectral pattern is here defined as a fixed spectrum which is learned from a signal with certain characteristics. The use of only one spectral pattern per combination allows us to compute the distortions with a low complexity signal decomposition method. This means that our method must learn in advance the spectral pattern associated to each unique combination of notes for the score. To this end, a state-of-the-art supervised method based on NMF with Beta-divergence and Multiplicative Update (MU) rules [17] is used, but in this work, we propose to apply it on synthetic signal generated from the MIDI score instead of the real audio performance.

Let us define a signal model, based on the one proposed in [18], which is the matrix based definition of the one described in (1)

$$\mathbf{Y}(f,\tau) \approx \hat{\mathbf{Y}}(f,\tau) = \mathbf{B}(f,k)\mathbf{G}(k,\tau) \quad (9)$$

where $\mathbf{Y}(f,\tau)$ is the magnitude spectrogram of the synthetic signal, $\hat{\mathbf{Y}}(f,\tau)$ is the estimated spectrogram,

$\mathbf{G}(k,\tau)$ matrix represents the gain of the spectral pattern for combination k at frame τ (i.e. $\mathbf{R}^\top(k,\tau)$) , and $\mathbf{B}(f,k)$ matrix, for $k = 1, ..., K$, represents the spectral patterns for all the combinations of notes defined in the score.

3.3 Online Alignment Stage

3.3.1 Observation model

As explained in section 3.2, the spectral patterns $\mathbf{B}(f,k)$ for the K different combinations of notes are learned in advance using a MIDI synthesizer and kept fixed. Each spectral pattern models the spectrum of a unique combination.

Now, the aim is to compute the gain matrix $\mathbf{G}(k,t)$ and the cost matrix $\mathbf{D}(\tau,t)$ that measures the suitability of each combination of notes belonging to each MIDI time τ to be active at each frame t (referenced to the input signal) by analyzing the likelihood between the spectral patterns $\mathbf{B}(f,k)$ and the input signal spectrogram [1] . Similar approach was used by Fritsch and Plumbey in [19], but they use one component per instrument and note plus some extra-components to model the residual sounds.

From the cost matrix $\mathbf{D}(\tau,t)$, a classical DTW approach can be applied to compute the alignment path. The computation procedure to obtain the values of $\mathbf{D}(\tau,t)$ is not included here because of space limitations, but it is fully explained at [18].

3.3.2 Path Computation

We use here a DTW based method to perform the alignment using the cost matrix $\mathbf{D}(\tau,t)$ obtained in section 3.3.1. This cost matrix is computed from the input signal $\mathbf{X}(f,t)$ and the "synthetic" spectral patterns per combination $\mathbf{B}(f,k)$ explained in section 3.2. The term "synthetic" comes from the fact that the spectral patterns $\mathbf{B}(f,k)$ are computed from the score using a MIDI synthesizer.

Details about the implementation of this method are given in Section 2.3. Although classic DTW method achieves the lowest computational cost, it remains offline, so that it is not suitable for partially unknown series.

As in [18], the online algorithm differs from the standard DTW algorithm in some points. Firstly, the signal is partially unknown (or the future of the signal is not known when making the alignment decisions), so the global path constraints cannot be directly implemented, in other words, the recursion backwards can not be traced from the last frame T of the signal. Secondly, the algorithm has a limitation in its anti-causal response. The latency of the algorithm (the delay in the decision) must be limited to a maximum value. Consequently, an incremental solution is required. The recursion backwards can be traced in equally spaced frames of the input signal making the latency equal to the difference in time of the frame when the backtracking is done and the input signal frame. Finally, in order to run in realtime, the complete algorithm should not increase the complexity with the length of the signal.

[1] Note that we are using $\mathbf{X}$ and t instead of $\mathbf{Y}$ and τ to represent the signal magnitude spectrogram and the time frames to distinguish between real world and synthetic signals.

Here, we propose an online algorithm with a fixed latency of just one frame. In order to obtain this low latency, no backtracking is allowed, taking the decision directly from the forward information at each frame t.

4. SEPARATION METHOD

Here the NMF factorization framework is composed of two parameters, the gains $g_{n,j}(t)$ and the spectral patterns $b_{n,j}(f)$, as described in Eq. (1), but it now includes more than one instrument, specified by index j. Algorithm 1 shows an overview of the separation system with fixed instrument models.

Algorithm 1 Separation Algorithm with fixed instrument models

1 Compute $X(t,f)$ from a the audio mixture to separation.
2 Initialise gains $g_{n,j}(t)$ with the ground-truth pitch transcription and the basis functions $b_{n,j}(f)$ with the previously trained ones.
3 **for** C iterations **do**
4 Update gains $g_{n,j}(t)$.
5 **end for**

The MIDI score-aligned information is used to initialize the gains $g_{n,j}(t)$ for the NMF factorization. A random positive value is given when the aligned MIDI score indicates that the corresponding pitch indexes of instrument j are active at frame t. The gains associated to non-active pitches of the instrument j at frame t are set to zero.

Once the gains $g_{n,j}(t)$ are initialised, and using the trained instrument models $b_{n,j}(f)$, an iterative algorithm is run as shown in Algorithm 1. This algorithm iteratively updates $g_{n,j}(t)$ according to Eq. (4), while keeping $b_{n,j}(f)$ fixed. Once the gains are estimated, the separated signals are computed using Wiener masks as described in [4].

Although the NMF factorization framework is intrinsically offline, when spectral patterns $b_{n,j}(f)$ are fixed, the only parameters to be updated are the gains $g_{n,j}(t)$. When using the updating equation, the gains at frame t just depend on the input signal $X(t,f)$ at frame t and, consequently, the algorithm becomes online.

5. EXPERIMENTS

5.1 Training and testing data

At the training stage (see Section 2.1), the spectral basis functions are estimated using the RWC musical instrument sound database [20]. Four instruments are studied in the experiments (violin, clarinet, tenor saxophone and bassoon). From the RWC database we select the files with one playing style (normal) and one dynamic level (mezzo), since this is the configuration used in the majority of published results.

The database proposed in [2] is used for the testing stage. The ground-truth alignment between MIDI and audio was interpolated from annotated beat times of the audio. The

annotated beats were verified by a musician through playing back the audio together with these beats as explained in [2].

5.2 Experimental set-up

5.2.1 Model parameters

The frame size and the hop size for the STFT are set to 128 ms and 32 ms respectively. Also, $C = 50$ iterations for the NMF-based algorithms is used.

In relation to the pitch resolution, in the training stage a pitch resolution of a semitone (the same as the training database) is used. In the separation stage, the learned basis functions $b_{n,j}(f)$ are adapted to a 1/8 semitone resolution in pitch by replicating the function for each semitone 8 times. Real instruments produce pitches not only at idealized semitones, due to pitch variations such as vibrato. With a 1/8 semitone resolution in pitch, we can better capture the pitch variation of real instruments. Here, $\beta = 1.5$ has been used as demonstrated in [4].

5.2.2 Testing metrics

For an objective evaluation of the source separation performance of the proposed method, we use the metrics implemented in [21] (BSS EVAL Toolbox 2.1). These metrics are commonly accepted by the research community in source separation, and therefore facilitate a fair evaluation of the method.

For the score alignment results, the MIREX [2] metrics, which are the most used among the community, have been used.

5.3 Results

The proposed system has two main stages: Score Following and Source Separation. Firstly, the online alignment method is assessed in front of the offline version and other state-of-the-art proposal. Then, both alignment methods are tested into a common SSS framework.

5.3.1 Alignment method results

Figure 1 shows the performance of the proposed online Score Following method. It is compared to its corresponding offline version, Duan&Pardo [2] (non-DTW-based state-of-the-art method) and Oracle which shows the maximum accuracy from the ground-truth data. Oracle is not a score following system but the provided aligned MIDI score assuming constant tempo and perfect synchronization between musicians. It represents the evolution of the accuracy for each method while the correct aligned decision threshold moves from $50ms$ to $2000ms$.

This test details that the ground-truth cannot be used as itself unless the considered threshold grows more than $200ms$. This is because this data has been obtained by a common alignment between all the independent instrument signals from the data base. Regarding this limitation,

[2] `http://www.music-ir.org/mirex/wiki/2015:Main_Page`

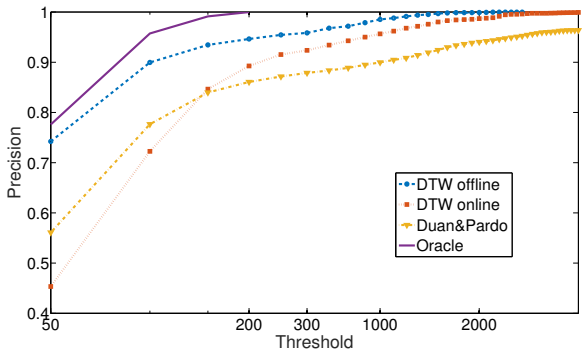


Figure 1. *States-based Offline DTW with tempo constraint path estimation*

Method	SDR	SIR	SAR
Oracle	16.54 dB	24.04 dB	17.32 dB
Proposed	12.85 dB	21.45 dB	14.67 dB
Duan&Pardo	11.94 dB	19.27 dB	13.11 dB

Table 1. SSS results with both alignment method and the Oracle ones. Audible examples are available at *http://anclas3.ujaen.es/onlineSSS/*

the proposed online method outperforms the state-of-the-art one. Besides, it has a minimum distance from the offline version, concerning that it only uses the forward side of the DTW algorithm.

5.3.2 Source Separation results

The aligned score from the proposed method and the one from [2] has also been tested over the Source Separation task. The Oracle signals for SSS are obtained by using the analysis/synthesis filter bank over the isolated signal, so that filter bank effects are avoided from the comparison. Here, both data sets have been used for initializing the gains matrixes as described in section 4 based on [4]. Then, results of each alignment method with a common SSS stage are shown in Table 1.

SSS results show the same tendency as in section 5.3.1. The proposed alignment method overcomes the state-of-the-art one with almost the equivalent difference as in section 5.3.1. Besides, the distance from Oracle results is not so big despite the alignment mistakes. This means that the SSS algorithm is using properly instrument models which are not causing so much distortion at the output signals.

6. CONCLUSIONS

The proposed alignment method has provided better results at the alignment assesment than the state-of-the art one. These results are also reflected at the SSS test where a better alignment provides better separation results. However, it can be seen that here the differences are lower. That could be caused because the lack or precision of the alignment stage would affect the whole note assessment in the alignment test but only a small note are (near the note onset) in the SSS test.

Acknowledgments

This work was supported by the Andalusian Business, Science and Innovation Council under project P2010- TIC-6762 and (FEDER) the Spanish Ministry of Economy and Competitiveness under Projects TEC2012-38142- C04-01, TEC2012-38142- C04-03 and TEC2012-38142- C04-04.

7. REFERENCES

[1] F. Rodriguez-Serrano, J. Carabias-Orti, P. Vera-Candeas, F. Canadas-Quesada, and N. Ruiz-Reyes, "Monophonic constrained non-negative sparse coding using instrument models for audio separation and transcription of monophonic source-based polyphonic mixtures," *Multimedia Tools and Applications*, vol. 72, no. 1, pp. 925–949, Apr. 2013.

[2] Z. Duan and B. Pardo, "Soundprism: An online system for score-informed source separation of music audio," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 6, pp. 1205–1215, 2011.

[3] J. Carabias-Orti, T. Virtanen, P. Vera-Candeas, N. Ruiz-Reyes, and F. Canadas-Quesada, "Musical Instrument Sound Multi-Excitation Model for Non-Negative Spectrogram Factorization," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 6, pp. 1144–1158, 2011.

[4] F. Rodriguez Serrano, Z. Duan, P. Vera-Candeas, B. Pardo, and J. J. Carabias-Orti, "Online Score-Informed Source Separation with Adaptive Instrument Models," *Journal of New Music Research*, 2015.

[5] Z. Duan, G. Mysore, and P. Smaragdis, "Online PLCA for real-time semi-supervised source separation," in *Proc Int Conf on Latent Variable Analysis and Signal Separation (LVA/ICA 2012)*, Apr. 2012, pp. 34–41.

[6] C. Joder, F. Weninger, F. Eyben, D. Virette, and B. Schuller, "Real-time speech separation by semi-supervised nonnegative matrix factorization." in *Proc Int Conf on Latent Variable Analysis and Signal Separation (LVA/ICA 2012)*, Apr. 2012, pp. 322–329.

[7] L. Simon and E. Vincent, "A general framework for online audio source separation," in *Proc Int Conf on Latent Variable Analysis and Signal Separation (LVA/ICA 2012)*, Apr. 2012, pp. 397–404.

[8] S. Dixon, "Live tracking of musical performances using on-line time warping," in *Proc. Int. Conf. on Digital Audio Effects (DAFx)*, Madrid, Apr. 2005, pp. 92–97.

[9] A. Cont, "A Coupled Duration-Focused Architecture for Real-Time Music-to-Score Alignment," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 6, pp. 974–987, Jun. 2010.

[10] T. Virtanen and A. Klapuri, "Analysis of polyphonic audio using source-filter model and non-negative matrix factorization," *Advances in Models for Acoustic Processing, Neural Information Processing Systems Workshop*, 2006.

[11] E. Vincent, N. Bertin, and R. Badeau, "Adaptive Harmonic Spectral Decomposition for Multiple Pitch Estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, 2010.

[12] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the beta-divergence," *Neural Computation*, vol. 23, Mar. 2011.

[13] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, pp. 556–562, 2001.

[14] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 23, no. 1, pp. 67–72, 1975.

[15] L. Rabiner and A. Rosenberg, "Considerations in dynamic time warping algorithms for discrete word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, pp. 575–582, 1978.

[16] D. Berndt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series." *KDD workshop*, pp. 359–370, 1994.

[17] J. Carabias-Orti, M. Cobos, P. Vera-Candeas, and F. Rodriguez-Serrano, "Nonnegative signal factorization with learnt instrument models for sound source separation in close-microphone recordings," *EURASIP Journal on Advances in Signal Processing*, vol. 1, Apr. 2015.

[18] J. Carabias-Orti, F. Rodriguez-Serrano, P. Vera-Candeas, F. Canadas-Quesada, and N. Ruiz-Reyes, "An audio to score alignment framework using factorization and Dynamic Time Warping," in *Proc 16th Int Society for Music Information Retrieval Conf (ISMIR)*, 2015.

[19] J. Fritsch and M. Plumbley, "Score informed audio source separation using constrained nonnegative matrix factorization and score synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2013*, May 2013, pp. 888–891.

[20] M. Goto, "Development of the RWC Music Database," in *Proceedings of the 18 th International Congress on Acoustics (ICA 2004*, 2004, pp. 553–556.

[21] E. Vincent, "Musical source separation using timefrequency source priors," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 91–98, 2006.

MULTI-CHANNEL SPATIAL SONIFICATION OF CHINOOK SALMON MIGRATION PATTERNS IN THE SNAKE RIVER WATERSHED

Ben Luca Robertson
Eastern Washington University,
Department of Music
Cheney, Washington – United States
benlrobertson@
aphoniarecordings.com

Dr. Jonathan Middleton
Eastern Washington University,
Department of Music
Cheney, Washington – United States
jmiddleton@ewu.edu

Jens Hegg
University of Idaho,
Water Resources Program
Moscow, Idaho – United States
hegg1432@vandals.uidaho
.edu

ABSTRACT

Spatialisation, pitch assignment, and timbral variation are three methods that can improve the perception of complex data in both an artistic and analytical context. This multi-modal approach to sonification has been applied to fish movement data with the dual goals of providing an aural representation for an artistic sound installation, as well as a qualitative data analysis tool useful to scientists studying salmon migration. Using field data collected from three wild Chinook Salmon (*Oncorhynchus tshawytscha)* living in the Snake River Watershed, this paper will demonstrate how sonification offers new perspectives for interpreting migration patterns, including the potential to display the impact of environmental factors on the lifecycle associated with this species. Within this model, audio synthesis parameters guiding spatialisation, microtonal pitch organization, and temporal structure are assigned to streams of data through software applications developed for the project. Collection and interpretation of field data was performed in partnership with the University of Idaho – Water Resources Program.

INTRODUCTION

Clearly and meaningfully presenting complex datasets within auditory displays is a problem shared by those creating musical compositions from data and those endeavoring to use sonification as an analytical tool. The sonification of multiple data streams is complicated by the difficulty in conveying the complexity of the data in sound space [1]. Spatialisation and timbral variation are two methods that can improve the perception of complex data in an artistic and analytical context; both of which are active areas of study within the sonification field [2]. Working within a cross-disciplinary team, we have utilized both scientific and musical perspectives to create meaningful sonifications of complex, temporal data of salmon movement. As such, consultation with the Water Resources Program has directly informed the choices and tuning of key sonification parameters.

Copyright: © 2015 Ben Luca Robertson, et al. 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.

The sonification of Chinook salmon migration data from the Snake River watershed to the Pacific Ocean is a multi-year project that is currently in the proof-of-concept stage of development. The project's primary goal is to track the life history of juvenile salmon (fry to smolt) via sonified out-migrations. The timing and duration of these movements are influenced by environmental factors experienced by individual fish, such as water temperature and food resources. Recent large changes in this migration timing may indicate adaptation to changes in the habitat of this species from human impacts such as dams and land use change. Subtle differences in movement timing can be difficult to easily discern graphically or statistically. Working in a multi-disciplinary team we seek to use data-to-sound auditory display in the dual role of artistic auditory installation and as an analytical aid for fish and wildlife scientists.

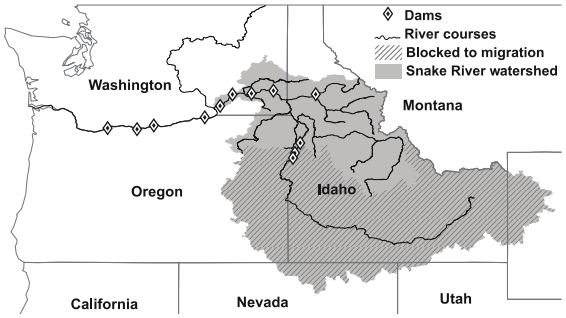


Figure 1. Snake River Watershed

1. OTOLITH STRUCTURE AND FUNCTION

Analysis of otoliths collected from wild salmon has played a fundamental role in tracing the geographic movements of these fish. Otoliths are balance and hearing organs in bony fishes (cartilaginous fish, such as sharks and rays do not possess them) [3]. These organs are roughly analogous in function to the human inner ear and are used for hearing, as well as orientation. The otolith is located below the brain in fluid-filled sacs and includes a pointed structure, called the rostrum, attached to nerves that sense the motion of the organ within this fluid. As the fish moves through the water, the otolith moves within the surrounding fluid relative to the fish's orientation to gravity. In structure, the otolith is not truly a bone; instead it is a calcium carbonate deposit

that accretes a new layer of calcium carbonate each day. These layers accumulate much like tree rings, wider during periods of fast growth and narrower when growth is low. This mineral is deposited as amorphous, crystalline aragonite. Aragonite is one of three forms of calcium carbonate, including calcite and vaterite. Also, like the rings of a tree, chemical signatures found in otoliths can reveal important details about the lives of these fish [3].

Because calcium and strontium are close on the periodic table, strontium (and other elements such as barium, manganese and magnesium) occasionally substitutes for calcium in a crystal structure. Since these isotopes and elements are unique for many rivers, it is possible to determine where a fish has traveled in fresh water by recovering the chemical and isotopic signatures of these substituted elements from within the otolith. This data is combined from measurements made on two different mass spectrometer devices; one that measures elemental ratios (weight of individual atoms of an element as a ratio with calcium) and another that measures isotopes ratios (the difference in the number of neutrons between atoms of the same element) [4].

2. SONIFICATION PARAMETERS

2.1 Temporal Framework for Otolith Measurement

Measurement and incremental assessment of the otolith of each fish in the study yields important details of its lifecycle. Not unlike reading the growth rings within a tree, measuring the distance from the center of the otolith indicates the age of the fish [4]. By statistically relating the distance from the otolith core (measured in microns, 1/1000 of one millimeter) with known chemical signatures from rivers in the study area, it is possible to determine the location and timing of life events; such as birth, maturation, migration patterns, and death. A significant finding of our project includes the practical translation of measurements associated with physical mediums, such as the otolith, into a temporal function inherent to sound generation.

For the purposes of auditory display, micron data is transformed into temporal parameters, establishing a rate at which aural events unfold. After multiple trials, the current sonification model utilizes a time rate of 50 milliseconds (ms.) per micron measured within each otolith. Three salmon were tracked in this study: one female (#5133) and two males (#2693 and #3206). These fish were chosen, in consultation with the University of Idaho – Water Resources Program, as representative data samples from the environment and as preliminary models for future sonification applications that could include a wider sampling of wild salmon. The choice of time scaling works to simultaneously display the entire lifecycles for each of the three fish within a manageable time period for observation. As the otolith of the fish with the greatest longevity (#3206) was measured at 2337 microns, the entire duration of playback for the display is 116,850 milliseconds, or ~117 seconds. Initially a playback rate of 100 milliseconds per micron was chosen. However, early tests conducted at the University of Idaho concluded this slower rate of playback to be less effective in tracking meaningful transitions in location and chemical signature. Further trials are certainly needed to optimize playback rates for meaningful auditory display.

Sonification linking the physical structure of otoliths to the inherently temporal nature of both the medium and the subjects' lifecycle is further accomplished by correlating the maturation period and overall longevity of each fish to changes in overall amplitude. Accordingly, these changes in amplitude signify the most significant life events for the subjects. In turn, each of the fish displayed is assigned a unique amplitude envelope. Important temporal markers, including birth, completion of the maturation period, and death**,** act as breakpoints within these overlapping envelopes (see Figure 2).

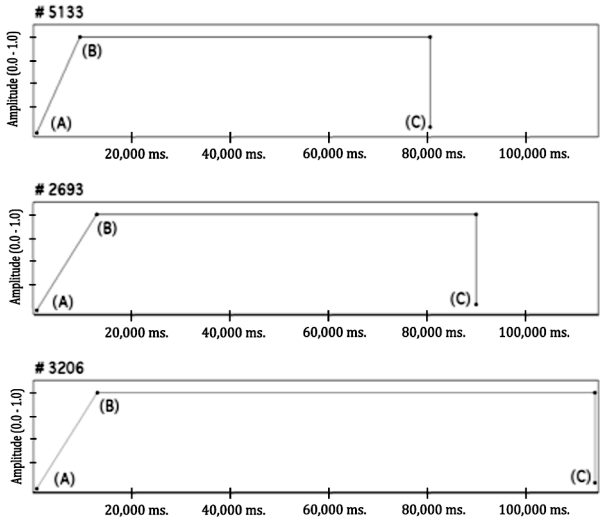


Figure 2. Simultaneous Amplitude Envelopes
[A = Birth] [B = End of Maturation] [C = Death]

Birth and death stages are represented by an overall amplitude value of 0.0. The maturation period is heard as a linear crescendo beginning with an amplitude value of 0.0 and ending with a sustained amplitude value of 1.0. For example, in both males (#2693 and #3206), a maturation period of just over 250 microns encompasses the center of the otolith, while the female matures within 181 microns. In turn, these measurements are represented by crescendos with durations of 125,500 and 9,050 milliseconds, respectively. Thereafter, death is punctuated by a sudden decrescendo into silence.

