INTUNE: A MUSICIAN'S INTONATION VISUALIZATION SYSTEM

Kyung Ae Lim
School of Informatics
Indiana University
kalim@indiana.edu

Christopher Raphael
School of Informatics
Indiana University
craphael@indiana.edu

ABSTRACT

We present a freely downloadable program, *InTune*, designed to help musicians better hear and improve their intonation. The program uses the musical score from which the musician plays, assisting our approach in two ways. First, we use score following to automatically align the player's audio signal to the musical score, thus allowing better and more flexible estimation of pitch. Second, we use the score match to present the tuning analysis in ways that are natural and intuitive for musicians. One representation presents the player with a marked-up musical score showing notes whose pitch characteristics deserve closer attention. Two other visual representations of audio overlay a musical time grid on the music data and allow random access to the audio, keyed by musical time. We present a user study involving 20 highly educated instrumentalists and vocalists.

1. INTRODUCTION

One of our most beloved music teachers was a forceful advocate for "facing the music," by which he meant listening to recordings of our playing. As with the first hearing of one's voice on recording, we each wondered "do I really sound like that!?" While sometimes revealing more than we were ready to hear, the long-term effect of this exercise helped us to "hear ourselves as others hear us." Thus armed, we initiate practice habits that, perhaps over many years, move our music-making toward a state we might admire in others.

The "face the music" approach begins by accepting that most of us are not born able to judge ourselves objectively, but can learn to do so when given the proper external perspective. We adopt this approach here, still in the service of music education, though we use visual, in addition to aural, feedback. While a visual representation of audio is necessarily an abstraction, it has the advantage that the observer can "visit" the image according to her will. For instance, she may see a note having a certain undesirable (or desirable) property; find the same trait in another note of the same pitch; formulate a hypothesis of systematic error (or accuracy); and validate or refute this theory on subsequent notes. In contrast, audio data must be digested nearly at the rate it comes into the ear.

We apply the "face the music" approach to the practice of intonation — the precise tuning of frequencies corresponding to different musical pitches. While good intonation, "playing in tune," is often neglected in the earliest years of musical practice, it is as essential a part of technique as the playing of fast notes or the control of emphasis. Intonation is also central to what some see as the illusion of tonal beauty — that is, for a sound to be beautiful it must commit itself clearly to the "correct" pitch. We introduce a system that allows musicians to visualize pitch in ways that leverage the centuries-long tradition of music notation, and are intuitive to the non-scientist.

The electronic tuner is, without doubt, one of the most widely used practice tools for the classically-oriented musician, thus justifying efforts to improve this tool. The tuner provides an objective measurement of the pitch or frequency with which a musician plays a note, which can be judged in relation to some standard of correctness (say equal tempered tuning at A=440 Hz.). Though the tuner has been embraced by a large contingent of performing musicians, it does have its weaknesses, as follows. The tuner gives only real-time feedback, requiring the user to synthesize its output as it is generated. The tuner takes time to respond to each individual note, making it nearly impossible to get useful feedback with only moderately fast notes. The tuner cannot handle simultaneous notes, such as double stops — this is actually part of the reason the tuner fails on fast notes, since past notes linger in the air, thus confusing the instrument. Perhaps most significantly, the tuner does not relate its output through the usual conventions of notated music, thus hiding tendencies and patterns that show themselves more clearly when presented as part of a musical score. Our program, *InTune* seeks to overcome these weaknesses by presenting its observations in an intuitive and readily appreciated format.

SMC 2009, July 23-25, Porto, Portugal

In what follows, we present our system, *InTune*, describing the three different views of audio the program allows, as well as the backbone of score-following that distinguishes our approach from others. We consider other approaches to this problem and place ours appropriately in this context. Finally we present a user study, giving reactions to our effort from a highly sophisticated collection of users. The program was developed in close consultation with two professors emeriti of music in the School of Music at our university, and is freely available for download at http://(removed for review, program and files can be provided upon request).

2. PAST WORK

We know of two recent examples addressing computer-assisted music instruction on intonation from the computer music community: [12], [13], though these efforts address several other performance aspects, including dynamics, rhythm, articulation, etc. Of these two, [13] shares our use of score following, though their use is based on on-line recognition, and thus is somewhat limited in its ability to relate its measurements to the musical score. We make analogous use of on-line recognition for real-time feedback, but focus mostly on off-line alignment, due to its greater accuracy and appropriateness for the off-line nature of the performer's analysis of a performance [12] shares some of the basic kinds of displays as our work, though the effort is restricted to the playing of long tones, rather dramatically restricting its reach. This work also does not relate the results to a musical score, thus shifting the burden of interpretation to the musician.

There has been significantly more commercial interest along these lines as exemplified by [1], [2], [3], [7], [14], [15]. The basic kinds of visual music display we use are found in the cited examples as well. [2] [15] uses the spectrogram, [3] use pitch trace representation, and [7], [14] annotates a musical score to reflect a specific performance. However, there are some important ways in which we differ from these efforts. Most important is our use automated score alignment, which allows us to relate the music data directly to our score representation. While [7] and [14] use traditional music score display, they relate the music data to the score by requiring the player to play along with a rigid accompaniment. While this "solves" the alignment problem, it imposes a foreign constraint on the musician for typical intonation practice. Other cited efforts either prompt the musician to play specific notes, or try to estimate the notes of the musical score from audio.

The most significant difference between our work and these cited is our use of score alignment as the fundamental means of relating our measurements to the music itself. Using score alignment we can link our three representations, thus allowing the user to move freely between them while retaining focus on the current position or note. An additional difference between our work and [7], [13], [14] is our deliberate effort not to grade the musician's performance, but rather to give them the objective feedback needed by the musician in reaching independent conclusions.

3. ESTIMATING PITCH

The backbone of *InTune* is a score-following system that aligns the audio input with a musical score. Thus we assume the musician plays from a known score. We base our approach on score following since the quality of blind (no score) music recognition degrades rapidly as complexity increases — we know of no blind recognition approaches, including our own, that produce good enough results for our task at hand. Furthermore, we wish to present our feedback in the context of the musical score. Since the score must be known for this to happen, we might as well put this knowledge to good use.

Our score following is based on a hidden Markov model, as documented in [10]. This model views the audio input as a sequence of overlapping frames, with about 30 frames per second, which form the observable part of the HMM, $y = y_1, y_2, ..., y_N$. We construct small (10-or-so-state) Markov models for each score note modeling, among other things, the distribution of the number of frames devoted to the note. These sub-models are concatenated together, in "left to right" fashion, to form our state graph. The hidden Markov chain, $x = x_1, x_2, ..., x_N$, corresponds to the path taken through this state space. Given audio data and a musical score, we perform alignment by computing the onset time of each score note, $\hat{n}_i$ as

$$\hat{n}_i = \arg\max_n P(x_n = \text{start}_i \mid y_i, ...y_N)$$

where start_i is the unique state that begins the *i*th note model. This approach performs well when confronted with the inevitable performance errors, distortions of timing, and other surprises that frequently occur in musical practice, and has been the basis for a long-standing effort in musical accompaniment systems [11].

Our score following approach tells us when the various notes of the musical score occur, thus giving us the approximate pitch for each frame of audio. That is, if the score match designates a frame to belong to a score