During simultaneous playback, changes in cumulative amplitude generated by these envelopes function as a coarse, auditory indication to the listener of how many fish are currently active within a watershed or marine system at a given time. Furthermore, the rate of crescendo (amplitude gain versus time in milliseconds) informs the listener of the rate of maturation for individuals or groups of fish.

2.2 Spatialisation and Strontium Isotopes ($^{87}Sr/^{86}Sr$)

Just as time may be measured as a function of physical growth, past locations for each salmon can be indicated by the presence of a specific range of strontium isotopic ratio ($^{87}Sr/^{86}Sr$) values sampled from the otolith. The ability to approximate location from this data is owed to the fact that isotopes of strontium are not recognized by the organism's cells. Therefore, the chemical signature of the water through which the fish migrates is the same as the value recorded in the otolith. This signature originates from the decay of rocks in the watershed and is unique to each environment. Thus, the strontium isotopic ratio can be used to link a fish's location to a specific watershed [5].

Application of strontium data is used to define spatialisation within the listening environment. Since this sonification model may be presented as a public installation, the program is written so that sound sources representing data from groups or individual fish can be distributed across a four-speaker (quadraphonic) sound system. Within this model, strontium isotope values indicative of the presence of fish in the Lower Snake River (LSK) are represented by the assignment of corresponding aural materials to the left-front (LF) loudspeaker (in relation to observers), sound materials indicative of data from the Clearwater River System (CWS) are assigned to the right-front (RF) loudspeaker, indications of fish in the Upper Snake River (USK) to the right-rear loudspeaker (RR), and data suggesting the arrival of fish in the Pacific Ocean signified by sound in the left-rear (LR) speaker. For stereo playback (i.e. headphones) applications, LSK and Ocean data will sound in the Left Channel, while CWS and USK data will sound in the Right Channel.

It should be noted that quadraphonic diffusion is far from optimal in terms of modeling continuous motion between loudspeakers. However, in the project's current proof-of-concept stage, spatial modeling of data is being treated as primarily, discrete sound sources. The majority of strontium isotope data associated with the LSK, USK, and CWS is displayed according to a range in which an average of these values collected from a given watershed is assigned to a specific speaker (LF, RF, LR, RR) and aural materials associated with an individual or group of fish are assigned accordingly. This component of the auditory display is key to identifying migration behaviors within separate water systems. Meanwhile, transitional values between these averages are expressed as motion between speaker channels. As such, the panning algorithm is entirely linear, whereas the mean value within a given crossover range of strontium data is assigned a 50/50 division of amplitude between two respective loudspeaker channels.

Sonifying results connected to migration patterns into the Pacific Ocean requires further insight into characteristics of strontium isotope distribution, universally present in otolith samples associated with a marine environment. For example, samples connected with time spent in the Pacific Ocean represent the stabilization of the strontium isotope ratio at 0.70918, a marine signature consistent throughout the world's oceans [6]. The presence of this data necessitates a less linear approach to sonification. Accordingly, the audio software is written so that hard panning to the LR speaker (associated with migration to the Pacific Ocean) only occurs when the strontium isotope values consistently match a mean value of 0.70918 over the course of eight, consecutive micron samples. To efficiently process the large quantities of data collected, samples are taken every three microns. Because the mass spectrometer used in collecting chemical data measures a small quantity of atoms at a given time, a great deal of variation in the strontium isotope signature is present. Consequently, this data is averaged using a 20-point rolling average over time in order to display meaningful results [4].

Referencing only strontium isotope signatures, it is also worth noting that a great deal of crossover between chemical signatures is present in fish migrating through multiple locales. For example, certain strontium isotope signatures associated with the CWS were also measured in fish inhabiting the LSK. Likewise, similar values collected from fish migrating through the LSK were also present within otolith measurements from fish known to be present in the USK (see Table 1 and Figure 3).

Location:	Strontium Isotope ($^{87}Sr/^{86}Sr$):	Loudspeaker:
LSK	0.709080 < 0.711560	LF
CWS	0.710927 < 0.713022	RF
Pacific	0.708684 < 0.710072	LR
USK	0.708983 < 0.709584	RR

Table 1. Strontium Isotopic Ratio ($^{87}Sr/^{86}Sr$) Range

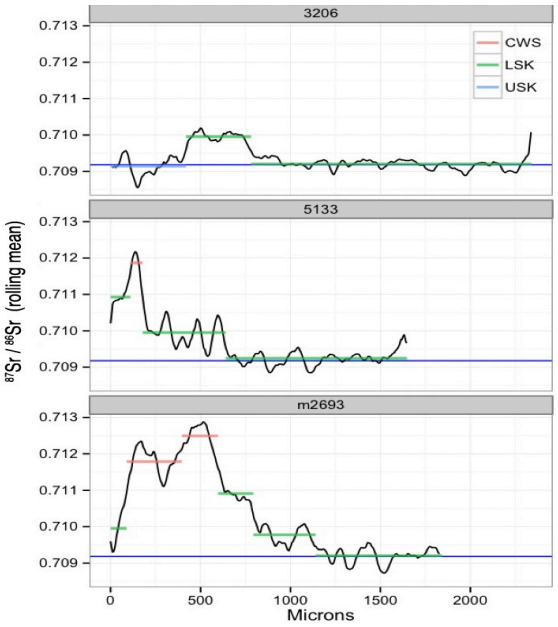


Figure 3. Strontium Isotopic Ratio ($^{87}Sr/^{86}Sr$) Range in Three Otolith Samples

Insight into this variability can be gained by understanding that chemical signatures present in the otolith at early ages are informed by the signature inherited from the mother. This signature is heavily influenced by the marine environment. As the egg grows within the mother while she is still in the ocean, the chemistry reflects this trait. Young fish are nourished by a yolk sac derived from the egg until they grow large enough to emerge from the gravel and feed on their own. Consequently, the first ~250 microns of $^{87}Sr/^{86}Sr$ data are not exact representations of the small fish's location [4].

Consequently, while strontium isotope signatures indicate general patterns of migration, this data alone does not necessarily pinpoint the exact location of a fish within its lifecycle. Creating an accurate auditory display that addresses both chemicals signatures and known location requires a multi-modal approach encompassing the inclusion of other discrete forms of data assignment performed in concert with spatialisation.

2.3 Pitch Change and Geographic Location

To address disparities between data from known locations and locations suggested by strontium isotope data, a multi-modal sonification technique employing both spatialisation and momentary pitch change is utilized. While spatialisation via four-channel diffusion intimates the motion in migration patterns, early tests at the University of Idaho indicate that panning alone does not adequately identify the exact moment of entry into a given water system. Consequently, instantaneous pitch change has been employed to give a clear and unmistakable indication of entrance into specific water systems. As such, a specific pitch is assigned to each system. Using conditional data algorithms programmed in Max/MSP (i.e. "if, then, else" expressions), changes in pitch occur when incoming micron values match the first measurement taken at a specific site, as confirmed by field data. For example, fish #2693 enters the CWS at 252 microns. Thus, at 12,600 milliseconds into playback (252 microns at a playback rate of 50 milliseconds per micron unit), the pitch for this stream of data changes from 275 Hertz to 385 Hertz (See Table 2).

	Maternal	**CWS**	**USK**	**LSK**	**Ocean**
Pitch	110 Hz	275 Hz	330 Hz	385 Hz	605 Hz
	f * (1/1)	f * (5/2)	f * (6/2)	f * (7/2)	f * (11/2)
f = 110 Hz	A1 + 0 cents	C#3 - 14 cents	E3 + 2 cents	G3 - 31 cents	D4 + 51 cents
5133	0 - 9050 ms			9050 - 32,000 ms	32,001 - 82,500 ms
	0 - 181 Microns			181 - 641 Microns	641 - 1645 Microns
2693	0 - 12,600 ms	12,600 - 32,000 ms		32,000 - 57,100 ms	57,100 - 91,950 ms.
	0 - 252 Microns	252 - 600 Microns		600 - 1142 Microns	1142 - 1839 Microns
3206	0 - 12,700 ms		12,700 - 21,100 ms	21,100 - 39,050 ms	39,050 - 116,850 ms
	0 - 254 Microns		254 - 422 Microns	422 - 781 Microns	782 - 2337 Microns

Table 2. Pitch Assignment By Location

Pitch assignment for each location is intended to aid the listener in making clear distinctions between environments encountered during migration. The central frequency range spans just under four and one-half octaves; with the "Maternal" and "Ocean" signatures occupying the lower and upper bounds at 110 Hertz and 605 Hertz, respectively. All three fish sampled pass through these two systems, as well as the LSK. Consequently, an easily distinguishable application of range was a conscious choice in programming this component of the auditory display.

Likewise, the decision to include non-tempered, or *just*, intervals to represent both discreet stages in migration and concurrent geographic relationships between groups of fish is informed by this necessity. Utilizing the Maternal Signature assignment of 110 Hertz as the fundamental frequency (*f*), octave transpositions of the 5th, 6th, 7th, and 11th partials in the overtone series yield the just intervals, 5/2, 6/2, 7/2, and 11/2. These just intervals correspond with entry into the CWS, USK, LSK, and Ocean, respectively. While concurrent pitch relationships between the Maternal Signature, CWS, and USK reference, familiar – albeit, justly tuned – triadic harmonies, inclusion of pitch assignments for the LSK and Ocean introduce less-familiar, microtonal intervals. As these two regions are encountered by all three fish throughout the study, the ability to aurally identify change and convergence is vital for data interpretation. In turn, distinctive intervallic choices, such as the *septimal* minor third sounding between data streams from fish #5133 and #3206 at 254 microns, are aimed at making these momentary distinctions unmistakable, as well as lending unique harmonic colouration. (See Figure 4 to view these changes in the form of a musical score).

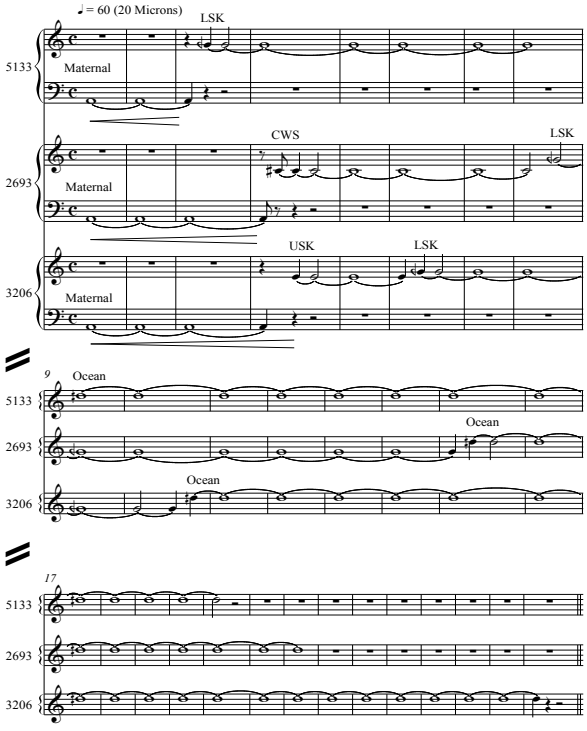


Figure 4. Musical Score for Pitch Assignments

An emergent property of assigning pitch values to location data includes cumulative changes in amplitude. As such, crescendos and decrescendos result from greater or fewer fish occupying a given location. Through analysis of the entire spectra produced in playback, three fish assigned to 605 Hertz create a greater spectral peak at 605 Hertz than a single fish would; thus indicating a larger number of fish present in the corresponding location, the Pacific Ocean.

Another component of the lifecycle suggested – yet only coarsely touched upon through spatialisation – is the maturation period, or "maternal signature". Within this range, chemical signatures, such as strontium isotope ($^{87}Sr/^{86}Sr$) and Strontium/Calcium ratios (Sr/Ca), exhibit chemical traits reflecting the chemistry of the mother [4]. Higher chemical signatures from this time in each young salmon's life are consistently presented within the first 250 microns of otolith samples. These high signatures, retained from the mother, can likely be attributed to her travels through the Columbia River, a habitat with similarly high signatures [3]. Prior to evidence of entry into the CWS, USK, or LSK systems, sounding of a slow crescendo with a central pitch of 110 Hertz indicates the presence of this maternal signature during playback. Again, punctuation of this component of the lifecycle is displayed using discrete changes in pitch working in concert with other aural dynamics.

2.4 Variations in Timbre and Strontium/Calcium (Sr/Ca) Ratio Intensity

In addition to strontium isotope signatures, the intensity of strontium/calcium (Sr/Ca) ratios collected from otoliths offer another useful chemical indicator of location. Sharp increases in Sr/Ca intensity are particularly evident upon entry into a marine environment [3]. In fact, the Water Resources Program's choice to include Sr/Ca data as a key sonification parameter – as opposed to barium or magnesium signatures, also present in otolith samples – reflects the particular importance of calcium as a primary indicator of marine migration. As indicated by initial tests at the University of Idaho, sonifying transitions to the Pacific Ocean requires a multi-modal approach to sound generation that extends beyond simple, spatial diffusion. Just as changes in pitch are used in concert with panning to punctuate changes in location, variation in timbre is also linked to spatialisation. In terms of chemical indicators (on-site field observations aside), concurrent stabilization of $^{87}Sr/^{86}Sr$ signatures to a mean value of 0.70918 and sharp peaks in Sr/Ca intensity are evident when migration into the marine environment is likely. These peaks are generally observed when the Sr/Ca intensity exceeds a value of 1.0. An average $^{87}Sr/^{86}Sr$ range of $0.7085 < 0.7098$ retains a mean value of approximately 0.70918, the stabilized $^{87}Sr/^{86}Sr$ intensity indicative of entry into a marine environment. Though the entire continuum of strontium data measured from all otolith sections affected by the Pacific Ocean encompasses a wider range of $0.708684 < 0.710072$, measurements from the periphery of this data are applied as transitional breakpoints during spatialisation. From this data, a conditional algorithm determines when audio signals representing migration patterns of an individual are to be assigned to the Left-Rear (LR) speaker channel; thus indicating entry into the Pacific Ocean. This conditional expression reads as follows:

If $[0.7085 < {}^{87}Sr/{}^{86}Sr < 0.7098]$ *&* $[Sr/Ca > 1.0]$*, then hard-pan signal to Left-Rear (LR) speaker channel.* (1)

In contrast, the evolution of timbre and associated parameters are derived solely from Sr/Ca intensity values. Initially, the team explored the use of additive resynthesis techniques to address each chemical signature as a separate, spectral component. However, the lack of perceptual traceability – particularly by those observing from a biological discipline – led the group to simply its approach. Ultimately, a modified form of amplitude modulation (AM) synthesis using a unipolar, random-amplitude carrier waveform ("rand~" object in Max/MSP) and a sine waveform modulator was chosen. The frequency of the modulating waveform (f_{mod}) is controlled by pitch assignments from each location signature (Maternal = 110 Hz; CWS = 275 Hz; USK = 330 Hz; LSK = 385 Hz; Ocean = 605 Hz). This value determines the center frequency of the resulting AM waveform. As discussed, these pitch values vary discretely, according to entry into each water system. The carrier waveform generates random amplitude values (between -1.0 and 1.0) at a specified rate (f_{rand}). This rate is derived from scaling Sr/Ca intensity data. According to data collected, the range of Sr/Ca intensity values for the marine environment vary between 0.947882 (Sr/Ca_{min}) and 2.55 (Sr/Ca_{max}). These values are scaled in a linear fashion, to an output range between 50 Hz ($f_{randmin}$) and 800 Hz ($f_{randmax}$).

$$f(x) = c\,[1 - (x - a) / (b - a)\,] + d\,[(x - a) / (b - c)]$$

Whereas:

$f(x)$ = Carrier Frequency (f_{rand})	x = Sr/Ca Intensity
a = 0.947882 (Sr/Ca_{min})	c = 50 Hz ($f_{randmin}$)
b = 2.55 (Sr/Ca_{max})	d = 800 Hz ($f_{randmax}$)

(2)

As this modified AM synthesis model utilizes a unipolar, random-amplitude carrier waveform, amplitude output is divided in half and then increased by a value of 0.5 to constrain signal values to a range of 0.0 to 1.0. The use of a unipolar carrier distinguishes this algorithm from most traditional forms of AM synthesis, which generally utilize a unipolar, modulating waveform [7].

$$[rand\,(2\,\pi\,f_{rand}) * 0.5 + 0.5] * [cos\,(2\,\pi\,f_{mod} + \phi)]$$

Whereas:
rand = Random amplitude values (0.0 < 1.0)
f_{rand} *= Carrier Frequency (noise bandwidth)*
f_{mod} *= 110 Hz, 275 Hz, 385 Hz, or 605 Hz*

(3)

In preliminary tests, identification of changes aural characteristics were readily noted by the team from the University of Idaho – Water Resources Program, suggesting the saliency of this technique and the range of synthesis parameters chosen. As the carrier frequency is increased, the noise bandwidth surrounding the center frequency (f_{mod}) is broadened and recognition of a perceived pitch is obscured. In contrast, Sr/Ca intensities below 0.947882 reflect migration through freshwater environments and generate timbres that retain a more sinusoidal, or *pure*, tone. On a figurative level, the increased noise bandwidth resulting from higher Sr/Ca intensity (indicative of entrance into the ocean) has been described by scientific team members as eliciting aural sensations associated with the wind and surf (i.e. "noisy" or "washy" timbre) of a marine environment. From a sound design perspective, the associative qualities and simplicity of the modified AM synthesis algorithm speak to its potential in conveying the character of environments encountered during out-migration.

CONCLUSIONS

Utilizing continuous timbral variation, microtonal pitch organization, site-specific spatialisation, and the translation of physical measurement into temporal structure, this project has sought to create clear and useful auditory displays of salmon movements gleaned from the unique chemistry of otoliths. This data, being inherently temporal, is an ideal data source for sonification. However, its complexity makes statistical and graphical analysis of subtle movements difficult. Therefore, our goal is to eventually create a comprehensive sonification tool that allows scientists to explore these complex datasets more easily. Further validation of observations made in the development process could certainly aid in honing the auditory display. Likewise, we await additional input from other members of the scientific community as to the total scope and potential applications for auditory display in the evaluation of environmental impacts on salmon migration. As such, a future stage in the project could include the administration of perceptual testing with other fish and wildlife scientists from beyond the University of Idaho to address the effectiveness of sonification techniques, specifically the efficacy of spatialisation and timbral variation in conveying complex data.

From the perspective of the composers and sound artists working on the project, the role of salmon migrations in the Pacific Northwest and the richness of the available data are compelling in their musical merit. Looking forward, our immediate plans for presentation of this auditory display include a public installation in Spokane, Washington. At the moment, and in consideration of the current scale of sites addressed in the study (CWS, USK, LSK, & Pacific Ocean), real-time diffusion via four loudspeakers has been employed. However, additional locations added to the study and inclusion of a more comprehensive diffusion system of eight or more channels could certainly benefit presentation of such data. The accumulation of otolith signatures from more subjects, in addition to data from the first three fish already displayed, could also add a dimension of richness and density; both in terms of aesthetic and scientific potential. One could feasibly envision microtonal clusters derived from data of twenty or more fish enveloping listeners.

In regard to process and creative development, working in a multi-disciplinary team has yielded important insights that could be otherwise overlooked operating in a single field. The dual concerns of artistic engagement and maintaining scientific cogency seem best addressed via a multi-modal approach. Exploring a means of sharing the unique lifecycle and potential human impacts on the Chinook salmon, an emblematic species of the greater Pacific Northwest, certainly warrants the careful attention, innovation, and cooperation from both scientific and artistic disciplines.

Acknowledgements

This research was supported by Tekes - The Finnish Funding Agency for Innovation (decision 40296/14).

REFERENCES

[1] Hermann, T., Hunt, A., & Neuhoff, J. G. (2011). *The Sonification Handbook*. Logos-Verlag.

[2] Kramer, G., Bonebright, T., & Flowers, J. H. (2010). *Sonification Report: Status of the Field and Research Agenda, Report prepared for the National Science Foundation by members of the International Community for Auditory Display*.

[3] Campana, S. E., & Thorrold, S. R. (2001). "Otoliths, increments, and elements: keys to a comprehensive understanding of fish populations?" *Canadian Journal of Fisheries and Aquatic Sciences*, *58*(1), 30–38.

[4] Hegg, J. C., Kennedy, B. P., Chittaro, P. M., & Z bel, R. W. (2013). Spatial structuring of an evolving life-history strategy under altered environmental conditions. *Oecologia*, *172*(4), 1017–1029. doi:10.1007/s00442-012-2564-9

[5] Hegg, J. C., Kennedy, B. P., & Fremier, A. K. (2013). "Predicting strontium isotope variation and fish location with bedrock geology: Understanding the effects of geologic heterogeneity", *Chemical Geology*, *360-361*, 89–98. doi:10.1016/ j.chemgeo.2013.10.010

[6] Faure, G., & Mensing, T. M. (2004). *Isotopes: principles and applications*. John Wiley & Sons Inc.

[7] Roads, C. (1996). "Modulation Synthesis," in *The Computer Music Tutorial*, fifth ed. Cambridge, MA: MIT Press, pp. 215-226.

Mapping brain signals to music via executable graphs

Katie Crowley
Trinity College Dublin
k.crowley@tcd.ie

James McDermott
University College Dublin
jmmcd@jmmcd.net

ABSTRACT

A method for generating music via a mapping from brain signals is proposed. The brain signals are recorded using consumer-level brain-computer interface equipment. Each time-step in the signal is passed through a directed acyclic graph whose nodes execute simple numerical manipulations. Certain nodes also output MIDI commands, leading to patterned MIDI output. Some interesting music is obtained, and desirable system properties are demonstrated: the music is responsive to changes in input, and a single input signal passed through different graphs leads to similarly-structured outputs.

1. INTRODUCTION

Mappings between sensory modalities are fascinating. Synaesthesia is an example: some people report experiencing certain colours when they hear certain pitches, for example; others report an association between colours and letters [1]. One of the authors recently heard a two-year-old child refer to some intense crayon work as "doing loud on the paper". Feeling the kick drum in your chest is indispensable to some forms of music. Jean-Michel Jarre and many others have made mappings between light and music. Douglas Hofstadter poses questions like, what would a poem be like if it were in the medium of painting instead [2] – and recreational drug users sometimes report answers. In our favourite songs, we often feel that there is an essential link between the words and the music – not just that they are well-suited, but that they are synchronised, with a clear mapping between them at each point in time. Something similar happens with film scores.

The voltages that are produced by the human brain as a by-product of its normal activity are not a sensory modality similar to, say, sight or hearing. Nor are they a modality we have obvious control over, like speech. However, a mapping between the brain's activity and the resulting voltage signals can be established [3]. It is therefore of interest to think about mappings from these signals to other modalities. Because these signals are time series, it is particularly natural to consider mappings to a time-based medium like music. The fact that a feedback loop is possible – brain to signal to music to ear to brain – greatly increases the possibilities and the interest.

Copyright: ©2015 Katie Crowley 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.

In the long term, we hope to use mappings and feedback between brain signals and music for forms of music therapy [4, 5]. In this initial study the goal is much less ambitious: it is to map brain signals to engaging, listenable music which is synchronised with the brain signal and reflects changes in it. It is thus a form of sonification. Success in these initial steps is required for the longer-term goal.

The output of such a mapping will depend on both the input and the mapping itself. However, the goal is to achieve a sort of separation of control between the two. The mapping should be capable of achieving a somewhat listenable, if dull "steady state" of music in response to a completely static input, but should also be responsive to changes in the input. Similarly, a single signal mapped through different mappings should give results which, if not really similar in style or content, are similar in temporal structure.

2. BACKGROUND & PREVIOUS WORK

2.1 Brain-Computer Interfaces

The human brain is made up of billions of neurons, which emit electrical impulses and changes in hemodynamics when interacting. The electrical impulses form a measurable voltage on the scalp that can be detected by electroencephalogram (EEG) devices. A Brain-Computer Interface (BCI) is a system which measures changes in this voltage in real-time [6,7]. Typically the raw EEG signals are pre-processed to produce a usable time-series or in some cases a command output. BCIs have applications in human computer interaction.

Modern BCIs can be non-invasive, portable, low-cost, and easy to use, with high temporal resolution. The cheapest ones may use just one or two EEG sensors fitted to a light-weight headset, in contrast to medical grade BCIs.

2.2 BCI Music

Research into BCIs for music is a growing area with potential in artistic, scientific, recreational and therapeutic fields. The earliest reported example of EEG-based musical analysis was in *Brain* in 1934 [8, cited by Miranda [9]], however, it is generally accepted that EEG-based composition began with Lucier's *Music for Solo Performer*, a percussive piece composed by the performer wearing an EEG cap. Teitelbaum used various physiological signals including EEG and electrocardiogram (ECG) to control electronic synthesisers [10]. Rosenboom also examined the use of EEG signals to generate art, including music, and developed EEG-based musical interfaces [11]. Rosenboom introduced a musical system whose parameters were driven

by EEG signals associated with changes in the performer's selective attention [12]. Interested readers may refer to Williams [13] for a comprehensive review of the history of BCI music.

Affective Algorithmic Composition (AAC) is a proposed umbrella term [13] referring to an interdisciplinary field which combines computer-aided composition with affect analysis (or emotion assessment). AAC algorithms are driven by an intended affective response from the listener, who in turn, can become the composer. AAC includes any system for composition designed to respond to an affective target and/or to create an affective response in the listener [13]. A listener might use a bio-signal device to measure some physiological response, for example, to generate affectively responsive music.

Existing work uses brain-computer control systems to allow users to control musical parameters via EEG [14–16]. Miranda et al. [4] describe the evaluation of a pilot brain-computer musical interface allowing a patient with Locked-in syndrome to control amplitude and other musical parameters via EEG for the purposes of music therapy and palliative care [4]. Advances in modern BCIs and non-clinical EEG provide an opportunity to develop more commercially-accessible neurofeedback-derived control over musical features in response to individual affective responses resulting in real-time biophysical sensing of emotions to control AAC systems.

2.3 Other Mappings

Moving away from BCI, one important stream of research in mapping signals to music has arisen in the context of evolutionary computation (EC). EC is a class of population-based metaheuristic search and optimisation algorithms inspired by Darwinian evolution. EC approaches to music usually take advantage of some form of mapping rather than trying to create music directly. An interesting mapping was proposed by Hoover [17]. Time is divided into time-steps. At each time-step, some variables are fed into a neural network. They represent the events in the corresponding time-step of some pre-existing music. The neural network maps these values to produce multiple outputs, which can be interpreted as MIDI commands. By running the network once per time-step, with the input signals varying over time, the result is a new piece of music synchronised with – because it is created as a mapping from – the input piece. Naturally, the mapping must be of sufficient complexity that the output is not a simple monotonic transformation of the input.

This method was used to interactively create drum tracks to accompany pre-existing harmonic and melodic material [17]. The network was trained through interactive evolution, that is via preferences for one network's results over another's, expressed interactively by a listener. An appealing feature of the representation is that in a neural network, computation is "shared" – the same result calculated at one node can be re-used by multiple nodes at the next layer, and so the multiple outputs can be expected to be related. Of course, that is a desirable property for the multiple voices of many types of music.

The same idea was later extended to create pitched accompaniment material, and to also use simple signals indicating the "semantics" of the current time-step – whether it is the start of a beat, and whether it is the start of a bar [18]. "Complex conductors", i.e. arbitrary time series as further input variables, were also proposed.

Inspired by these mappings, the second author has developed [19] a directed acyclic "executable graph" representation which again uses input variables representing the time-step's "semantics", shared computation in the graph, and multiple outputs mapping to MIDI commands. The main differences from NEAT Drummer and subsequent work are: (1) the model of computation does not use an implicit weighted sum of inbound edges at each node. The arity of the function executed by a node determines the number of inbound edges that it requires. (2) No input music is used. (3) The output nodes are *stateful*, that is their inputs and outputs in previous time-steps can affect their outputs in the current time-step. In another implementation, trees (rather than graphs) are used, and input signals are supplied by the user via a mouse or Nintendo Wiimote [20].

These representations are capable of generating listenable music, at least over short time-scales. It is natural to consider using them as general methods for sonifying any type of time series. That is the point we take up in this paper. We propose a mapping and investigate how well it achieves our goals:

- A static BCI input signal should lead to listenable (if dull) steady-state music;
- The music should respond to changes in the input signal;
- The temporal structure of the input signal should be reflected in the output;
- As a consequence, a signal mapped through different graphs should lead to pieces of music which share temporal structure.

We begin by describing more details of the mapping, including novel features not used in previous work.

3. MUSIC WITH EXECUTABLE GRAPHS

3.1 Music as a function of time

The representation is a development of that in *XG* [19], as inspired by that of *NEAT Drummer* [17]. Time is divided into even time-steps, e.g. six steps per quarter-note. At each time-step, the values of some numerical variables (described later) are fed into a graph. The nodes of the graph may output MIDI note-on or note-off messages. In this way, temporal patterns in the input variables give rise, via a mapping, to temporal patterns in the output. In this representation we can think of music as a function of time, and of the input time-series.

3.2 Model of computation

The graph is directed and acyclic. Input nodes carry the input BCI signals. Other nodes have incoming edges and carry out numerical computations such as +, *, or *sin*. The graph is constrained to have the right number of inbound edges to each function. On the other hand, each node may

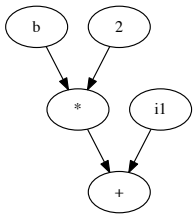
Table 1: Labels, computations, and arities for all node types.

label	result	arity
i_1	input signal 1 at current time-step	0
i_2	input signal 2 at current time-step	0
b	beat at current time-step (an integer)	0
0.5	constant 0.5	0
1	constant 1	0
2	constant 2	0
unary-	$-x$	1
*	$x * y$	2
+	$x + y$	2
-	$x - y$	2
pdiv	$x/\sqrt{1+y^2}$	2
pmod	$x \,\%\, \sqrt{1+y^2}$	2
sin	$\sin(x)$	1
cos	$\cos(x)$	1
if	if $x \geq 0.0$ then y else z	3

send its output to any number of other nodes. In contrast to a neural network, there is no implicit weighted sum of inbound edges' signals.

Because the graph is constrained to be acyclic, the nodes can be sorted using "topological sort", i.e. nodes which have no inputs are placed first, and every node is placed after the nodes which give its inputs. Thus each node can be executed in this order, to execute the entire graph. The graph is thus *executable*. The labels, computations, and arities for all node types are shown in Table 1.

As an example, Fig. 1 shows a very simple graph and its effect. Just one input signal i_1 is used, together with the beat signal b. The values of these signals over six successive time-steps (two bars in 3/4 time) are shown, together with the output of the $+$ node. We can interpret these outputs as pitch values for MIDI note-on commands. (This is a simplified example, with a very small graph, just one input signal, and ignoring the effects of restrictions on output nodes, accumulators and thresholds, and sigmoid and diatonic mappings, to be described in detail below.)



time-step	0	1	2	3	4	5
b	0	1	2	0	1	2
i_1	50	50	50	58	58	58
output (at $+$)	50	52	54	58	60	62
pitch	D	E	F#	A#	C	D'

Figure 1: A simplified example of the mapping process.

3.3 Graph generation

The graph generator starts by creating one node each with the labels `i1`, `i2`, `b`, `0.5`, `1`, and `2`, i.e. all those with arity 0. It then adds 100 nodes, each with a label randomly-chosen from those with non-zero arity. For each node, it adds the appropriate number of inputs (according to arity), taken from the output of any previously-added node. In this way, the property of acyclicity is also guaranteed.

3.4 Output nodes: accumulators and thresholds

The graph as described so far deals with purely numerical values. In order to produce music, these numerical values must be mapped to MIDI note-on/note-off commands. This takes place at output nodes. An output node is just a normal node of the graph, with a label chosen from Table 1 as usual. However only a node with at least two inputs, and such that there is at least one path from an input node to this node of length 3 or greater, will be used as an output node. This multiple-output representation is reminiscent of that used in *single-node genetic programming* [21].

An output node's inputs are used for a numerical computation, determined by its label, and resulting in a numerical output as usual. In addition the inputs are interpreted as pitch and activity controls. An output node has an accumulator variable which is increased by the value of the activity control, via a sigmoid mapping, at each time-step. Whenever the activity is above a numerical threshold (set to 1.25 in experiments reported here), two things happen: a MIDI note-on command is output, with pitch controlled by the pitch control input, via a sigmoid mapping and a diatonic mapping, and with velocity controlled by the degree to which activity exceeds the threshold; and the accumulator variable is decreased to account for this command. Whenever the activity is below a second, lower threshold (0.0625), a MIDI note-off command is issued, for the pitch most recently switched on. Thus pitch, velocity, and note duration are explicitly controlled. Whenever the activity is between these two thresholds, there is no MIDI output, but the activity variable is decreased.

The sigmoid mapping is standard, $x \rightarrow 1/(1+e^{-x})$. The motivation for the mapping in both pitch and activity is that the output of a node can vary widely, especially with multiplication and division. The mapping "squashes" large values (positive or negative).

These computations lead to quite human-sounding variations in note volume and note density. Although it is still fully deterministic, it tends to avoid the metronomic or robotic feeling that can easily arise in generated music, e.g. to some extent in the output of previous work [20]. The individual voices in the music play and rest and form phrases with pleasant dynamics.

The restriction on input path length for output nodes helps to prevent very simple (e.g. monotonic) transformations of the input from occurring in the output.

There are several important parameters in the representation, including the accumulator threshold and the number of nodes in the graph. The values given above for these parameters have been found to give good results, but investigation of optimal values is postponed to future work.

4. BCI HARDWARE AND PROCESSING

An EEG signal is a voltage that is measured on the surface of the scalp, arising from neural activity e.g. mental state, cognitive activity etc. Fluctuations in the EEG signal occur within defined frequency bands that have been associated with brain states such as attention (Beta: 13–30Hz), engagement, frustration, meditation (Alpha: 8–13Hz) and so on. Changes in the signal within these frequencies bands can be measured by EEG devices, which reflect changes in neural activity. Some of these bands relate to emotion-based responses, and concentrating on these frequencies, we can capture emotional response data.

4.1 NeuroSky MindWave

NeuroSky Technologies have developed a minimally invasive, dry biosensor to read neural activity representing states of attention (Beta) and meditation/relaxation (Alpha). The *MindWave* headset consists of a single dry sensor positioned at the forehead on a position known as FP1, to capture activity from the pre-frontal cortex in the front of the brain where higher thinking occurs. Emotions, mental states, concentration, etc. are all dominant in this area. The MindWave captures raw neural signals at FP1 and provides information on a user's Alpha, Beta, Gamma, Delta and Theta bands. The signals are captured at 512Hz, filtered and processed using a Fourier transform, and passed to a proprietary algorithm which generates *eSense* values, custom measures of attention and meditation [22]. For each of attention and meditation, the algorithm returns one value per second on a scale from 0 to 100, representing the level of attention or meditation of the subject [23].

In a previous study, Crowley et al. [24] identified threshold values for these eSense scales in order to categorise response intensity. Using these threshold measures, an eSense value between 40 to 60 at any given moment in time is considered "neutral". A value from 60 to 80 is considered "slightly elevated", and values from 80 to 100 are considered "elevated". Similarly, on the other end of the scale, a value between 20 to 40 indicates "reduced" level of response, while a value between 1 to 20 indicates "strongly lowered" levels.

The meditation value returned by the headset is used to record the users' state of arousal, which indicates the level of a user's mental "calmness" or "relaxation". If the user is relaxed and not under stress then the value returned is high (high meditation = low stress). The eSense Attention meter indicates the intensity of a user's level of mental "focus" or "attention", such as that which occurs during intense concentration and directed (but stable) mental activity. The attention value captures the users' level of effort. If the user's effort level is high then the output can near 100 whereas if they make no effort at all it is nearer 0 [24]. While the headset records both the raw EEG signal and the eSense measures, our analysis focuses on the custom attention and meditation scales for their potential as easy-to-use, "off-the-shelf" measures of EEG signal activity that could be used by signal processing novices.

4.2 BCI Data Collection and Preparation

To produce the input BCI data for the system a number of tasks were completed. The aim was to use both baseline and task-related BCI data as inputs for the system. A subject was fitted with the NeuroSky MindWave device and asked to complete a number of tasks while wearing the BCI headset. Firstly, the subject was asked to sit quietly for 5 minutes while baseline recordings were measured. Three *stressor* tasks were then administered – The Towers of Hanoi, an N-Back Task and an electric wire loop game. These are common in psychological and BCI research as described, e.g. by Crowley [24, 25]. These are not musical tasks, hence the system is functioning as a sonification of the BCI data rather than a method for the subject to control music.

The three stressor tasks produced BCI data that varied in attention and meditation levels from baseline. Each task is designed to elicit varying degrees of stress (low meditation) and require different amounts of cognitive load (attention) depending on the individual response of the participant. The eSense meters of *attention* and *meditation* for each task were extracted from the BCI recordings and used as inputs i_1 and i_2 to the executable graph.

Several composite signals were created, with the goal of imposing clear temporal structure:

ABA means A signal of 48s in ABA format, where A uses the mean value of the subject's baseline recording for 16s, and B uses the mean value of the subject's Towers of Hanoi recording (lowered meditation and raised attention) for 16s.

ABA non-means A similar signal in ABA format, but using 16s of raw signal from the baseline and Towers of Hanoi recordings, rather than means.

ABACADA non-means A similar signal in ABACADA format, where C and D are raw signal from the N-Back and Wire Loop tasks (both tasks again leading to lowered meditation and raised attention).

5. RESULTS

The executable graph mapping was used to generate many pieces using various BCI signals. Here we concentrate on pieces made using two different graphs, and using the composite signals described above. The "ABA means" signals were used to demonstrate that a static input signal leads to a static musical pattern. The simplicity of the output then made it suitable for use during auditioning of multiple graphs. We chose two graphs which led to interesting patterns, corresponding to random seeds numbers 1 and 5. The latter graph is shown in Fig. 2. For the former we chose a minor scale mapping [1] and for the latter, a major scale.

The "ABA non-means" signals were then used to investigate the result of non-static input signals. The results were encouraging: the non-static input signals introduce variation, but not so much that the piece loses a sense of close similarity with the "means" version.

[1] Refer to the subject2_means_aba_seed1 mp3 available for download.

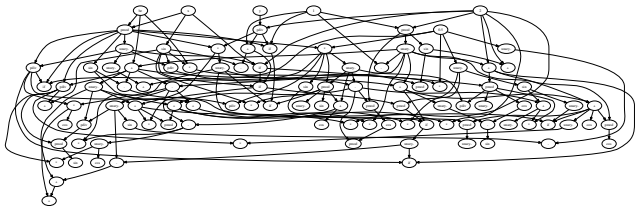


Figure 2: One of our chosen graphs. Due to its size (100 nodes) the labels are not readable here, but the graph is available online.

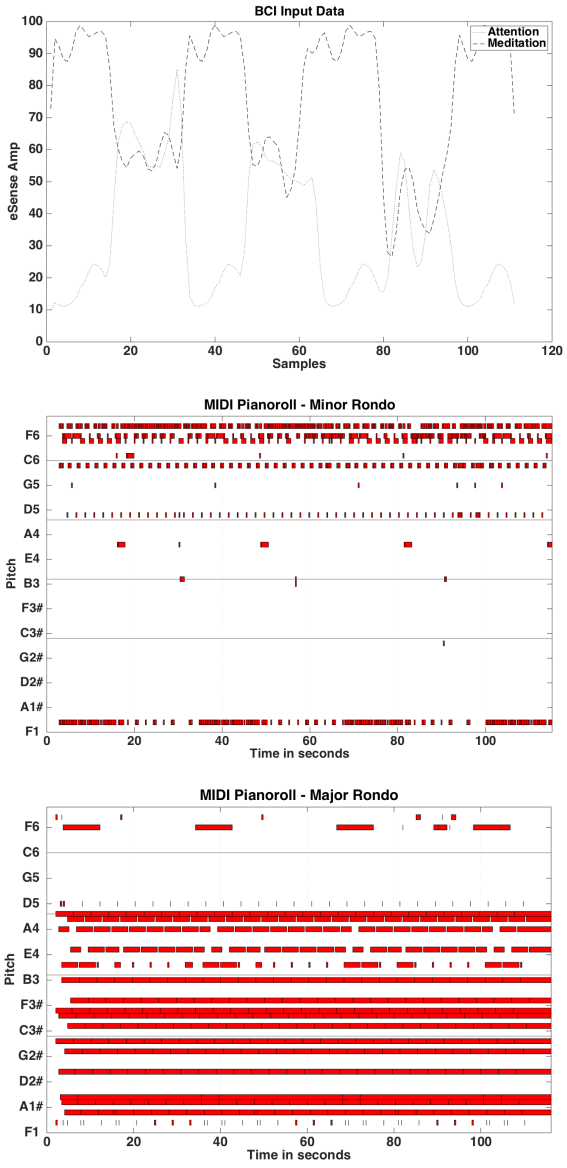


Figure 3: A BCI input (with two signals) in *ABACADA* form and two distinct pieces of music, with the same temporal structure, resulting from mapping this input via two distinct graphs.

Finally, we moved to the "ABACADA non-means" signals, still using the same graphs for mapping. Figure 3 shows a MIDI pianoroll for the pieces generated with these signals, along with the composite BCI signal (attention and meditation eSense scales)[2]. Each piece is a rondo with the form *ABACADA*. In both pieces, the MIDI pianoroll shows clear repeated themes that are in sync with changes in the attention eSense meter. Increases in the attention level of the subject has a direct impact on thematic variations in the piece. Similarly, decreases in meditation (increased stress) also shapes the melody of the piece. The minor piece shows the impact of decreased meditation on the MIDI output. Both pieces share a similar temporal structure, even though the graphs used are entirely different. Thus, we have achieved a sort of separation of control between the graph (responsible for musical material) and the BCI input (responsible for temporal structure).

In previous iterations of this work evolutionary computation was used to search for good graphs. We have found that search is not necessary in this iteration, since the graph generator used to make initial graphs seems to give multiple "good" pieces out of every 5 generated. The pieces described above are using random seeds 1 and 5, where auditioning began at seed 0.

The pieces described here are available together with code, composite BCI signals, and a small collection of other pieces with low-numbered seeds, from
http://www.skynet.ie/~jmmcd/xg.html.

6. DISCUSSION & FUTURE WORK

We have succeeded in our initial steps: our representation can map BCI signals to music. It is responsive to changes in the signal, but not so responsive that changes in the signal lead to unrecognisable music. Static input signals lead to interesting musical patterns in a significant proportion of randomly-generated graphs, while the addition of variation in the input signals can lead to quite good "miniature" musical pieces. These statements are subjective, of course. One necessary step for future work is an objective validation.

The next phase of this project will then involve using the eSense meters in real-time. We will then have a feedback loop in our signal path: from the brain via the BCI, the graph, and the music, to the ear, and thence the brain.

Other issues to be investigated include: more fine-grained input data signals, rather than the two summary signals output by the eSense meters; the algorithm's sensitivity to parameters mentioned in Section 3.4; and the use of headsets for controlling an evolutionary search based on attention to multiple pieces of music in a population. The categorisation thresholds identified by [24] will be used to determine the success level of the generations, resulting in adaptive feedback composition. Multiple headsets will then allow collaborative composition.

7. REFERENCES

[1] O. Sacks, *Musicophilia: Tales of music and the brain*. New-York, Vintage Books, 2008.

[2] D. R. Hofstadter, *Le ton beau de Marot: In praise of the music of language*. Basic Books New York, 1997.

[3] A. W. Toga and J. C. Mazziotta, *Brain mapping: the methods*. Academic press, 2002.

[2] Refer to the subject2_nonmeans_abacada mp3s available for download.

[4] E. R. Miranda, W. L. Magee, J. J. Wilson, J. Eaton, and R. Palaniappan, "Brain-computer music interfacing (BCMI) from basic research to the real world of special needs," *Music and Medicine*, vol. 3, no. 3, pp. 134–140, 2011.

[5] A. Kirke, E. R. Miranda, and S. Nasuto, "Learning to make feelings: Expressive performance as a part of a machine learning tool for sound-based emotion therapy and control," in *Cross-Disciplinary Perspectives on Expressive Performance Workshop*, 2012.

[6] G. G. Molina, T. Tsoneva, and A. Nijholt, "Emotional Brain-Computer Interfaces," in *Affective Computing and Intelligent Interaction and Workshops*, 2009, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5349478&isnumber=5349257

[7] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[8] E. D. Adrian and B. H. C. Matthews, "The Berger rhythm: Potential changes from the occipital lobes in man," *Brain*, vol. 57, no. 4, 1934.

[9] E. R. Miranda, "Brain-computer music interface for composition and performance," *International Journal on Disability and Human Development*, vol. 5, no. 2, pp. 119–126, 2006.

[10] R. Teitelbaum, "In tune: Some early experiments in biofeedback music (1966-1974)," *Biofeedback and the Arts, Results of Early Experiments. Vancouver: Aesthetic Research Center of Canada Publications*, 1976.

[11] D. Rosenboom, *Biofeedback and the arts, results of early experiments*. Not Avail, 1976.

[12] ——, "The performing brain," *Computer Music Journal*, pp. 48–66, 1990.

[13] D. Williams, A. Kirke, E. R. Miranda, E. Roesch, I. Daly, and S. Nasuto, "Investigating affect in algorithmic composition systems," *Psychology of Music*, p. 0305735614543282, 2014.

[14] J. Eaton and E. Miranda, "Real-time notation using brainwave control," in *Sound and music computing conference (SMC)*, 2013.

[15] E. Miranda and A. Brouse, "Toward direct brain-computer musical interfaces," in *Proceedings of the 2005 conference on New interfaces for musical expression*. National University of Singapore, 2005, pp. 216–219.

[16] E. R. Miranda, K. Sharman, K. Kilborn, and A. Duncan, "On harnessing the electroencephalogram for the musical braincap," *Computer Music Journal*, vol. 27, no. 2, pp. 80–102, 2003.

[17] A. K. Hoover, M. P. Rosario, and K. O. Stanley, "Scaffolding for interactively evolving novel drum tracks for existing songs," in *Proceedings of EvoWorkshops*, ser. LNCS, vol. 4974. Springer, 2008, p. 412.

[18] A. K. Hoover and K. O. Stanley, "Exploiting functional relationships in musical composition," *Connection Science*, vol. 21, no. 2, pp. 227–251, 2009.

[19] J. McDermott and U.-M. O'Reilly, "An executable graph representation for evolutionary generative music," in *GECCO '11*, Dublin, 2011.

[20] J. Shao, J. McDermott, M. O'Neill, and A. Brabazon, "JIVE: A generative, interactive, virtual, evolutionary music system," in *EvoMUSART: Proceedings of EvoWorkshops*. Springer, 2010.

[21] D. Jackson, "A new, node-focused model for genetic programming," in *Proceedings of the 15th European Conference on Genetic Programming, EuroGP 2012*, ser. LNCS, A. Moraglio, S. Silva, K. Krawiec, P. Machado, and C. Cotta, Eds., vol. 7244. Malaga, Spain: Springer Verlag, 11-13 Apr. 2012, pp. 49–60.

[22] NeuroSky, "NeuroSky's eSense Meters and Detection of Mental State," NeuroSky, White Paper, September 2009.

[23] G. Rebolledo-Mendez, I. Dunwell, E. A. Martínez-Mirón, M. D. Vargas-Cerdán, S. de Freitas, F. Liarokapis, and A. R. García-Gaona, "Assessing neurosky's usability to detect attention levels in an assessment exercise," in *Human-Computer Interaction: New Trends*, ser. Lecture Notes in Computer Science, J. A. Jacko, Ed. Springer Berlin Heidelberg, 2009, vol. 5610, pp. 149–158. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02574-7_17

[24] K. Crowley, A. Sliney, I. Pitt, and D. Murphy, "Evaluating a Brain-Computer Interface to Categorise Human Emotional Response," in *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*, July 2010, pp. 276 –278.

[25] ——, "Capturing and using emotion-based bci signals in experiments: how subject's effort can influence results," in *Proceedings of the 25th BCS Conference on Human-Computer Interaction*. British Computer Society, 2011, pp. 132–138.

ANALYSIS OF MUSICAL TEXTURES PLAYED ON THE GUITAR BY MEANS OF REAL-TIME EXTRACTION OF MID-LEVEL DESCRIPTORS

Sérgio Freire
School of Music
Federal University of Minas Gerais
(UFMG)
sfreire@musica.ufmg.br

Pedro Cambraia
School of Engineering
(UFMG)
cidorage@gmail.com

ABSTRACT

The paper presents a set of mid-level descriptors for the analysis of musical textures played on the guitar, divided in six categories: global, guitar-specific, rhythm, pitch, amplitude and spectrum descriptors. The employed system is based on an acoustic fretted nylon string guitar with hexaphonic pick-ups, and was programmed in Max. An overview of the explored low-level audio descriptors is given in the first section. Mid-level descriptors, many of them based on a general affordance of the guitar, are the subject of the central section. Finally, some distinctive characteristics of five different textures – two-voice writing, block chords, arpeggios, fast gestures with legato, slow melody with accompaniment– are highlighted with the help of the implemented tools.

1. INTRODUCTION

The development of mid-level acoustic descriptors represents the ongoing stage of a project started a few years ago, based on an acoustic nylon string guitar equipped with hexaphonic pickups, and implemented in real-time in Max. Previous efforts were dedicated to the analysis of different guitar techniques, such as tremolos, block chords, voice/layer balance, and rhythmic phrasing [1, 2].

The main low-level features of guitar sounds may be adequately represented by scalars (single values) associated with one event (a note); this fact makes easier the planning and implementation of mid-level descriptors, since it is not necessary to deal with the comparison of low-level data curves of different lengths. We believe that the combination of a few mid-level descriptors is able to establish a consistent description of different musical segments played on the guitar, what may be useful for later comparisons, and may even lead to some kind of categorization. The computation of these descriptors is made on pre-established segments, which can be worked out by manual or automatic segmentation processes. Despite the existence of segmentation strategies, the excerpts analyzed in this paper were segmented by hand, since they represent typical textures on guitar, extracted from the well-established repertoire of the instrument. In these cases, the presence of a score facilitates the interpretation of the results and the evaluation of the descriptors' adequacy.

The text is divided in three sections. Firstly, we discuss the low-level descriptors in use, and some sonic features not covered by them. The second section presents and discusses the implementation of mid-level descriptors, which were based on a – somehow hypothetical – general affordance of the guitar, avoiding any particular bias towards specific musical styles. The descriptors were divided in different categories that should not be interpreted as totally independent from each other: global, guitar-specific, rhythm, pitch, amplitude and spectrum descriptors. Finally, we present some musical examples and their corresponding mid-level descriptors. Distinctive values for each chosen texture are discussed. Although the small number of excerpts prevents any kind of generalization, the preliminary results encourage their use in performance analyses. Besides, the implemented mid-level descriptors represent a consistent base for the next stages of the research: expansion of the database, creation of new descriptors, real-time analysis, comparison and classification of musical segments, comparative analysis of a sequence of segments. Upon the latter features, we intend to build a tool for the practice of interactive improvisational music, based on Rowe's player paradigm [3]. A virtual partner, provided with a dedicated nylon guitar samples library, should be able to mimic, and propose variations on the textures played by the guitarist [1]. The mid-level descriptors will be also explored in spatialization routines and in interaction with gestural descriptors.

2. LOW-LEVEL DESCRIPTORS

Every note event, within a specific musical segment, is defined by the following parameters: event number; time of the event (onset or offset); number of the string in use; fundamental frequency; amplitude; spectral centroid; slur flag; presence of bend or vibrato.

The time of the occurrence of the event, measured from the beginning of the segment, is expressed in milliseconds. The detection of onsets and offsets are described in [1],

Copyright: ©2015 Sérgio Freire 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] At this writing, the system is based on a Spanish acoustic guitar Alhambra and LR Baggs pickups. Two features of this hardware –the significant influence of one string on the remaining ones, and a remarkable timbral difference between the acoustic sound and the picked up signal– have pushed us towards a more symbolic approach, instead of using the direct audio as the starting material for interactive processes.

within an error margin of 10 ms. Concerning the string choice, it is worth remembering that the guitar strings are numbered from 1 to 6, from high to low.

For the extraction of the fundamental frequencies and its variations we use the object *fiddle*, created by M. Puckette [4]. The fundamental frequency is expressed in Midi notes, quantized to integers. The exact tuning of the instrument is a pre-requisite to a correct extraction of frequency deviations, such as vibratos and bends, which are expressed with floating point values. Note that we do not use fiddle's own attack detection; when our system detects an onset, it waits for a short period (ca. 50–60 ms) before requesting a reliable fundamental frequency. Some onsets are detected without a definite frequency. In these cases, they are named non-pitched events, and receive a very high fixed value (120). The same values calculated on the onset for the fundamental frequency are stored to be used in the corresponding offset event. The offset threshold is set to -74 dB.

The amplitude estimation was already described in previous works, where we detected a dynamic range lying around 35 and 40 dBs. The offsets are marked with a zero value. For the estimation of the spectral centroid, we use the object centroid, developed by T. Apel, J. Puterbaugh and D. Zicarelli. As the guitar notes do not present a steady spectrum, we choose the highest value occurring just after the onset detection. For the pitched events, we use a relative value for the centroid (centroid/fundamental); for the non-pitched events we set an absolute value in Hz.

In the acoustic guitar practice, slurring refers to the hammer-on and pull-off techniques, while *glissando* is performed by means of sliding a finger along one string. This does not produce a real glissando as in fretless instruments, but, instead, a sudden change in the frequency happens when the finger surpasses a fret. The slur flag is set when a new fundamental frequency is extracted without the detection of a new attack (or onset), what is mostly likely to occur with the use of the techniques just described.

We have not yet developed a complete tool for dealing with frequency variations occurring after the note onset. So far, the system is able to detect whether the fundamental frequency played on one string has suffered a variation beyond certain threshold once (bend) or more than two times (vibrato). The value of 8 cents has been used as the threshold, and is supported by experimental data collected in our lab [5]. It is necessary to limit the frequency variations to a whole itone, otherwise variations due to resonances may be wrongly interpreted as vibrato or bend data. This information is stored by the offset event, since the vibrato or bend may happen anytime after the onset.

Our system still lacks tools dedicated to the handling of harmonics and sympathetic resonances – two very pregnant features of the guitar timbre. Besides, the percussive use of the guitar body also deserves a dedicated study.

In the near future, we also plan a systematic study of the accuracy of the system, dedicated to the low-level descriptors in use. This will be done on the Alhambra guitar, already mentioned in the paper, and also on a recent addition to the project: an instrument by Yamaha, equipped with RMC pickups. The use of two instruments will provide an opportunity not only to refine the current parameters, but also to test the dependence of their settings on specific hardware configurations. Based on our previous experiments, we may say that the accuracy of the system depends not only on its physical components and algorithms, but also on the expertise of the players and on the musical content; all these variables must be taken in account for a fair evaluation of the system. For this paper, we had to correct up to 8% of the played notes in the worst case.

3. MID-LEVEL DESCRIPTORS

So far, the implemented mid-level descriptors may be divided into six categories: global, guitar-specific, rhythm, pitch, amplitude and spectrum descriptors.

3.1 Global Descriptors

The global descriptors are in number of three: duration, density of notes and 1/3-quantiles.

The duration of the segment is expressed in seconds, being measured between the first onset and the last offset. The density of notes is given by the ratio between the number of onsets and the duration. The 1/3-quantile descriptor expresses some basic temporal information about the distribution of onsets throughout the segment. The total duration of the segment is divided in three parts, and for each part we calculate the number of onsets. The descriptor's output comprises two values, one for the weight of the number of onsets in the first part, the other for the cumulative weight of the onsets in the first and second parts.

3.2 Guitar-specific Descriptors

3.2.1 Number of Used Strings – String "Centroid"

The number of used strings needs no further explanation; it is sufficient to remember that we are not taking in account the sympathetic resonances in the present study. We also calculate a vector with the number of onsets in each string. Based on this vector, it is possible to calculate a string "centroid", taking the weighted average of each item in the vector. To improve the selectivity of this centroid – once the same value may represent activity in the middle strings or in the outer strings – we add a second value, which calculates the contribution of strings 3 and 4 (in %) in the selected segment.

3.2.2 Fret Range – Open Strings (%) – Slur (%)

Given a pre-determined tuning for the guitar, it is possible to determine which frets were pressed by the left hand fingers during the performance. The fret range is calculated from the lower and upper limits of the frets explored in the segment under analysis. This value may be related to the longitudinal displacement of the left hand along the fretboard. Besides the range, the descriptor includes also the border values. The percentage of used open strings – in comparison to the total number of onsets – is a quite straightforward descriptor, which nevertheless may express an important feature of the musical texture

under analysis. The slur descriptor represents the percentage of onsets that were detected without a sharp attack, or, in other words, the percentage of onsets derived from hammer-on, pull-off and glissando gestures.

3.2.3 Presence of Block Chords

It is not a trivial task to detect the presence of chords in a segment. Chords may be played in different ways: at once (block chords or *plaqué*), strummed, arpeggiated, or as *rasgueado*. For the definition of block chords, we opted for a somewhat arbitrary threshold of 30 ms: two onsets within this threshold are considered to belong to the same chord. This descriptor lists the chord sizes (in number of notes) found in the segment and their contribution (in % of "events" [2]) to the global texture.

3.2.4 String "Jump"

We call string "jump" the absolute value of the difference between the string indexes of two adjacent onsets. This jump can vary between the integers 0 and 5, and indicates the spatial proximity of two successive attacks. This descriptor is expressed by the mean value and the standard deviation found in each segment. Since the presence of block chords may somehow blur these results, it is also useful to calculate these values not taking the chords in account.

3.2.5 Periodicity of string indexes

The presence of periodicities in the sequence of string indexes may have a strong influence on the global gestalt of segments, and even on the gestalt of entire pieces. Before searching for periodicities, it is useful to replace the note indexes of a chord with a single index. In this way, 2-note chords indexes become the single index 20, 3-note chords indexes become 30, and so on. After this pre-processing stage, we calculate the average magnitude difference function (AMDF) of the list of indexes, using a window of 10 values (see equation 1). When the minima of this function are equally spaced, we may infer some sort of periodicity. If these minima are zero, we have an exact periodical exploration of string indexes. The descriptor's output consists of two lists, one with the minima values, the other with their spacing.

$$f(n) = \sum_{i=n+1}^{n+11} |x(i) - x(i-n)| \qquad (1)$$

3.3 Rhythm Descriptors

3.3.1 Superimposition Index

The simultaneous sounding of different guitar strings is a central characteristic of this instrument. This may happen even during the play of a single melody in more than one string. The superimposition index returns a single value for each segment, and is calculated as follows: the durations (time intervals between onsets and offsets) of all notes are added together, and the resulting sum is then divided by the global duration of the segment.

[2] Note that here a chord is equaled to one event.

3.3.2 Most Prominent IOIs

In the analysis of the set of IOIs (inter-onset intervals) extracted from each segment, there is no intent to search for beats, rhythmic patterns or meter, for we are interested in a more generic rhythmic activity. This is due not only to the significant variations found in performances of even very simple rhythmic relations [1], but also to the chosen approach, mostly influenced by the guitar affordance and by some specific technical features.

As already mentioned, IOIs below 30 ms are considered to belong to a single chord. This is a reasonable but not definitive choice, for we are aware that it is possible to have more than six successive onsets fulfilling this condition. Fast tremolos on different strings, or *rasgueados*, may cause situations like this. In these cases, the algorithm must also take in account the string indexes and amplitudes, before the chord segmentation. On the other side, IOIs from fast tremolos, fast strummed chords and similar gestures are expected to occur within a threshold of 80 ms. The lower limit of 80 ms was chosen by two main reasons. Firstly, it represents the occurrence of 12.5 equally-spaced onsets in one second, which is a fair threshold for the conscious control of rhythmic values on the guitar, valid also for trills. Secondly, as the system runs in real-time, it prevents that the margin error surpasses 10% of the calculated IOIs.

IOIs larger than 80 ms are divided in seven ranges, following a 3/2 proportion; these ranges may be further subdivided, depending on their variance. After that, the system analyzes the proximity of the mean values in each range and may calculate a more appropriate mean value. Finally, the values whose weight stands above 5% are selected and depicted as the output of this descriptor.

We are aware that the procedure just described is somewhat naïf, founded on our own experience and intuition. A comparison with other methods –such as cluster analysis– might validate the results obtained so far, suggest its refinement or even the replacement of the procedure.

3.3.3 Presence of Silences

If the lack of activity on all strings surpasses 300 ms, we define this situation as a silence in the segment. The number of occurrences and their duration (in ms) are the outputs of this descriptor.

3.4 Pitch Descriptors

3.4.1 Pitch range – (Most) Used Pitch Classes

For each segment, we calculate the pitch range in semitones, and a vector holding the number of occurrences of each pitch class. Observing this vector, it is easy to say which pitch classes - and how many - are in use. A list with the two – or three – most used pitch classes is also depicted by this descriptor.

3.4.2 3-Note Prime Form

If the previous list has three elements, this descriptor returns the prime form attributed to their set by Forte's theory [6]. The prime forms dedicated to 3-note sets are rela-

tively small in number – 12 – and constitute an effective guide for an improvisation process on the guitar, along with the nominal pitch classes represented in the list [3] .

3.4.3 Non-Pitched Events – Vibrato – Bend

These descriptors express the percentage of occurrence of each of the listed features, in comparison to the total set of events played in the segment. These features may give a very distinctive character to the segment.

3.5 Amplitude and Spectrum Descriptors

These remaining descriptors are quite straightforward, and are represented by a mean value and its standard deviation. Note that only the pitched events are considered in the calculation of the mean value of the spectral centroid.

All extracted data is stored for further use, even if they are not displayed in the analysis main window. They may be used in the future variations processes.

Some of these descriptors are related to more than one category; it is sufficient to mention that the use of open strings has also effects on the pitch content, and that the superimposition index and the presence of chords – listed here as guitar-specific descriptors – are also important rhythm descriptors.

4. ANALYSIS OF FIVE EXCERPTS

Five short musical excerpts were chosen to illustrate the use of the mid-level descriptors discussed above. We have opted to present only one rendition for each excerpt – played by undergraduated guitarists – since the paper is focused on the implementation of descriptors, not on the comparison of performances. The excerpts are the following: I) the initial phrase of Bach's *Bourrèe* [8], a simple two-voice excerpt consisting of bass and melody lines (see Figure 1); II) the beginning of Brouwer's *Études Simples II*, based on a choral texture, as depicted in Figure 3; III) the beginning of Villa-Lobos's *Étude no. 1*, dedicated to *arpeggios* (see Figure 4); IV) the first section of Brouwer's *Études Simples VII*, which explores fast notes with legatos (Figure 5); V) the beginning of Villa-Lobos's *Prélude no. 4*, which presents a tenor melody interrupted now and then by chords (Figure 6). We will first point out the most distinguishing mid-level descriptors values for each excerpt, before proposing a more global and comparative approach.

The mid-level descriptors values for excerpt I (Bach's *Bourrèe*) can be seen in Figure 2. Despite the fact that a significant part of the values are self-explanatory, we would like to comment some of these results. The calculated string "centroid" is 3.55, but the contribution of strings 3 and 4 is only 20%, indicating a more intense use of both low and high strings. The (no-chords) string jump points to the use of the same or of neighbor strings in

[3] In a different context, J. Bernard [7] justifies his analytical preference for trichords: "Groups of three pitches (trichords) are far more promising, for with trichords it is possible not only to define spatial configurations based on pairs of intervals but also to define relationships between trichords with a few simple operations. (...) However, even for tetrachords, the number of derivatives would increase to the point of being unwieldy."

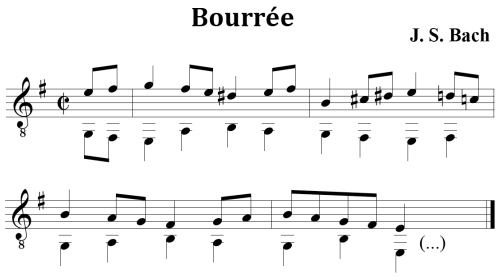


Figure 1. Beginning of J. S. Bach's *Bourrée*, from the Suite for Lute in E-minor.

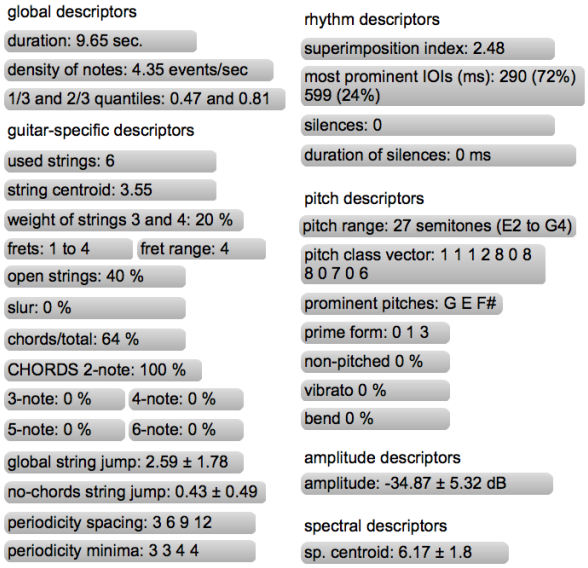


Figure 2. Mid-level descriptors for Bach's *Bourrèe* displayed in a Max window.

the disaccompagined notes. All two-note chords were detected, and also the two basic rhythmic values, which are in the proportion 1:2. The density of notes –4.35– is coherent with these values. The left hand remains in the first position, between frets 1 and 4. The most played pitch classes are G, E and F♯, in this order. The superimposition index is 2.48, pointing to a more sustained voicing, which is also helped by the significant amount of open strings. The rendition does not present significant variations in amplitude and spectrum.

In Brouwer's *Coral* (excerpt II), the 3-note chords represent 88% of the total events played. Only two most prominent pitch classes were indicated: G (with 11 occurrences) and D (with 8 occurrences). A tie between C, F and A occurred in the third position. The value 2.75 for the superimposition index may appear a bit surprising, since all chords have 3 notes. Despite that, we have observed in an earlier study [2] that block chords are hardly played without a considerable interruption between them, since we obtained a maximum value of 83% for the proportion between the sounding and the total (sounds + rests) parts. The two most prominent IOIs are 1626 and 771 ms, bearing a relation very close to 2:1.

The rendition of Villa-Lobos's study of *arpeggios* (ex-

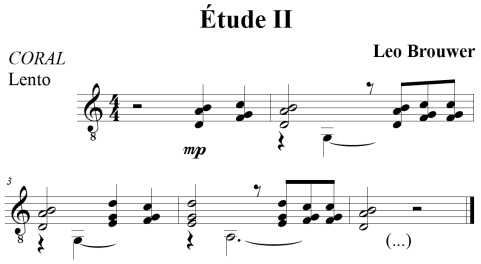


Figure 3. Beginning of Brouwer's *Étude II* [9].

.

cerpt III) was made in a moderate tempo, resulting in a value of 181 ms for the one and only calculated IOI. As the excerpt is played with a fixed position for the left hand in every measure, the superimposition index is quite high: 5.56. The regularity of the sequence of the right hand fingers causes a value of 1.49 for the string jump, with a standard deviation of 0.5. The most played pitches are the E-minor triad, corresponding to the prime set (1 3 7). The spectrum descriptor showed a high standard deviation value: 6.49. The periodicity between every 16 onsets was also detected.

The analysis of Brouwer's seventh study (excerpt IV) detected the two interruptions present in the score, one with 468 ms and the other with 336 ms. These values point to a common tendency to accelerate towards the third (and longer) segment of the excerpt. The slur articulation was responsible for 25% of the onsets, while 2% of non-pitched onsets were also detected. As expected, the string jump is very low – 0.55 –, with a standard deviation of 0.99. With the exclusion of the last chord, this descriptor is even lower: 0.4.

Excerpt V (also by Villa-Lobos) presents the most varied rhythmic activity in this study. Two segments of silence were detected, measuring 1489 and 744 ms respectively. They are located after the third *pianissimo* chord in measures 2 and 4. The difference between the two values is due not only to the use of vibrato, rubatos and fermatas in this piece, but also to the very soft and sometimes irregular playing of these chords. Most prominent IOIs are 1489, 315, 499 and 230 ms, which may be related to the proportions 3/2:1/3:1/2:1/4. Vibrato was detected in 17% of the notes. The fret range is 12, measured between frets 3 and 14. Amplitude mean value is very low (-46 dB), with a considerable standard deviation (10 dB). The comparison between the two values related to string jumps reveals the melodic character of the non-chordal passages.

So far, our efforts in comparing different excerpts have three complementary strategies. The first is the search for pregnant features, like an intense exploration of non-pitched onset, legatos, vibratos, bends, open strings, periodicities, etc. The setting of the thresholds for this characterization demands a more definite context or a larger database than the one we are currently using.

The second strategy is the search for features that may express, in a very loose way, the musician's activity on the instrument during that excerpt. The presence of chords, their proportion to the total events and their types are a good

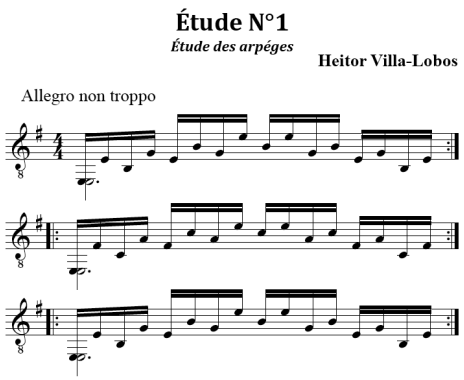


Figure 4. Beginning of Villa-Lobos's *Étude no. 1* [10]

.

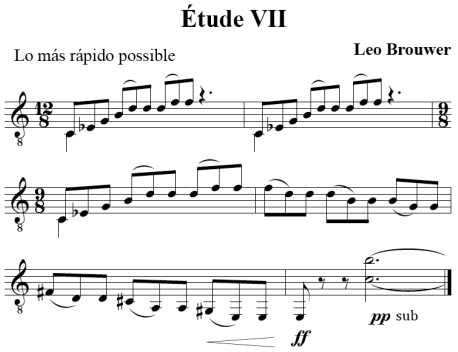


Figure 5. Beginning of Leo Brouwer's *Étude VII* [9]

.

starting point. The most prominent IOIs and their proportions are also very important in this task. The density of events, string centroid and the superimposition index descriptors add further details. More specifically related to the left hand activity, we have also the descriptors related to fret exploration and pitch content.

A third concern is about regularity or homogeneity. As mid-level descriptors are based on averages, they are prone to hide discontinuities, interruptions, contrasts. Despite the presence of descriptors that may somehow express changing features – rhythmic quantiles, presence of silences, the string jumps, periodicity of string indexes –, their contribution is clearly very modest. On the other side, our automatic segmentation procedure, to be used in improvisational contexts, follows roughly the limits given by G. Lewis for the change of phrase behaviors in his *Voyager* system [12]: there is an assumption of homogeneity within segments ranging from 3 to 7 seconds. The excerpts presented here are significantly longer, and their heterogeneity could be better analyzed with shorter segments or with sliding windows.

Table 1 and Table 2 illustrate the potential and challenges of this kind of analysis. It is not difficult to detect a homogeneous arpeggio texture, as in excerpt III, but it is much more difficult to describe some details of excerpt V based only on these values. A larger database and the introduction of variation processes in this project will certainly bring new ideas and tools regarding discontinuities.

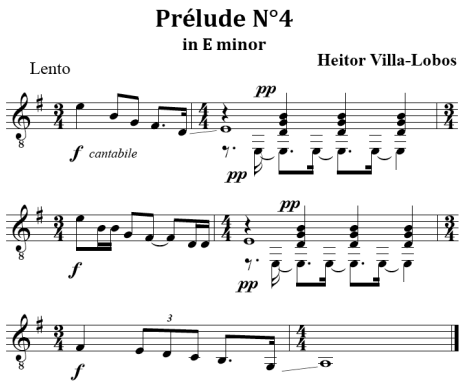


Figure 6. Beginning of Villa-Lobos's *Prélude no. 4* [11]

.

	chords types	IOIs (ms) (pauses)	density of notes (strings)	super. index (st. jump)
I	64% total 2-note(100%)	290 599	4.35 (6)	2.48 (2.59/0.5)
II	88% 3-note(100%)	1626 771	1.91 (5)	2.75 (1.48/1.0)
III	0%	181	5.48 (6)	5.56 (1.5/1.5)
IV	2% 2-note(100%)	159 197 (2 pauses)	4.87 (6)	1.33 (0.55/0.4)
V	18% 3-note(83%) 2-note(17%)	1489/315 489/230 (2 pauses)	1.35 (5)	1.48 (1.2/0.28)

Table 1. First set of mid-level descriptors values for the analyzed excerpts. Both string jump values –global and no-chords – are depicted.

5. FINAL REMARKS

The paper presented a series of mid-level descriptors related to the performance on an acoustic guitar, and their application in the analysis of a few selected excerpts. It is worth noting that many aspects not easily inferred from the excerpt's score are revealed through the analysis of its performance, although the small number of analyzed excerpts does not allow any generalizing attempt.

The preference for a non-stylistic approach does not exclude the incorporation of different algorithms already developed for tonality and metric detection, among others. For instance, beats could be acquired in real-time by means of beat tracking.

The development of a virtual player for interactive improvisation will also help refining the mid-level descriptors in use, and will certainly demand the enlargement of sound typologies and their corresponding low and mid-level descriptors.

Acknowledgments

We have received financial support for this research since 2010 from two Brazilian funding agencies, Fapemig and CNPq.

	frets	string centroid (open str.)	pitches prime form	1/3 and 2/3 quantiles
I	4 1–4	3.55 (40%)	G E F♯ (0 1 3)	0.47 0.81
II	3 1–3	3.18 (44%)	G D none	0.44 0.71
III	4 1–4	3.25 (48%)	E B G (0 3 7)	0.4 0.72
IV	7 1–7	3.08 (35%)	D B none	0.41 0.7
V	12 3–14	4.35 (52%)	E B none	0.34 0.66

Table 2. Second set of mid-level descriptors values for the analyzed excerpts.

6. REFERENCES

[1] S. Freire and L. Nézio, "Study of the *tremolo* technique on the acoustic guitar: experimental setup and preliminary results on regularity," in *Proc. Int. Conf. Sound and Music Computing*, Stockholm, 2013, pp. 329–334.

[2] S. Freire, L. Nézio, and A. Reis, "Analysis of the simultaneity, voice/layer balance, and rhythmic phrasing in works for guitar by Rodrigo, Brouwer, and Villa-Lobos," in *Proc. ICMC-SMC Joint Conf.*, Athens, 2014, pp. 1010–1015.

[3] R. Rowe, *Interactive Music Systems*. MIT Press, 1993.

[4] M. Puckette and T. Apel, "Real-time audio analysis tools for pd and msp," in *Proceedings of the ICMC 1998*, Ann Arbor, 1998, pp. 109–112.

[5] M. Rodrigues, *O vibrato no violão: aspectos qualitativos e quantitativos*. Master Dissertation, Federal University of Minas Gerais, 2014.

[6] A. Forte, *The Structure of Atonal Music*. Yale University Press, 1973.

[7] J. Bernard, *The Music of Edgard Varèse*. Yale University Press, 1987.

[8] J. S. Bach, *Suite in E Minor*. BWV 996, 1712-17?

[9] L. Brouwer, *Études Simples pour guitar (1ere Série)*. Max Eschig, 1973.

[10] H. Villa-Lobos, *12 Études pour guitare*. Max Eschig, 1973 (composed in 1928-29).

[11] ——, *Prèludes pour guitare*. Max Eschig, 1954 (composed in 1940).

[12] G. Lewis, "Too many notes: complexity and culture in voyager," in *Leonardo Music Journal*, 2000, pp. 33–39.

A Tambourine Support System to Improve the Atmosphere of Karaoke

Takuya Kurihara, Naohiro Kinoshita, Ryunosuke Yamaguchi, and Tetsuro Kitahara
Nihon University, Japan
{kurihara,kinoshita,yamaguchi,kitahara}@kthrlab.jp

ABSTRACT

Karaoke is a popular amusement, but people do not necessarily enjoy karaoke when they are not singing. It is better that non-singing people engage in karaoke to enliven it, but this is not always easy, especially if they do not know the song. Here, we focus on the tambourine, which is provided in most karaoke spaces in Japan but are rarely used. We propose a system that instructs how a non-singing person plays the tambourine. Once the singer choose a song, the tambourine part for this song is automatically generated based on the standard MIDI file. During the playback, the tambourine part is displayed in a common music game style with the usual karaoke-style lyrics. The correctness of the tambourine beat is fed to the display. The results showed that our system motivated non-singing people to play the tambourine with a game-like instruction even for songs that they did not know.

1. INTRODUCTION

Karaoke is an amusement quite familiar to most people. Many people enjoy singing karaoke. However, they do not necessarily enjoy karaoke when they are not singing. In fact, our survey of 30 students shows that 27 of the 30 students sometimes feel bored when they are not singing. This is mainly because they do not know what to do and is especially common when they do not know the song being sung. In most karaoke spaces in Japan, a percussive instrument, such as tambourine, is provided and customers can freely play it with the song. If someone can appropriately play such an instrument, it enhances the karaoke experience for everyone. In reality however, this instrument is rarely used because people do not know how to play or are too shy to play it. Here we hypothesize computing technologies can facilitate tambourine play by showing how to play it and by reducing shyness. This would be an effective too to enliven karaoke.

There have been many attempts to support and/or enhance karaoke. Cano et al. developed a system that modifies the user's singing voice by morphing it into a prerecorded voice of the same melody sung by another person [1]. Liu et al. developed a system that evaluates singing voice based on pitch and rhythm [2]. Daido et al. developed a system for evaluating singing enthusiasm using three acoustic features: A-weighted power, fall-down, and vibrato extent [3]. Tsai et al. proposed an automated singing evaluation method close to the human rating that exploits various acoustic features, including pitch, volume, and rhythm [4]. Thus, there are no attempts to enhance karaoke with a percussive instrument.

In this paper, we propose a tambourine support system, called Karatan to enhance karaoke. This system automatically generates a tambourine part (instructs how to play the tambourine) from a MIDI file and shows it with a usual karaoke-style lyrics display. A non-singing person can play tambourine according to the system's instruction without thinking of how to play it. Because he/she can enjoy playing the tambourine like common music games, the player will likely not feel self-conscious about it.

The remainder of the paper is organized as follows: In Section 2, we describe the concept and details of our system. In Section 3, we report experimental results conducted to confirm the effectiveness of our system. Finally, we conclude the paper in Section 4.

2. KARATAN: TAMBOURINE SUPPORT SYSTEM FOR KARAOKE

This system is designed to improve the engagement of non-singing people in karaoke. This happens because they do not know what to do when no singing. In general, non-singing people simply listen to the song being sung. However, just listening to the song does not necessarily enliven the experience. If they know the song being sung, they can sing it together. However, even songs known to all people are not always sung by all.

Here, we focus on the tambourine that is provided in most karaoke boxes in Japan, but rarely used. There are two possibilities why the tambourine is not used. The first is that they do not know how to play the tambourine effectively for unknown songs. The second possibility is that no one is motivated to play the tambourine. They think they might be embarrassed by inappropriately playing the tambourine rather than making karaoke more exciting by playing the tambourine.

Our system facilitates use of the tambourine through the following functions:

- Automatic tambourine part generation
  Once a song is chosen, the tambourine part is automatically generated from the standard MIDI file (SMF). We use SMFs because most karaoke machines are performed with music data equivalent to SMFs.
- Real-time tambourine performance feedback
  The tambourine part is displayed on the karaoke video screen and the performance of the player (whether he/she correctly played) is displayed in a common music game style. With this game-style display, people can easily try the tambourine performance.

Copyright: ©2015 Takuya Kurihara 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. Wii Tambourine

2.1 System Overview

This system is used with the Wii Tambourine, which we developed by building the Wii Remote into a tambourine (Figure 1). While the user plays the tambourine, the performance is analyzed by observing the accelerations of the Wii Remote. The procedure in this system consists of the following three steps:

1. Practice
   Once this system is launched, the system tries to establish a connection with the Wii Tambourine. After it succeeds, the practice mode starts. The user learns to play the tambourine in the parctice mode. This mode is needed not only for the user to learn this system's tambourine part display but also for the system to acquire the data for performance identification.
2. Tambourine part generation
   After the practice mode completes, the user chooses a song. Then, the system generates a tambourine part from the SMF of the chosen song. The tambourine part here includes all aspects of instructions for playing the tambourine (Figure 2). It includes intensities and body motions as well as the timing of the beat.
3. Real-time tambourine performance feedback
   Once the tambourine part is generated, the system starts to playback the SMF. During the play back, the tambourine part is displayed in a music game style along with the usual karaoke-style lyrics display (Figure 3). The user plays the tambourine while the singer sings along with this display. The tambourine performance is analyzed in real time and, if correctly played, is fed back to the display.

In the rest of this section, we describe these three steps in more details.

2.2 Practice mode

First, the user practices how to play the tambourine according to the instructions on the display (Figure 4). The contents of the practice are (1) beating freely, (2) beating strongly, (3) shaking, (4) beating at the upper position, and (5) beating at the lower position. In this order, the user freely practices the tambourine until he/she is satisfied. The system acquires the following data needed for performance identification:

- The mean value of the temporal differentials of accelerations in beating freely (α_1 in Equation (1)).
- The mean value of the number of direction changes within 20 frames of shaking (α_2 in Equation (3)).

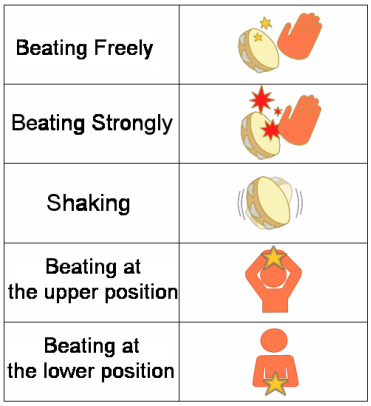


Figure 2. Instructions in tambourine performance



Figure 3. Karaoke screen

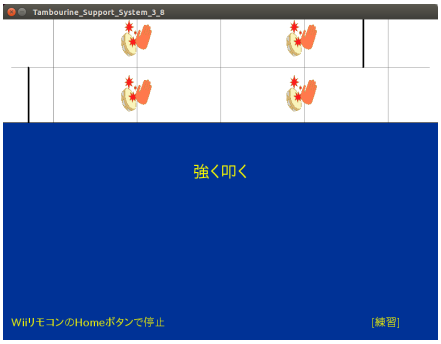


Figure 4. Practice mode screen

- The mean value of the temporal differentials of accelerations in strong beating (α_3 in Section 2.4.3).
- The mean value of temporal differentials and directions of rotation etc. used in body motion classification (Section 2.4.4).

2.3 Tambourine part generation

The difficulty in playing the tambourine in karaoke lies in a dilemma in that the tambourine should enhance the experience but not disturb the singing. In general, the singing voice in the chorus section (i.e., the high point of a song) in the verse-chorus form tends to be excited, so loud tambourine sound and a large body motion would not disturb the singing voice. Also in instrumental solo sections, the tambourine can be relatively freely played because there is no singing voice during these parts. We therefore adopt the following basic policies:

- The tambourine is basically played in the same way

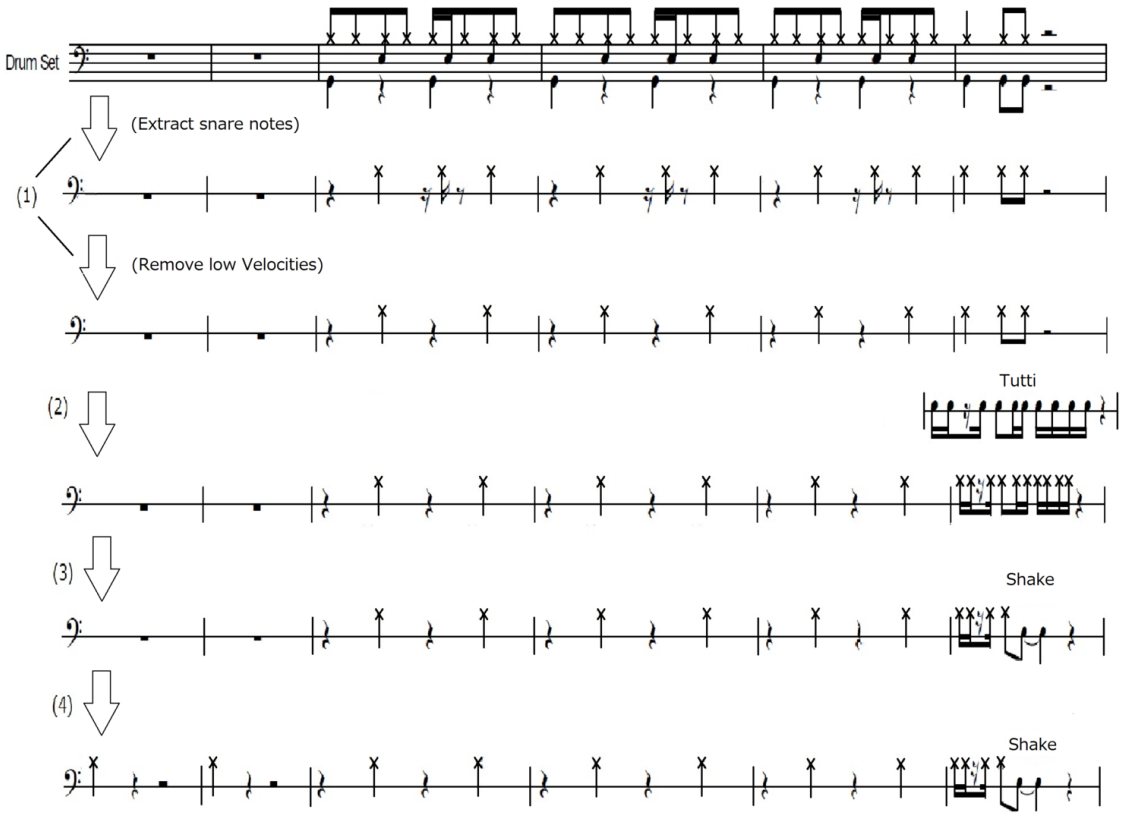


Figure 5. Example of tambourine part generation

as the snare drum.

- The tambourine is sometimes shaken instead of beaten.
- The tambourine is played quietly for quiet songs.
- Body motions are made for chorus and instrumental solo sections.

Based on these policies, our system generates the tambourine part according to the following algorithm:

2.3.1 Tambourine part generation

(1) First, a note sequence for the snare drum is extracted at the eighth-note level from the given SMF and regarded as tentative tambourine part. The snare drum part sometimes contains notes with low velocities that are not easy for non-musicians to play with the tambourine. The notes with velocities lower than a defined threshold are therefore removed. The threshold is determined as a velocity at which the histogram of the velocities of all notes in the given SMF's snare part has the lowest valley (Figure 5 (1)).

(2) Next, this tentative tambourine's part is corrected according to tuttis. When seven or more instruments play three or more notes in the completely same rhythm within one measure, these notes are considered a tutti. The tambourine is also played in the same rhythm whether the snare is played or not (Figure 5 (2)).

(3) Successive short notes could be difficult for non-musicians to play. When three or more eighth notes are played successively, the user is allowed to shake the tambourine instead (Figure 5 (3)).

(4) According to this algorithm, the tambourine is not played in the sections that have no snare notes. From the viewpoint of enlivening up karaoke, no tambourine sections should be avoided. For every measure without snare notes, we insert a tambourine note to the first beat (Figure 5 (4)).

2.3.2 Instruction of intensities

After the tambourine part is generated, the intensity is determined for each note of the tambourine part. Here, we adopt two types of instructions: "no particular instruction (i.e., with your favorite intensity)" and "beat strongly." We call these a normal note and a strong note, respectively. These instructions should be linked to the mood of the song such that the ratio of strong notes are controlled according to the overall dynamics of the song.

The overall dynamics of the song is defined by the mean value m of the velocities of all snare drum notes. If this value is higher, the ratio of strong notes should be higher. We therefore define the ratio of strong notes by

$$R = \begin{cases} \frac{1}{3}(1 + \frac{m-M}{127-M}) & (m > M) \\ \frac{m}{3M} & (otherwise) \end{cases}$$

where M is the mean value of m obtained from a variety of songs.

2.3.3 Instruction of body motion

Moving the body would be more effective than just playing the tambourine, but doing it from the beginning to the end would be burdensome. For the chorus and instrumental solo sections, the system gives instructions on body motions to the user. Specifically, the system gives an instruc-

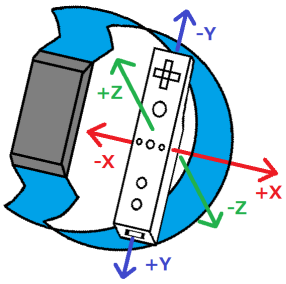


Figure 6. Wii Tambourine

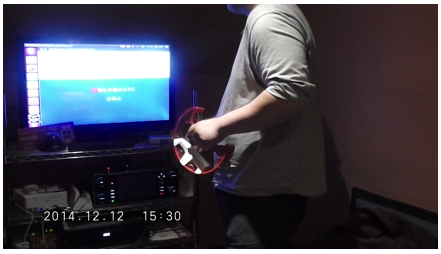


Figure 7. Photo of the experiment

tions on "at the upper position" or "at the lower position" alternately for every measure in the chorus and instrumental solo. The chorus sections are detected on the basis of Takada's method [6], which is based on Goto's idea [7] that the chorus section is included in the longest one of the repeated melodies, but is simplified because the target is a MIDI signal, not an audio signal. The instrumental solo sections can be easily found because the SMF we used includes an annotation of the instrumental solo sections as lyrics data.

2.4 Real-time tambourine performance feedback

While the song is being played, the generated tambourine part is displayed in a common music game style together with a usual karaoke-style lyrics (Figure 3). At the same time, the user's tambourine performance is analyzed through the acceleration data obtained from the Wii Remote. The accuracy of the performance is fed back through the display. Because the Wii Remote is built into the tambourine like Figure 6, it moves along the x-axis when it is beaten or shaken.

2.4.1 Beat detection

Every 16 ms (so the frame rate is 60 fps), the three-dimensional acceleration is observed. When the temporal differential of the x-axis acceleration is higher than a threshold, that is

$$|a_t - a_{t-1}| > 0.7\alpha_1, \tag{1}$$

where a_t is an acceleration at time t, the tambourine is considered to be beaten at time t. The threshold α_1 is the mean value of the temporal differentials of accelerations when tambourine is beaten in the practice mode.

2.4.2 Shake detection

Every 16 ms, the system counts how many times the direction of the tambourine's motion is changed for the last 20 frames. The change in the direction is detected from the zero-crossing of the acceleration. When

$$a_t a_{t-1} < 0, \tag{2}$$

the direction of the tambourine's motion is considered to be changed at time t. A shake is defined as the number c of direction changes is higher than a threshold, that is,

$$c \geq 0.7\alpha_2. \tag{3}$$

The threshold α_2 is the mean value of the numbers of direction changes within 20 frames in the practice mode.

2.4.3 Strong beat detection

Whether the tambourine is beated strongly is identified. This is performed in the same way as Section 2.4.1 except that a different threshold α_3 rather than as instead of α_1. The threshold α_3 is the mean value of the temporal differentials of accelerations when the tambourine is strongly beaten in the practice mode.

2.4.4 Body motion (hand's position) classification

The body motion in playing the tambourine ("at the upper" or "at the lower") is identified based on the method of Sawada et al [8]. This is performed by calculating the summation of the dissimilarities from the template, obtained in the practice mode, in the xy-, yz-, and zx-planes. Below, the definition of the dissimilarity in the xy-plane is described (the dissimilarities for the yz- and zx-planes are defined in the same way).

Every 16ms, the accelerations in the x- and y-axes are observed. Then, some features such as the mean values of temporal differential and direction of rotation for the last S frames are extracted ($S = 20$ in the current implementation). The dissimilarity in the xy-plane is defined as the mean square deviation of these features from those obtained in the practice mode. The dissimilarities in the yz- and zx-planes are defined in the same ways. We can then solve for the motion ("upper" or "lower") that minimizes the summation of the dissimilarities in the xy-, yz- and zx-planes.

3. EXPERIMENTS

We conducted experiments to assess the effects of our system. The participants were divided into groups, each of which consisted of three persons that played the roles of a singer, a tambourine player, and a listener. The three participants in each group knew each other because people usually go to karaoke with friends or acquaintances. We conducted two experiments. In Experiment 1, we used real Japanese popular songs that were included in a karaoke ranking and were known to the participants. In Experiment 2, we used songs unknown to the tambourine players to compare the effects of our system for tambourine players on known songs and unknown songs.

3.1 Experiment 1

3.1.1 Experimental conditions

We asked the participants to play karaoke with our system and the baseline system. The baseline system was implemented by removing the tambourine part generation and

display as well as the performance feedback function from our system. This was equivalent to standard karaoke machines. The participants were 12 university students and were divided into four groups. The role (a singer, a tambourine player, or a listener) of each participant was determined considering the familiarity to the target songs described below. The most familiar person sings, and the least familiar person listens. The procedure is as follows:

For Groups 1 and 2:

1. Practice the Wii Tambourine
2. Sing the singer's favorite song with a real karaoke machine
3. Play karaoke with our system (an easy song)
4. Play karaoke with our system (a hard song)
5. Play karaoke with the baseline system (an easy song)
6. Play karaoke with the baseline system (a hard song)

For Groups 3 and 4:

1. Practice the Wii Tambourine
2. Sing the singer's favorite song with a real karaoke machine
3. Play karaoke with the baseline system (an easy song)
4. Play karaoke with the baseline system (a hard song)
5. Play karaoke with our system (an easy song)
6. Play karaoke with our system (a hard song)

We adopted such a crossover style in order to avoid the order effect. As easy songs, we used *Zankoku Na Tenshi No These* (Yoko Takahashi) for our system and *Memeshikute* (Golden Bomber) for the baseline system. As hard songs, we used *Senbonzakura* (WhiteFlame feat. Miku Hatsune) for our system and *Kimi No Shiranai Monogatari* (Supercell) for the baseline system. These songs were taken from the karaoke ranking of JOYSOUND 2013 [9]. We used different songs for our system and the baseline system because we aimed to avoid the effect of experience—if a participant uses the baseline system for a song after using our system for the same song, he/she may play the tambourine better with the baseline system because he/she has the completely same experience. We therefore carefully chose different songs so that their tambourine performance difficulties (e.g., the numbers of beatings, rhythmic complexities, etc.) are similar. The goal of Step 2 (sing the singer's favorite song) is to let the participants relax.

After using our system or the baseline system, we asked the participants the following questions:

For singers:

Q-S1 Do you think the listener enjoyed your song?
Q-S2 Could you sing comfortably?
Q-S3 Did the tambourine performance disturb your song?
Q-S4 Were tambourine's timings appropriate?
Q-S5 Do you want to try the tambourine performance?

For tambourine players:

Q-T1 Was it easy to play the tambourine?
Q-T2 Could you play the tambourine along to the music?
Q-T3 Was your performance monotonous?
Q-T4 Could you play the tambourine to the rhythm?
Q-T5 Could you enliven karaoke through the tambourine?

For listeners:

Table 1. Results of Experiment 1

(a) Singers

	Baseline system		Our system	
	Easy song	Hard song	Easy song	Hard song
Q-S1	5.25	4.75	4.25	4.75
Q-S2	5.50	5.25	4.75	5.25
Q-S3	6.00	5.25	4.75	5.00
Q-S4	5.75	3.75	4.75	2.75
Q-S5	5.00	4.00	4.00	2.75

(b) Tambourine players

	Baseline system		Our system	
	Easy song	Hard song	Easy song	Hard song
Q-T1	4.75	3.50	4.50	4.75
Q-T2	5.00	4.00	5.00	3.50
Q-T3	3.25	1.00	6.00	4.75
Q-T4	4.25	2.00	5.00	4.75
Q-T5	3.75	3.00	3.75	4.50

(c) Listeners

	Baseline system		Our system	
	Easy song	Hard song	Easy song	Hard song
Q-L1	4.50	5.50	3.75	4.75
Q-L2	4.25	4.75	3.50	4.50
Q-L3	5.25	5.25	3.25	4.00
Q-L4	4.75	5.00	3.50	4.75
Q-L5	3.50	3.50	4.50	4.25
Q-L6	3.25	2.25	4.50	4.25

Q-L1 Did you feel bored?
Q-L2 Did you have an interest in the song?
Q-L3 Was the tambourine noisy?
Q-L4 Did the tambourine correspond to the music?
Q-L5 Do you want to sing along with the tambourine?
Q-L6 Do you want to try playing the tambourine?

The answers to these questionnaires were on a scale from one to six. A positive answer has a high value. For example, "6" for Q-L1 means "definitely no" while "6" for Q-L2 means "definitely yes".

3.1.2 Experimental results

The results are listed in Table 1.

Tambourine players rated our system highly. In particular, it was good that our system did not make the performance monotonous. However, a participant answered that the generated tambourine score is complicated in part. In particular, it was difficult to play the tambourine while moving the body. It would be better to introduce a mechanism for controlling the difficulty according to the player's musical skill.

For listeners, the answers to Q-L5 and Q-L6 were particularly high. This means that our system successfully gave listeners high motivations to play the tambourine. Participants gave us an opinion that this system increases the enjoyment as a music game as well as a tambourine support system.

Singers did not evaluate our system highly. This is probably because the tambourine player's mistakes were noticeable because the system sometimes required complex performances over the player's skill. This also suggests that difficulty control should be introduced.

Table 2. Results of Experiment 2

(a) Tambourine players

	Baseline system		Our system	
	Known song	Unknown song	Known song	Unknown song
Q-T1	4.50	3.25	5.00	5.00
Q-T2	5.50	3.00	4.75	5.50
Q-T3	1.75	1.25	4.25	1.75
Q-T4	5.00	2.50	4.75	5.50
Q-T5	2.75	2.00	4.50	4.00

(b) Listeners

	Baseline system		Our system	
	Known song	Unknown song	Known song	Unknown song
Q-L1	5.50	3.25	4.75	4.25
Q-L2	5.50	3.75	4.75	3.75
Q-L3	4.50	3.75	4.00	5.00
Q-L4	4.00	2.50	4.00	3.75
Q-L5	5.00	1.75	4.25	5.00

3.2 Experiment 2

3.2.1 Experimental conditions

When the tambourine player did not know the song being sung, our system was expected to support the tambourine player. To confirm this, we conducted an experiment with a song unknown to the tambourine player (we call it an unknown song). We took an unknown song from RWC Music Database (Popular Music) because this database consists of J-pop-style songs originally composed for research purposes, which are certainly unknown to the participants. We used *REAL Na Gofun* (RWC Music Database) as an unknown song and *Memeshikute* (Goldenbomber) as a known song for our system, and *Replica* (RWC Music Database) as an unknown song and *Zankoku Na Tenshi No These* (Yoko Takahashi) as a known song for the baseline system.

The procedure of this experiment and the questionnaires were the same as Experiment 1 except that the unknown songs were used in Steps 4 and 6 instead of the difficult songs. Because this experiment focused on the tambourine player's behavior, we played the role of singer. Hence, there are no questions for the singer. The participants were the same as those in Experiment 1 except for the singers.

3.2.2 Experimental result

The questionnaire results are listed in Table 2. Our system was highly evaluated in almost all conditions for unknown songs. We found that without our system, it is difficult to enhance karaoke with tambourine play because such play is very conservative. With our system, the tambourine players successfully played the tambourine for unknown songs. In turn, listeners also enjoyed the songs.

4. CONCLUSION

Tambourines are provided in most karaoke boxes in Japan and are expected to play a significant role in enhancing karaoke, but they are rarely used in practice. To give an opportunity to play the tambourine, we developed a tambourine support system that generates a tambourine part. Through this system, the user can play the tambourine like a common music game even if he/she does not know the song being played. The results showed that our system motivated tambourine players to some extent but singers were confused when the tambourine player made a mistake due to complex tambourine instructions over the player's skill. Future issues will include difficulty adjustment of the tambourine part, support of more various body motions, and improvement of tambourine performance identification.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 26240025.

5. REFERENCES

[1] Pedro Cano, Alex Loscos, Jordi Bonada, Maarten de Boer and Xavier Serra, "Voice Morphing System for Impersonating in Karaoke Applications" in *ICMC*, Germany, 2000.

[2] Liu Yuxiang, Jin Zeyu, Jia Jia and Cai Lianhong, "An Automatic Singing Evaluation System" in *Applied Mechanics and Materials*, Switzerland, 2012, Vol 128-129, pp. 504–509.

[3] Ryunosuke Daido, Seong-Jun Hahm, Masashi Ito, Shozo Makino and Akinori Ito, "A System for Evaluating Singing Enthusiasm for Karaoke" in *ISMIR*, Miami, 2011.

[4] Wei-Ho Tsai and Hsin-Chieh Lee, "An Automated Singing Evaluation Method for Karaoke System" in *IEEE Transactions on Audio, Speech and Language Processing*, 2011, pp. 2428-2431.

[5] http://www.nintendo.co.jp/wii/controllers/index.html

[6] Tomonori Takada and Hiroki Hashiguchi, "Detection of refrain from melody track in MIDI" in *Saitama University Bulletin*, Japan, 2006-7, pp. 107-109.

[7] Masataka Goto, "A Chorus Section Detection Method for Musical Audi Signals and Its Application to a Music Listening Station" in *IEEE Transactions on Audio, Speech and Language Processing*, 2006-9, Vol 14, No.5, pp.1783–1794.

[8] Hideyuki Sawada and Shuji Hashimoto, "Gesture Recognition Using Acceleration Sensor and Its Application to Musical Performance Control" in *Electronics and Communication*, Japan, 1997, Vol 80, No 5, pp. 9–17.

[9] http://www.joysound.com/st/2013yearranking

[10] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura and Ryuichi Oka, "RWC Music Database:Music Genre Database and Musical Instrument Sound Database" in *ISMIR*, 2003-10, pp. 229–230.

Follow the Tactile Metronome: Vibrotactile Stimulation for Tempo Synchronization in Music Performance

Marcello Giordano **Marcelo M. Wanderley**
Input Devices and Music Interaction Laboratory (IDMIL)
Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT)
McGill University
Montréal, Québec, Canada
marcello.giordano@mail.mcgill.ca

ABSTRACT

In this paper we present a pilot study evaluating the effectiveness of a tactile metronome for music performance and training. Four guitar players were asked to synchronize to a metronome click-track delivered either aurally or via a vibrotactile stimulus. We recorded their performance at different tempi (60 and 120 BPM) and compared the results across modalities. Our results indicate that a tactile metronome can reliably cue participants to follow the target tempo. Such a device could hence be used in musical practice and performances as a reliable alternative to traditional auditory click-tracks, generally considered annoying and distracting by performers.

1. INTRODUCTION

Much research has so far been devoted to the study of the psycho-physical properties of the sense of touch [1]–[4] and to the development of new, tactile-enabled interfaces capable of addressing this rich sensory channel. In recent years, research in this field has been fostered by the ongoing "mobile revolution" and the widespread availability of actuators for mobile and wearable devices [5].

Increasing interest has also been dedicated to the role of haptic feedback and stimulation in the context of musical interaction [6]–[9]. Haptic perception plays an important part in the process of embodiment of a musical instrument, in shaping the perceived qualities of an acoustic musical instrument [10] and, in expert performance, for tasks such as articulation and timing [11]. Guitarists, for instance, have stated to rely on tactile cues for timing tasks [1].

At the same time, several patents for tactile metronomes [12], [13] have been filed in the last decade, and commercial tactile-augmented devices have started to appear on the market [14], [15], claiming to be able to provide musicians with reliable tempo cues. Surprisingly though, no quantitative evaluation of the capability of the sense of touch to process such information, in the context of music performance, has been conducted so far [16]. Most of the literature in the field of synchronization studies to a metronome signal is mostly limited to tapping experiments [17], in which participants are not actively engaged in any activity. The results from these experiments, while still extremely valuable to evaluate the performance of tactile perception in synchronization tasks, cannot be directly applied to the context of music performance.

The development of the aforementioned commercial products, although not supported by perceptual evidence, testifies nonetheless the interest of researchers, industry and general public for haptic-enabled interfaces targeted to musicians.

With this study, we aimed at performing a pilot, quantitative evaluation, by providing evidence that vibrotactile stimulation can proficiently be used to convey tempo information in music performance. We designed a synchronization tasks in which participants are actively engaged in musical task, such as performing an ascending and descending scale on a classical guitar.

Ultimately, our goal is to provide evidence that haptic technology can be extremely beneficial for musical training and rehearsal: by delivering musical information through haptic cues, auditory cognitive load can be reduced, allowing musicians to redirect their auditory attention to other tasks.

2. PREVIOUS WORK

Few researchers have investigated the use of tactile stimulation in the context of musical interaction. Several works, for instance, have addressed the evaluation of rhythm perception through the sense of touch [18], [19], by showing that tactile rhythm discrimination performs comparatively well compared to audition.

Michailidis and Berweck [20] developed a *tactile feedback tool* to inform muscians about the successful activation of effects using a foot pedal during live performance.

Schumacher et al. [21] investigated the effectiveness of using haptic cues in the context of interaction with a live-electronics environment for composition and performance. By means of two vibrating actuators attached to the back of a performer, the authors conveyed information about the state of the live-electronics system as well as tempo and articulation cues. These cues are traditionally delivered via auditory or visual displays [22], but this practice is often

Copyright: ©2015 Marcello Giordano 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.

judged as obtrusive and distracting by performers. Participants in the pilot study expressed positive feedback about the tactile display, especially for what concerns the tempo information, but no evaluation was carried on about the effectiveness of the haptic cues.

3. EVALUATION OF A TACTILE METRONOME

A prototype of a tactile metronome was designed and built using off-the-shelf hardware components. This device was used for conducting a pilot test aimed at characterizing the effectiveness of such a device in music performance.

3.1 System Design

A VPM2 actuator driven by an Arduino Mini-Pro and a ULN2803A [1] motor driver was used to deliver the metronome signal. This inexpensive motor can be driven using a PWM signal which allows only one control parameter: the duty cycle of the PWM wave, ranging from 0 to 1. This can be effectively considered as a control over the intensity of vibration.

The intensity was set to a 0.8 value for the duty-cycle of the PWM wave; this value is well above the perceptual threshold we identified in a previous study [23], which is set at 0.2. The intensity value remained constant throughout the tests and across participants.

The actuator was fixed to the left upper arm of the performer (see Fig. 1). This location was chosen mainly because it did not hinder participants movements, and because sensitivity in this area is reported to be high enough [24] to allow reliable perception of the tactile metronome signal. Other locations were tested: the left wrist was judged by the performers as obtrusive and disturbing; the left ankle caused some of the stimuli to go undetected since some of the non classically trained participants tended to tap along the tempo with their left foot.

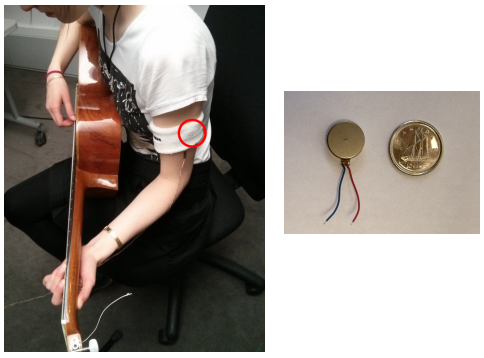


Figure 1: On the left, one participant wearing the tactile display on her left arm (circled in red). On the right, one VPM2 actuator.

A software control environment written in Max [2] was designed to generate synchronized tactile and auditory metronome tracks, and to record musicians' performance. The auditory metronome was delivered through headphones connected to an external RME Fireface 400 [3] audio interface. Only the left channel was used to match the lateralization of the tactile stimulus. Headphones were also used to deliver quiet white noise to mask actuator noise during the tactile metronome trials.

The overall latency of the system was also evaluated: a PCB Piezotronics accelerometer [4] was attached to the actuator. The output from the sensor was connected to one of the external audio interface input channels. The delay of the interface had previously been estimated to be in the order of 2.5 ms looping the output back into the input, and using a Max input-output buffer. Using this setup, the delay between the serial commands sent to the Arduino board and the activation of the motor was evaluated to be in the order of 3 ms. In a previous study [23], we evaluated to 15 ms the time needed for the actuator to reach its supra-threshold amplitude vibration from a steady state. Hence, the system combined (i.e. mechanical and perceptual) total delay was estimated to be around 18 ms. This delay has been taken into account in the analysis of the preliminary data. For the auditory metronome and recording system, the aforementioned audio interface was used. We assumed the delay added by transmission through the headphones and microphone wires to be negligible, thus obtaining an overall delay for the recording apparatus and auditory metronome to be in the order of 2.5ms. These delays have been taken into account in the analysis of the preliminary data.

3.2 Methodology

Four guitar players [5] were asked to play the first seven notes (ascending and descending) of a G major scale while synchronizing to either a tactile or an auditory metronome. The G major scale is generally recognized as an easy exercise for experienced players and was hence chosen so not to present participants with a too demanding task [25], while still engaging them in an ecologically valid musical task. The tactile stimuli were delivered through our display, while the auditory stimuli were delivered via headphones.

To ensure that neither the auditory nor the tactile metronome stimuli would be perceived as *stronger*, an equalization phase was performed for each participant prior to the beginning of the experiment. Each was presented with both the tactile and the auditory click-tracks and asked to evaluate the *perceived intensity* of the auditory track relatively to the tactile one, which was fixed as reference. Participants instructed the experimenter to increase or decrease the volume of the auditory click-track in order to match the perceived intensity of the two stimuli. Subsequently participants were exposed to four metronome conditions, varying sensory modality and metronome speed (expressed in Beats Per Minute, or BPM) in random order:

- Tactile Metronome at 120 BPM,

[1] `http://pdf.datasheetcatalog.com/datasheet/SGSThomsonMicroelectronics/mXssxrt.pdf`

[2] `http://www.cycling74.com`

[3] `https://booking.cirmmt.org/media/model/71/fface400_e.pdf`

[4] `https://booking.cirmmt.org/media/model/420/352C23.pdf`

[5] Participants reported having at least 3 years of formal or informal practice on the instrument

- Tactile Metronome at 60 BPM,
- Auditory Metronome at 120 BPM,
- Auditory Metronome at 60 BPM

Participants were asked to play synchronously with the metronome track while being recorded through a microphone. They could start their performance at any time after the metronome signal had started, and each recording session lasted 60 seconds (i.e. after they played 120 notes at 120 BPM or 60 notes at 60 BPM). Participants performed each condition only once.

At this preliminary stage we decided to test only two tempi. The 60 and 120 BPM values were chosen since they are generally associated by musicians to, respectively, a slow and a fast tempo [26].

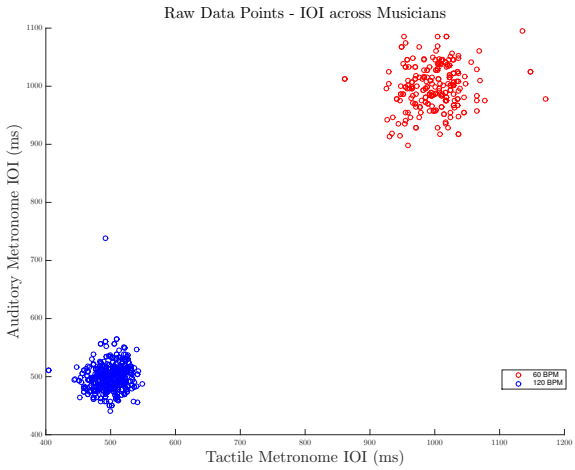


Figure 2: Raw data points showing time interval between each note played by musicians, at both tempi and metronome modalities. The target tempo corresponds to 1000 ms for 60 BPM (in red) and 500 ms for 120 BPM (in blue).

3.3 Data Analysis

The audio files were analyzed to extract onset information from participants' performances. We tested several onset detection algorithms but none succeeded in accurately analyzing the recorded data. Therefore we manually annotated the onset on the audio files to match the attack of each guitar pluck (see Fig. 3). Interpolation was used to compensate for missing plucks.

These onsets were compared to the metronome signal in order to evaluate participants' asynchrony in each modality, moreover participants' deviation from the target tempo was also evaluated.

3.4 Results

An asynchrony vector (representing participants' *lag* or *delay* with respect to the metronome signal) was computed for each participant by subtracting plucking-time vectors

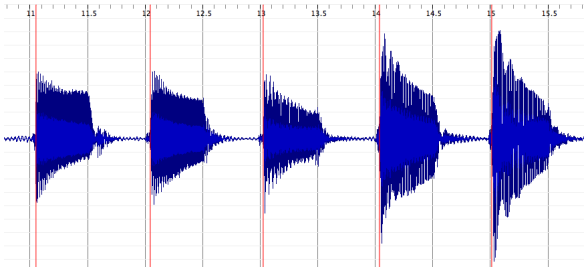


Figure 3: An excerpt of one performance (60 BPM Tactile). Recorded guitar plucks are shown in blue. The red lines indicate the manually annotated onsets. Time scale is visible on the top (time in seconds).

to metronome-time vectors. The average results across participants are shown in Tab. 1: participants plucks happen after the corresponding metronome signal and this delay is much more pronounced for the tactile modality. This suggests, in accordance with findings reported in the psychophysical literature [27], that the processing time for the tactile metronome signal is higher than for the auditory one. Fig. 4 illustrates the distribution of the asynchrony data about the median value.

Modality	60 A	60 T	120 A	120 T
Asynchrony (ms)	41.77	77.63	31.73	111.85

Table 1: Average asynchrony times across participants for each modality (A for auditory metronome, T for tactile metronome). Positive values indicate that, on average, the plucking happened **after** the metronome tick.

Fig. 2 illustrates the distribution of the raw data points for each note Inter Onset Interval (IOI) across participants, tempi and sensory modalities. The plot shows that, for both the tactile and auditory conditions, participants play each note with an interval fluctuating around 500 ms or 1000 ms for 120 BPM and 60 BPM respectively. The fluctuations are more pronounced at 60 BPM, consistently for both modalities.

Fig. 5 shows the distribution of the average deviation time from the target tempo participants were supposed to follow. This is expressed as a deviation from IOI. Again, as remarked in fig. 2, participants' timing is more accurate for the 120 BPM tempo compared to the 60BPM, and their performance is comparable across modalities in this case. This suggests that the tactile metronome can still be used to cue participants to follow the right tempo.

4. DISCUSSION

Results from the asynchrony analysis (Tab. 1 and Fig. 4) give multiple indications:

Asynchrony is generally more pronounced at 60 BPM in the auditory modality, while the mean value indicates an increased value for the 120 BPM fot the tactile modality. Participants' reaction time to the metronome ticks delivered via the tactile display is substantially slower than for

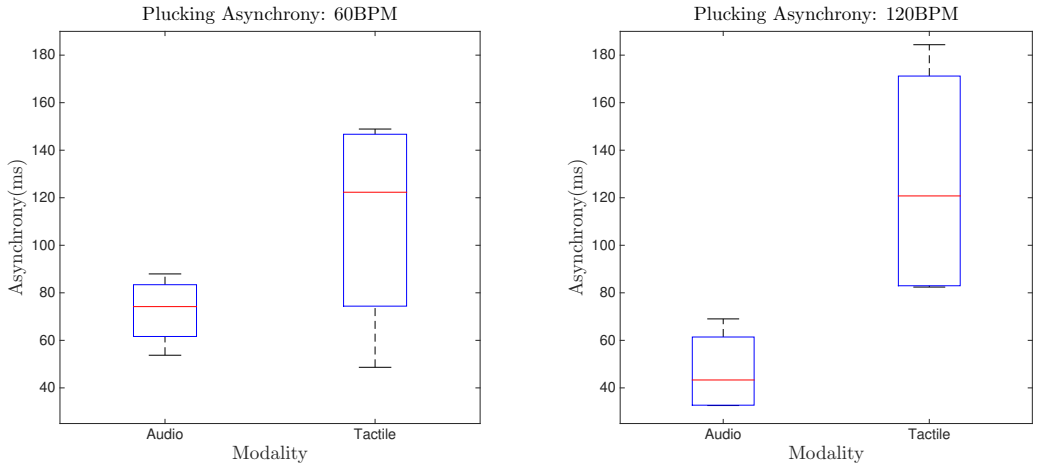


Figure 4: Distribution of the RMS median values illustrating asynchrony with respect to the metronome signal for each modality.

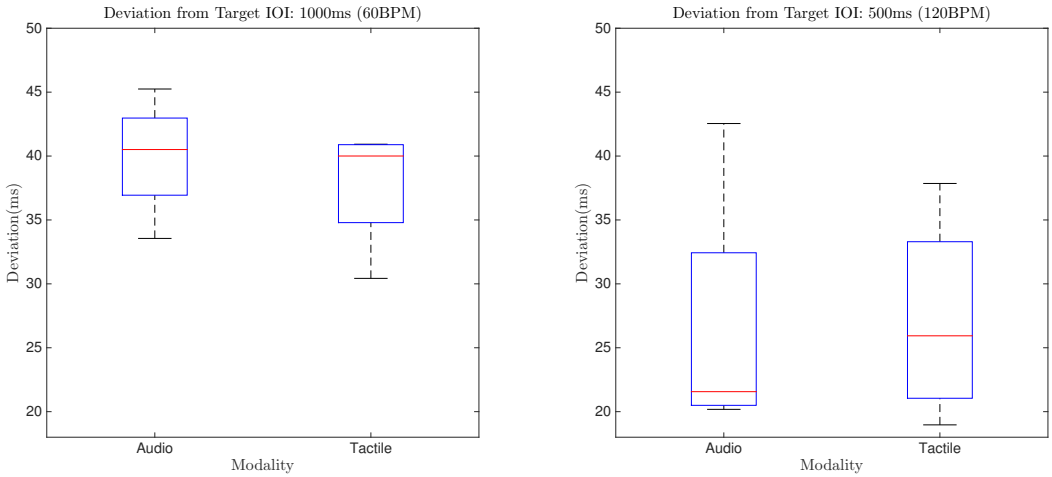


Figure 5: Distribution of the RMS median values for the deviation from the target tempo expressed as an Inter Onset Interval (IOI).

the auditory click-track. Also the response time fluctuates considerably around the median value, suggesting that adaptation and masking effects might influence the perception of the tactile stimuli during each take. This increased delay in reaction time could be due to multiple factors: low intensity of the tactile stimuli; transmission speed through the tactile sensory system. The slower response could also be due to the high motor complexity of our task (playing guitar) which could. Given the small sample size we could not identify a trend in this fluctuations.

The deviation-from-target-IOI analysis (Fig. 2 and 5) indicates that using the tactile click-track, participants are capable of following the target tempo as accurately as with the auditory metronome. For the 1000 ms target (60 BPM), the median deviation value is around 40 ms for both modalities, and the magnitude of fluctuation is comparable. For the 500 ms target (120 BPM), the media is at 22 ms for the auditory modality and 27 ms for the tactile. In both cases the data fluctuates more around the median for this target IOI than the previous one.

Overall these preliminary, descriptive analyses indicate that a tactile metronome can cue performers to follow a given target tempo with an accuracy comparable to that of an auditory click-track. The absolute delay is increased up to an average of 111.85ms for the 120 BPM tactile click-track. This increased reaction time could be compensated for by anticipating the tactile-click track by the necessary amount of time necessary for perception and processing of the tactile stimuli.

4.1 Future Work

Further investigation is needed to more precisely characterize the asynchrony due to increased processing time of the tactile stimuli. Moreover, different body locations should be tested to assess the variation of delay in reaction time in relation to the stimulated body part.

The small sample size allowed us only to perform a descriptive analysis of the data; a larger sample size will be needed for further statistical analysis. These questions will be object of the continuation of this study which will be

performed on 12 expert guitar players.

5. CONCLUSIONS

We presented a pilot experiment conducted on guitar players to assess the effectiveness of a tacile metronome for a music performance and practice. Participants were asked to play synchronously with a click-track, which was delivered either aurally or via a vibrotactile display.

The presented results, while preliminary, show that vibrotactile stimulation can effectively provide musicians with the necessary cues to play at the given tempo. In particular, deviation from target IOI was shown to be comparable between the auditory and the tactile modality. Absolute timing with respect to the metronome track showed higher asynchrony for the tactile click-track; this is most likely due to the increased processing time of a tactile stimuli. This asynchrony, if well characterized in advance, can be accounted for in the design of future tactile metronome interfaces.

Ultimately, we have showed that a tactile metronome can be successfully used to convey tempo information to musicians engaded in a demanding task. Such devices, if routinely integrated in musical practice, could expand performance and rehearsal possibilities.

Acknowledgment

This research was funded by a Natural Science and Engineering Research Council of Canada (NSERC) Discovery grant.

References

[1] R. T. Verrillo, "Vibration sensation in humans", *Music Perception*, vol. 9, no. 3, pp. 281–302, 1992.

[2] O. Franzén and J. Nordmark, "Vibrotactile frequency discrimination", *Perception & Psychophysics*, vol. 17, no. 5, pp. 480–484, Sep. 1975.

[3] S. J. Bolanowski, G. A. Gescheider, R. T. Verrillo, and C. M. Checkosky, "Four channels mediate the mechanical aspects of touch.", *The Journal of the Acoustical Society of America*, vol. 84, no. 5, pp. 1680–1694, Nov. 1988.

[4] M. Grünwald, *Human haptic perception: basics and applications*. Birkhäuser, 2008.

[5] S. Choi and K. J. Kuchenbecker, "Vibrotactile display: perception, technology, and applications", *Proceedings of the IEEE*, pp. 1–12, 2012.

[6] C. Chafe, "Tactile audio feedback", in *Proceedings of ICMC*, 1993, pp. 76–79.

[7] J. Rovan 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] D. M. Birnbaum and M. M. Wanderley, "A systematic approach to musical vibrotactile feedback", in *Proceedings of ICMC*, Citeseer, 2007.

[9] F. Fontana, F. Avanzini, H. Järveläinen, S. Papetti, F. Zanini, and V. Zanini, "Perception of interactive vibrotactile cues on the acoustic grand and upright piano", in *Proceedings of the joint International Computer Music Conference (ICMC) and Sound and Music Computing Conference (SMC)*, 2014.

[10] I. Wollman, C. Fritz, and J. Poitevineau, "Influence of vibrotactile feedback on some perceptual features of violins", *The Journal of the Acoustical Society of America*, vol. 136, no. 2, pp. 910–921, 2014.

[11] M. Puckette and Z. Settel, "Nonobvious roles for electronics in performance enhancement", in *Proceedings of the International Computer Music Conference*, 1993, pp. 134–134.

[12] C. V. Parsons, *Tactile metronome, united states patent 20040099132*, 2004.

[13] S. L. Fulford, *Tactile tempo indicating device, united states patent 5959230*, 1999.

[14] Peterson Tuners. Peterson bodybeat sync, [Online]. Available: `http://www.petersontuners.com/index.cfm?category=163` (visited on 04/15/2015).

[15] Soundbrenner. Soundbrenner pulse, [Online]. Available: `http://www.soundbrenner.com/` (visited on 04/15/2015).

[16] B. H. Repp, "Sensorimotor synchronization: a review of the tapping literature.", *Psychonomic bulletin & review*, vol. 12, no. 6, pp. 969–992, 2005.

[17] R. Brochard, P. Touzalin, O. Després, and A. Dufour, "Evidence of beat perception via purely tactile stimulation.", *Brain research*, vol. 1223, pp. 59–64, Aug. 2008.

[18] K. Kosonen and R. Raisamo, "Rhythm perception through different modalities", in *Proceedings of Eurohaptics*, 2006, pp. 365–370.

[19] M. Jokiniemi, R. Raisamo, J. Lylykangas, and V. Surakka, "Crossmodal rhythm perception", *Haptic and Audio Interaction Design (HAID), Lecture Notes in Computer Science (LNCS)*, vol. 5270, pp. 111–119, 2008.

[20] T. Michailidis and S. Berweck, "Tactile feedback tool : approaching the foot pedal problem in live electronic music", in *Proceedings of ICMC*, 2011.

[21] M. Schumacher, M. Giordano, M. M. Wanderley, and S. Ferguson, "Vibrotactile notification for live electronics performance: a prototype system", in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2013.

[22] E. McNutt, "Performing electroacoustic music: a wider view of interactivity", *Organised Sound*, vol. 8, no. 03, pp. 297–304, Apr. 2004.

[23] E. Frid, M. Giordano, M. M. Schumacher, and M. M. Wanderley, "Physical and perceptual characterization of a tactile display for a live-electronics notification system", in *Proceedings of the joint International Computer Music Conference (ICMC) and Sound and Music Computing Conference (SMC)*, 2014.

[24] M. Bikah, M. S. Hallbeck, and J. H. Flowers, "Supracutaneous vibrotactile perception threshold at various non-glabrous body loci.", *Ergonomics*, vol. 51, no. January 2015, pp. 920–934, 2008.

[25] D. J. Wright, P. S. Holmes, F. Di Russo, M. Loporto, and D. Smith, "Differences in cortical activity related to motor planning between experienced guitarists and non-musicians during guitar playing", *Human Movement Science*, vol. 31, no. 3, pp. 567–577, 2012.

[26] R. a. Duke, "Musicians' perception of beat in monotonic stimuli", *Journal of Research in Music Education*, vol. 37, no. 1, p. 61, 1989.

[27] S. J. Lederman and R. L. Klatzky, "Haptic perception: a tutorial.", *Attention, perception & psychophysics*, vol. 71, no. 7, pp. 1439–1459, 2009.

LICHTGESTALT: INTERACTION WITH SOUND THROUGH SWARMS OF LIGHT RAYS

Jonas Fehr
Aalborg University Copenhagen
jfehr13@student.aau.dk

Cumhur Erkut
Aalborg University Copenhagen
cer@create.aau.dk

ABSTRACT

We present a new interactive sound installation to be explored by movement. Specifically, movement qualities extracted from the motion tracking data excite a dynamical system (a synthetic flock of agents), which responds to the movement qualities and indirectly controls the visual and sonic feedback of the interface. In other words, the relationship between gesture and sound are mediated by synthetic swarms of light rays. Sonic interaction design of the system uses density as a design dimension, and maps the swarm parameters to sound synthesis parameters. Three swarm behaviors and three sound models are implemented, and evaluation suggests that the general approach is promising and the system has potential to engage the user.

1. INTRODUCTION

This paper presents the design and development of LichtGestalt: an interactive sound installation that is explored by movement. The movement aspects of the installation has been reported in [1], here our focus is on the mapping between movement and sound, sound synthesis, and dynamic generation of control.

Previously, direct manipulation of three tangible interfaces equipped with sensors was controlling the sounds and colors of the installation [2]. With collaboration in mind, the installation was designed as three tangible interfaces mounted on "branches" coming from a central trunk (Figure 1).

Inspired by works like *Polymetros* [3], the focus was on a minimal musical expression. Each interface was equipped to track its position in a two-dimensional space and was mapped to an individual voice allowing manipulation of timbre and the tempo of a tremolo effect. When all three interfaces were active, the audience could play a drone, with polyrhythmic qualities changing according the tempo of the tremolo. An additional sound quality, a high pitch crackle, was controlled through parameters based on the sum of all three interfaces. To provide *transparency* between movement and sound, visual feedback based on multicolour LEDs was designed representing both sound qualities in combination with the tracked position.

Copyright: © 2015 First author et al. 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.

An evaluation of SonoFluo indicated issues on control. In most cases the participants did understand what was controlled and what originally was planed as musical collaboration by movement became a collaborative exploration of how the system could work. Franinovic and Salter attribute this to designers' "*... almost formulaic understanding of interaction as a series of input-output processes ... This assumes an already fixed set of relations among the user/interactor, the object/instrument/sound-making body, and the environment in which the interaction with sound takes place."* [4]. To vary these relations, in LichtGestalt we have implemented an indirection between motion and sound through a dynamic, intangible interface, movement qualities (MQ), and a virtual ecosystem (Figure 2).

2. BACKGROUND

2.1 Gesture and Sound

One approach in designing new musical interfaces is the use of gestures. With the help of machine learning, interactive systems are able to learn and recognise gestures, which are subsequently used to control sound [5].

Gesture recognition focuses on specific movements, like drawing a circle into the air, employing a variety of different algorithms. The biggest algorithmic trade off is the usability for real-time applications. For many cases a gesture is first completely recognised when finished. To be used in musical performances it is important to get a feedback as instantaneous as possible. It is also interesting to estimate the variance to a reference gesture in order to design the feedback. Many systems work with a kind of likelihood feedback, reporting the probability to be a certain movement. Some systems also report the phase of the gesture, allowing for example to control the progress of a played sound file. A good overview of different techniques is provided by [5].

Machine learning can also be used to classify different intentions of a movement [6]. Instead of focusing on a complete gesture, the *quality of the movement* can be extracted. The Laban Movement Analysis (LMA), after the dance theorist and analyst Rudolf Laban, is based on four main qualities: Body, Effort, Shape, and Space. Mentis and Johansson provide a good summary on the LMA, especially on the effort and present an approach using movement analysis to control sound [7].

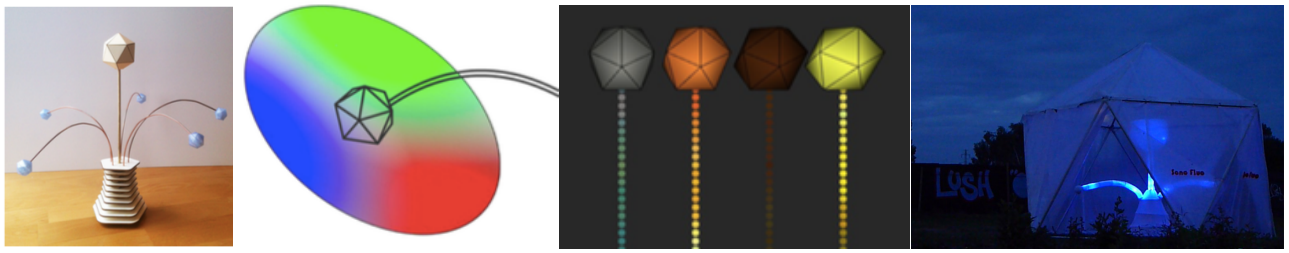


Figure 1. Different phases in the design refinement of SonoFluo.

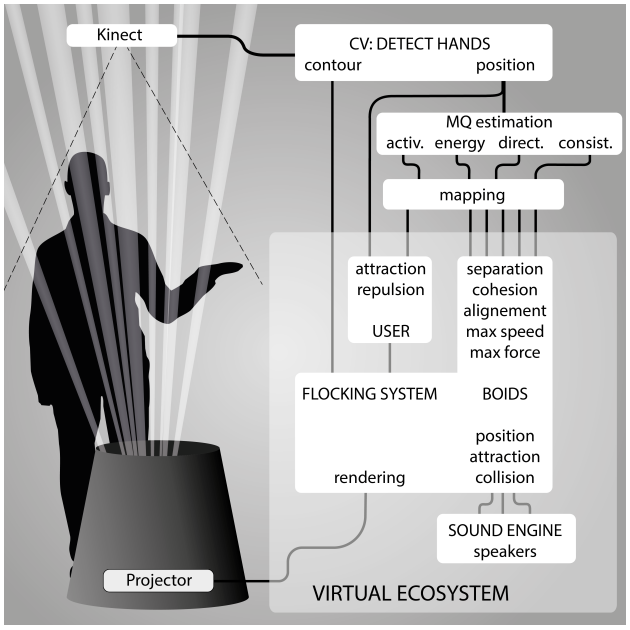


Figure 2. LichtGestalt: interaction with light-rays.

A depth camera is used in Mentis' and Johansson's research [7] to estimate and classify which of the basic elements was most present and play a 15 second sequence of especially for the element composed music. The system was evaluated with a LMA specialist and showed an accuracy of 67%. The audience however found the classification of movements in many cases not comprehensible and focussed on other qualities, like for example the energy put into the movement. The authors consider the use of LMA element as a too high level of classification for an untrained observer.

Physical modeling, as a sound synthesis method, has an important expressive potential. But the models are often considered difficult to control because of their complex dynamics and the high dimensionality of their parameter space [8], [9]. Alaoui et al, as an indirection, couple movement quality parameters to physical models (in their case mass spring systems) to create both visual and aural feedback for a dancer. This is of special interest, as physical models are in combination with a reasoning system are a good example of dynamical systems that facilitate plausible indirection between sound and movement. Another example is an agent-based system, which is described next.

2.2 Sound and Agents

A software agent is an autonomous entity that observes and acts upon an environment and directs its activity towards achieving goals. Agents, for instance, enhance the musical performance possibilities: different approaches where some of musical parameters are controlled by agents in parallel to the user have been described in [10]. In their implementation, agents acting in the same space manipulated virtual instruments based on physical models. In different scenarios, agents were influencing the outcome of an action with reaction, for example damping a string after it played or changing the string's properties while playing.

With the use of agents as procedures, the instrument takes on a life of its own and enhances the possibilities of the player. The installation *Room #81* is a good example, where an agent is used to create both a soundscape and light changes to frame the users interaction [11]. In different setting agents are guided by the user to produce different sounds when interacting with other objects in the game. In this case the agent is not just a trigger/control parameter but also visually represented on the screen.

Shacher and his colleagues recently suggested a classification of fundamental mapping relationships with the help of swarm simulations [12]. These include a number of strategies that relate to musical practice, highlighting the role of swarm simulations in mapping, especially in making the mapping relationships less predictable and more organic. Other studies on agents and sound include [13] and [14].

3. CONCEPT

In our installation, we consider an audio-visual composition as a complex virtual ecosystem embedded in its enactive landscape (see Figure 2). The user as part of the enactive landscape is able to directly manipulate this ecosystem.

The ecosystem is based on a flocking algorithm by Reynolds [15], which is represented in space through light-rays thrown by a projector. When the user enters the space of the ecosystem, he becomes a part of it, and the boids react towards him as a physical object. In parallel, the movement quality (MQ) is estimated in order to change the flocking behaviour and therewith the way the user interacts with it. The details of the MQ estimation have been presented in [1]. In parallel the position and

contour of the hands are translated into the virtual space of the flock in order to provide direct interaction possibilities with the boids.

Currently, the reaction of boids is considered as one of the three main conditions: *hunted, passive* and *curious*:
hunted is a basic reaction towards fast and directed movements of the user. The boids start to organise in swarms, and seek distance to the user.
Passive: when nothing of importance happens, the boids act independent (similar to a heard of wildebeests grazing). They move slowly, and seek distance to each other. The user has to move randomly and not to fast.
Curious: when moving very slowly, the boids become interested in the user and come closer (similar to insects seeking a light source).

To create the aural feedback, the position and acceleration of each boid is forwarded to a Max/Msp patch using OSC. After updating the flocking system, the boids are rendered and projected, which closes the circle (see Figure 2). Currently, all simulation and visual computing is done in C++ on openFrameworks[1], including the add-on openCV. The sound is generated in the environment of Max/MSP[2].

4. SONIC INTERACTION

In this section, we describe how we tackle the sonic interaction design of LichtGestalt. Our project is artistically driven and we do not rely on generative design approaches. For instance, we did not conduct design workshops or participatory design sessions. Yet, we find the action/sound relationships obtained as sonic interaction models in [16] relevant for our work. Especially, their *conducting* model that introduces the symbolic and semantic meaning of gestural interaction with sound becomes important when interacting with swarms in LichtGestalt. Conducting, with the help of dynamic control structures becomes a way to address the need of different levels between control and sound synthesis [17], [18].

4.1 Design Constraints

The user should easily understand how to interact with the virtual ecosystem. For simplicity, here we consider a case where the LichtGestalt is placed in a room akin to a large digital musical instrument, such as the ReacTable [19]. This is to shorten the process of audience interactions with generic artworks [20]. Because of the intangibility of the light rays, however, the generated sound in LichtGestalt should be very directly coupled to the visual representation. The user should get an idea when a sound occurs and what is generating it.

As a first design, we have considered three simplistic sound sources to experiment with the dimensionality of mapping, as suggested in [12]. Currently, not all of their parameters are used. Furthermore, they are all monophonic, in order to simplify the sound synthesis and explore the action/sound mapping. In later stages of design, we will also consider sound spatialization, as the position is the most common property in swarm visualization and auralization [13]

4.2 Density as a Design Dimension

Density is the degree to which something is filled, crowded or occupied. It is described as the quantity of something per unit measure. In our design, we consider the concept of density as a rich source of possibilities to combine the visual and the corresponding aural feedback. The *visual cues* regarding to the density are:
the amount of boids per area, visible as bunches of light-rays,
the closeness of boids in a swarm, measures as the amount of boids per the area of the swarm, visible as the intensity of the bunches, and
the amount of times the boids hit another object, visible as direction change in the movement.
These visual cues can be related to the sonic cues based on the following properties:
temporal: the amount of events per time, ranging from continuous to single events, from drumroll to single hits,
spectral; the spread of the sound in the spectrum, and
layers; a music is considered as more dense, when multiple instruments play simultaneously than when just a single does. This property is close to the spectral density, but can also be considered as multiple melody lines like for example in polyphonic music.

4.3 From Swarms to Sound

The tree conditions of swarm behaviour we have outlined in Sec. 3 serve as basic starting point in sound design. While hunted was hinting towards a climax, the other two conditions are considered as calmer soundscapes. In the hunting part, the boids organise in flocks. A computer vision based evaluation was programmed estimate the flock formation, together and with their density and group velocity.

Additionally, a temporal approach was implemented by the detection of collisions. Collisions happen from time to time and by distinction of different actors (boid/boid; boid/wall; boid/user) different sound qualities can be mapped towards them.

With the change of the condition either of the sound generations are addressed differently. This allows keeping the sound generation in general simplistic and static; only through the change of the general condition the user's focus switches towards the different feedbacks.

4.4 Sound Synthesis

So far, we have discussed the density coupling between three swarm behaviours and three sound sources. The first of these sound sources correspond to the swarms and implemented as a classical simple two-oscillator FM Synthesizer [21]. The other two correspond to collisions and

[1] http://openframeworks.cc
[2] https://cycling74.com/products/max/

implemented by physical models from the PeRColate library in Max/MSP: user/boid collusions trigger a marimba tone, whereas other collusions trigger a bowed bar tone. Specifically, *marimba* is triggered at a high pitch, by varying the pitch randomly over a half octave and the stick's hardness and position. In this way the created sound is always minimally varied. When multiple boids collide with the hands in very short time intervals, a ringing sound texture is created. The *bowedbar*, on the other hand responds to the collisions with the wall and creates a warm and deep sound, again with a random pitch in the range of three semitones. We next describe the continuous control of the swarms.

4.4.1 FM Synthesis of Swarms

Our system detects each group of boids as an individual flock, for each flock we use the same synthesis and mapping, so each flock has its own voice. For each swarm the system detects its velocity, its area, and the number of boids in it. By dividing the area through the number of boids, a measure of density is calculated and used to vary the amplitude of the modulator. The velocity is directly mapped to the frequency of the carrier. The loudness is calculated from the area covered by the swarm. The ratio between the modulator and the carrier is chosen non-integer (2.04) as it creates an interesting, gnarly sound.
The movement activity of the swarms thus reflects in the pitch change; if the flock moves slowly the pitch stabilizes. In states where no flocks are formed, no sound occurs. Instable conditions where flocks are formed for very brief moments create transient sounds. The creation of an individual voice per flock addresses the layered density, while the pitch change is connected to the spectral density.

5. EVALUATION

A test session was conducted with 11 subjects, in a room at the university over a time period of 3 days. As the installation is developed for a museum setting, this is not ideal. However, to find out how an individual subject reacts towards the installation and how she personally perceives it, the isolated setting seemed reasonable as it provides a controlled setting and better observation possibilities of the procedure. During this test session, the subjects were asked to investigate the installation and to fill out a questionnaire afterwards. In order to determine how consciously the subjects could control the installation, the subjects were asked to reproduce specific reactions. The gathered data was qualitatively evaluated. The three stages of the test procedure are described below.

5.1 Investigation of the installation

The subjects entered the room and were asked to explore the installation. They were left to explore freely, until they stopped by themselves or seemed to run out of ideas, repeating the same movements. The subjects were observed in terms of where they looked and which reactions the installation produced upon their actions. The observations were captured with a standardised form.

5.2 Questionnaire about the experience

After the investigation the test subjects were asked to fill out a questionnaire. The questions were designed towards following topics:

- Personal questions, to find out if the subject has any knowledge and background relevant to the evaluation.
- A question about how the different reactions were perceived, assuming the subjects recognise different reactions.
- Questions towards the sound and the connection towards to the light rays, and how the touch of the light rays was experienced.
- A question about the control experience; what could be controlled, and if they have an idea of how the system works.
- Through a question about the associations, the focus was shifted towards the aesthetics.
- Questions regarding the installation's sound aesthetics and visual appearance, their personal engagement and general impression.

Seven of the questions, especially those regarding aesthetics and interests, were posed with a Likert psychometric scale model, to provide a common measure. The subject was asked to rate within a range of 1 to 10; 1 standing for *not at all,* while 10 meant *quite a lot*/*very*. The liker-scale answers were followed by a text-field with the request to comment.

5.3 Performance session

The performance test was an experimental approach to see if the subjects were able to reproduce requested reactions. This was attempted, due to the fact that the vocabulary, which describes one's experience, is often difficult to find. With the performance session, a specific reaction could be addressed and the result directly be observed.

The requested tasks were designed after specific characteristics of the three conditions, *hunted*, *passive* and *curious*. They were as follows:

- *Can you collect the light rays?* There were two solutions to solve this task: With slow movement, the *passive* condition would allow to gather them in a corner, or when keeping the hand still, the *curious* condition would make the boids become attracted.
- *Can you move the light rays around?* This task can be seen as a follow up to the previous and if they are able to play with the conditions given. When in passive mode, they already had to move conscious. When using the *curious* condition, the hand could be moved very slowly and the boids would follow.
- *Can you make the installation create the high pitch sound texture?* The high pitch sound texture is just created in the *hunted* condition. If they apply the *curious* condition the drone had a lower pitch, so they were additionally asked if they could modulate the pitch.
- *Can you create the high pitch bell like sound?* This was an attempt to see if the user would understand that the collisions with the light rays are causing the sound.

- *Do you have any idea when the lower pitch percussive sound occurs?* The last is not really a task, as it does not expect any action of the user. The question was posed to see if the test subjects understood the connection between collisions with the edges and the therewith trigger sound.

6. RESULTS

Seven of the eleven participants reported some kind of technical background or have had some previous experience with interactive installations. Their age was widely spanned from 21 to 56 with an average age of 36 years. Three of the participants have heard about the installation before, while just one of them had some closer knowledge about the system. However, their results show no significant difference to the other participants. Their data are therefore evaluated equally with the others. The approximate duration of each session was about 30 minutes, most of the time taken by the questionnaire.

All qualitative data obtained has been coded by the first author, and complemented with his observations during the investigation and performance. The quantitative results are meant to complement the coding, by themselves they not reliable due to small sample size; nor it was the aim to gather statistics for interaction with an experiential installation like the LichtGestalt. We nevertheless present them in the sequel, as they are easier to summarize and draw conclusions compared to a qualitative data and observations. The full data including codes and implications on the design loop, and user associations such as "*Very romantic; Hot summer night and bugs flying. Quiet space in a not polluted nature by water*" or "*I tried to play a wizard, waving his hands to shoot out fireballs…*" can be assessed from the master thesis of the first author [1].

6.1 Aesthetics

Under the topic aesthetics, the answers from the questions regarding the sound, the visual appearance and their interplay are evaluated (see Fig. 3). The visual appearance was rated rather positively; the reasons behind the ratings varied considerably. Some mentioned the subtle and simplistic design as very pleasing, while others just call it nice. Negative points are regarding the projection on the ceiling and other possible improvements.

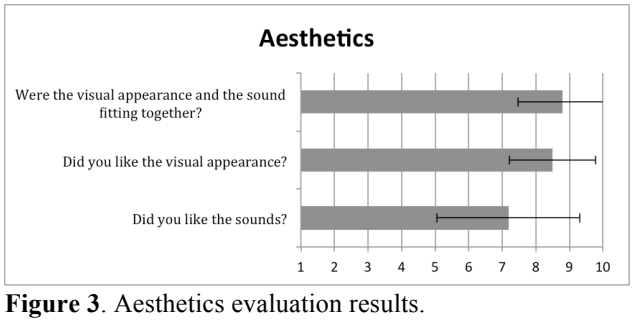


Figure 3. Aesthetics evaluation results.

The sound was slightly polarizing: for some there was not enough variation, other disliked the drone texture. While one found it too dull, another mentions a relaxing quality as positive. In general, the different layers of the soundscape are perceived as interesting and complementary. The interplay between the sounds and the visual appearance, on the other hand, may show how interaction fidelity may modulate even crude sound designs.

6.2 Experience

The answers regarding the experience and engagement were quite positive, as illustrated on Fig. 4. Many participants mentioned the different reactions to be explored as the main motivation, while others found the abstract design nice to give space for imagination. Also the sensation of manipulating sound is mentioned several times. The question about a possible recommendation to friends is borrowed from [3]: it gives a different angle about the experience.

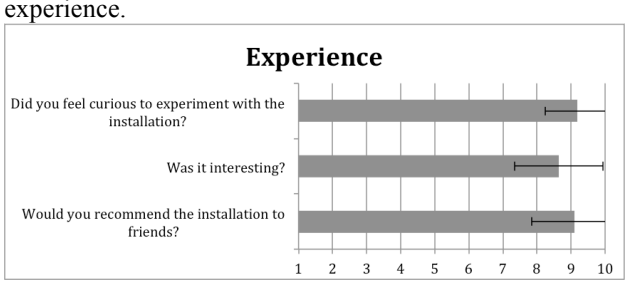


Figure 4. Experience evaluation results.

6.3 Control

The answers to the control question were positive but not consistent (Fig. 5). Control, as an abstract measure, seems subjectively perceived. When asked for elaboration, some of the subjects report less control due to the semiautomatic behaviour of the boids, while others seem to master the dynamics of the interaction with high ratings.

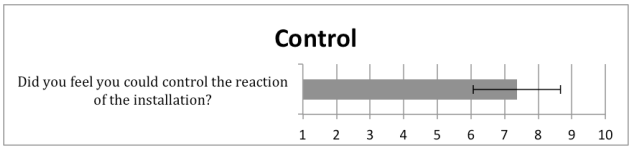


Figure 5. Control evaluation results.

7. CONCLUSIONS AND FUTURE WORK

We presented the LichtGestalt: a new interactive sound installation to be explored by movement via a dynamical system (a synthetic flock of agents). In the installation, the relationship between gesture and sound are therefore mediated by synthetic swarms of light rays. We have presented the sonic interaction design of the installation, which uses density as a design dimension and maps the swarm parameters to the sound synthesis parameters.

The system as a running installation has been tested on 11 subjects at the university. A qualitative evaluation of the results, which was not discussed here in detail, has indicated that the general approach is promising. The evaluation highlights also the potential of the system to engage the user, as it seems pleasing to explore and discover different dynamic reactions.

However, the evaluation has also disclosed some potential improvements: the current swarm behaviors and sound models can be extended based on the presented groundwork in the future. Also the swarms can be spatialized, and more elaborated sound synthesis and sonic in-

teraction design [22] can be incorporated in LichtGestalt installation.

8. REFERENCES

[1] J. Fehr, C. Erkut. "Indirection Between Movement and Sound in an Interactive Sound Installation", accepted for publication in MOCO'15, Vancouver, BC, Canada. See also Fehr, J. (2015, June 16). *LichtGestalt: An interactive audio-visual composition*. Master's Thesis, Aalborg University, Copenhagen. http://projekter.aau.dk/projekter/en/

[2] C. Erkut, S. Serafin, J. Fehr, H. M. R. F. Figueira, T. B. Hansen, N. J. Kirwan, and M. R. Zakarian, "Design and evaluation of interactive musical fruit," Proc. Interaction design and children, Aarhus, Denmark, 2014, pp. 197–200.

[3] B. Bengler and N. Bryan-Kinns, "Polymetros," *interactions*, vol. 21, no. 3, May 2014.

[4] K. Franinovic and C. L. Salter, "The Experience of Sonic Interaction," in *Sonic Interaction Design*, Chapter 2, K. Franinovic and S. Serafin, Eds. Cambridge, MA: MIT Press, 2013, pp. 39–76.

[5] B. Caramiaux and A. Tanaka, "Machine learning of musical gestures," Proc. NIME, London, UK, 2013, pp. 513–518.

[6] D. S. Maranan, S. F. Alaoui, T. Schiphorst, P. Subyen, L. Bartram, and P. Pasquier, "Designing for movement," Proc. Conf. Human Factors in Computing Systems, New York, New York, USA, 2014, pp. 991–1000.

[7] H. Mentis and C. Johansson, *Seeing movement qualities*. Paris, France: ACM, 2013, pp. 3375–3384.

[8] S. F. Alaoui, C. Henry, and C. Jacquemin, "Physical modelling for interactive installations and the performing arts," *International Journal of Performance Arts & Digital Media*, vol. 10, no. 2, pp. 159–178, Sep. 2014.

[9] S. F. Alaoui, C. Jacquemin, and F. Bevilacqua, "Chiseling bodies," Proc. Conf. Human Factors in Computing Systems, Paris, France, 2013, p. 2915.

[10] B. Schroeder, M. Ainger, and R. Parent, "A Physically Based Sound Space for Procedural Agents," Proc. NIME, Oslo, Norway, 2011, pp. 120–123.

[11] N. dAlessandro, R. Calderon, and S. Muller, "ROOM#81 - Agent-Based Instrument for Experiencing Architectural and Vocal Cues," Proc. New Interfaces for Musical Expression, Oslo, Norway, 2011, pp. 132–135.

[12] J. C. Schacher, D. Bisig, and P. Kocher, "The Map and the Flock: Emergence in Mapping with Swarm Algorithms," *Comp. Music J.*, pp. 1–15, Aug. 2014.

[13] T. Blackwell, "Swarming and Music," in *Evolutionary Computer Music*, no. 9, E. R. Miranda and J. A. Biles, Eds. London: Evolutionary Computer Music, 2007, pp. 194–217.

[14] A. Eigenfeldt and P. Pasquier, "A Sonic Eco-System of Self-Organising Musical Agents," Proc. EvoApplications 2011, 2011, vol. 6625, pp. 283–292.

[15] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, Aug. 1987.

[16] B. Caramiaux, A. Altavilla, S. Pobiner, and A. Tanaka, "Form Follows Sound: Designing Interactions from Sonic Memories," Proc. Conf. Human Factors in Computing Systems, 2015.

[17] C. Heinrichs, A. McPherson, and A. J. Farnell, "Human performance of computational sound models for immersive environments," *The New Soundtrack*, vol. 4, no. 2, pp. 139–155, Sep. 2014.

[18] C. Erkut, "Modular interactions and hybrid models: a conceptual map for model-based sound synthesis," Proc. EURASIP, Antalya, Turkey, 2005.

[19] S. Jordà, "On stage: the reactable and other musical tangibles go real," *IJART*, 2008.

[20] E. A. Edmonds, "Human Computer Interaction, Art and Experience," in *Interactive Experience in the Digital Age*, Chapter. 2, L. Candy and S. Ferguson, Eds. Springer International Publishing, 2014, pp. 11–23.

[21] P. R. Cook, *Real Sound Synthesis for Interactive Applications*. AK Peters, 2002.

[22] X. W. Sha, A. Freed, and N. Navab, "Sound design as human matter interaction," Proc. Conf. Human Factors in Computing Systems, Paris, France, 2013, pp. 2009–2018.

List of Authors

Alonaso, P., 337
Angelini, Ivana, 31
Armstrong, Newton, 265
Assayag, Gérard, 427
Avanzini, Federico, 31, 161

Baratè, Adriano, 287
Berndt, Axel, 91
Bertsch, M., 407
Bettineschi, Cinzia, 31
Bianchi-Berthouze, Nadia, 485
Bilbao, Stefan, 5
Bouche, Dimitri, 427
Boyd, Jeffrey E., 7
Breen, Aidan, 125
Bresson, Jean, 257, 427
Brown, Stephen, 317
Butković, Ana, 329
Böck, Sebastian, 241

Cabrera, Andres, 439
Cambraia, Pedro, 509
Canadas-Quesada, F., 337
Canadas-Quesada. Fransisco J., 491
Canazza, Sergio, 31, 221, 351
Carvalho Jr., Antonio D., 209
Chemillier, Marc, 427
Corona, Humberto, 363
Cortina, Raquel, 491
Crowley, Katie, 503
Cullimore, Jason, 413

Da Pos, Osvaldo, 351
Dalton, Nick, 15
De Man, Brecht, 147
De Poli, Giovanni, 31, 351
Del Piccolo, Andrew, 309
Delle Monache, Stefano, 309
Demir, Abdullah Onur, 301
Deotto, Giulia, 31
Ding, Jianhang, 393
Dixon Simon, 387
Dobashi, Ayaka, 99
Dong, Lu, 393
Dzhambazov, Georgi, 281

Eagle, David, 7
Egloff, Deborah, 169
Erkut, Cumhur, 49, 527
Eyal, Alon, 141

Fantozzi, Carlo, 31
Faresin, Emanuela, 31
Fazenda, Bruno, 463
Fehr, Jonas, 527
Fitzgerald, Derry, 3
Fober, Dominique, 229
Fohl, Wolfgang, 419
Fontana, Federico, 161
Franco, Ivan, 169
Freire, Sérgio, 509
Frid, Emma, 169
Fukayama, Satoru, 177
Fukayama,Satoru, 61
Fukuda, Tsubasa, 105
Furlong, Dermot, 477

Georgaki, Anastasia, 77
Gerhard, David, 413
Giordano, Marcello, 169, 521
Gold, Nicholas, 485
Goto, Masataka, 23, 61, 153, 177
Gouilloux, Guillaume, 229
Goulart, Antonio, 193
Grbac, Franco, 235
Großhauser, Tobias, 407
Gómez, Emilia, 371

Hacıhabiboğlu, Hüseyin, 301
Hadjakos, Aristotelis, 91
Hamasaki, Masahiro, 23
Haron, Anis, 343
Harte, Christopher, 295
Hattwick, Ian, 169
Haus, Goffredo, 379
Hegg, Jens, 497
Herrera, Perfecto, 111
Hirai, Tatsunori, 153, 323
Holland, Simon, 15
Huber, Stefan, 69
Hörschläger, Florian, 241

Iijima, Kosuke, 435
Ikemiya, Yukara, 99, 105, 153
Itoh, Takayuki, 43, 55
Itoyama, Katsutoshi, 99, 105

Jillings, Nicholas, 147
Jorda, Sergi, 111
Järveläinen, Hanna, 161, 185

Kanno, Saya, 55
Kato, Jun, 61
Kinoshita, Naohiro, 515
Kirwan, Nicholas John, 49
Kitahara, Tetsuro, 435, 515
Kleimola, Jari, 249
Knees, Peter, 241
Kreković, Gordan, 235, 329
Kreković, Miranda, 235
Kurihara, Takuya, 515
Kurita, Takio, 401
Lambert, Andrew J., 265
Lamontagne, Valerie, 169
Larkin, Oliver, 249
Lazzarini, Victor, 193
Letz, Stephanie, 229
Li, Juan, 393
Loughran, Roisin, 273
Ludovico, Luca Andrea, 287
Luz, Rosalía Soria, 37
Lähdeoja, Otso, 85

MacCallum, John, 257
Malavolta, Lorenzo, 161
Mandanici, Marcella, 221
Manzolli, Jônatas , 215
Martins, Mario, 455
Martinucci, Maurizio, 169
Mauro, Davide Andrea, 309, 379
Mayer, Thomas, 209
McDermott, James, 273, 503
Menegazzi, Alessandra, 31
Menendez-Canal, Jonatan, 491
Middleton, Jonathan, 497
Moffat, David, 147
Molin, Gianmario, 31
Morishima, Shigeo, 153, 323
Mudd, Tom, 15
Mulholland, Paul, 15
Murari, Maddalena, 351
Murphy, Damian, 141, 359

Nakano, Tomoyasu, 23, 153
Neff, Patrick, 185
Newbold, Joseph, 485
Nika, Jérôme, 427
Nolgaski, Malte, 419
Nomura, Ryo, 401

O'Leary, Sean, 471
O'Mahony, Michael, 363
O'Neill, Michael, 273
O'Riordan, Colm, 125
Okada, Misaki, 435
Oliver, Jorge, 317
Orlarey, Yann, 229
Overholt, Dan, 49
Ozono, Tadachika, 133

Papetti, Stefano, 161, 309
Pearce, Marcus, 295
Percival, Graham, 61
Pontes, Vânia Eger, 215
Popa, Iulius A.T., 7
Popp, Constantin, 37
Pošćić, Antonio, 329
Presti, Giorgio, 379
Pretto, Niccolò, 31

Queiroz, Marcelo, 77, 193

Ranilla, J., 337
Reiss, Josh, 147
Rimoldi, Gabriel, 215
Roads, Curtis, 439
Robel, Axel, 69
Robertson, Ben, 497
Rocchesso, Davide, 309
Roddy, Stephen, 477
Rodriguez-Serrano, Fransisco J., 491
Rodà, Antonio, 31, 221, 351
Rosli, Muhamma H.W., 439
Ruiz-Reyes, N., 337

Salamon, Justin, 371
Salemi, Giuseppe, 31
Salter, Christopher, 169
Sasaki, Shoto, 323
Schacher, Jan C., 185
Schubert, Emery, 351
Schwarz, Diemo, 471
Serra, Xavier, 281
Shintani, Toramatsu, 133
Shun Shiramatsu, 133
Silla Jr., Carlos N., 455
Smith, Jordan, 61
Song, Chunyang, 295
Song, Yading, 387
Soria Luz, Rosalia, 201
Strinning, Christian, 185

Takamura, Hiroya, 55
Tavares, Tiago F., 215
Thalmann, Florian, 119
Thul, A., 407
Timoney, Joseph, 193
Troester, G., 407
Truax, Barry, 1
Tsuruoka, ayaka, 435

Uehara, Misa, 43

Valero-Mas, Jose J., 371
Vera-Candeas, P., 337
Vidal, Antonio, 491
Vogl, Richard, 241

Waloschek, Simon, 91
Wanderley, Marcelo, 169
Wanderley, Marcelo M., 521
Weeter, Jeffrey, 359
Weyde, Tillman, 265, 301
Williams, Amanda, 485
Wilson, Alex, 463
Wright, Matt, 343
Wright, Matthew, 439

Yamaguchi, Ryunosuke, 515
Yamashita, Yuji, 435
Yang, Xinyu, 393
Yoshii, Kazuyoshi, 99, 105
Yoshii,Kazuyoshi, 153
Young, Gareth William, 359

Zanovello, Paola, 31

Ó Nuanáin, Cárthach, 111

sfi Science Foundation Ireland
Movidius
Dave Smith INSTRUMENTS
XILINX ALL PROGRAMMABLE™
iZotope®
MSB MOST SIGNIFICANT BIT MASTERING
Ardmore SOUND FILM SOUND IN IRELAND
rocudo™ music machines
HOTONE DESIGN INSPIRATION
MAKER.IE
I-CUBEX
FAW
imro Cherishing Music In A Changing World
supporters of MUSIC MATTERS WHYMUSICMATTERS.ORG
Fáilte Ireland National Tourism Development Authority
meetinireland.com

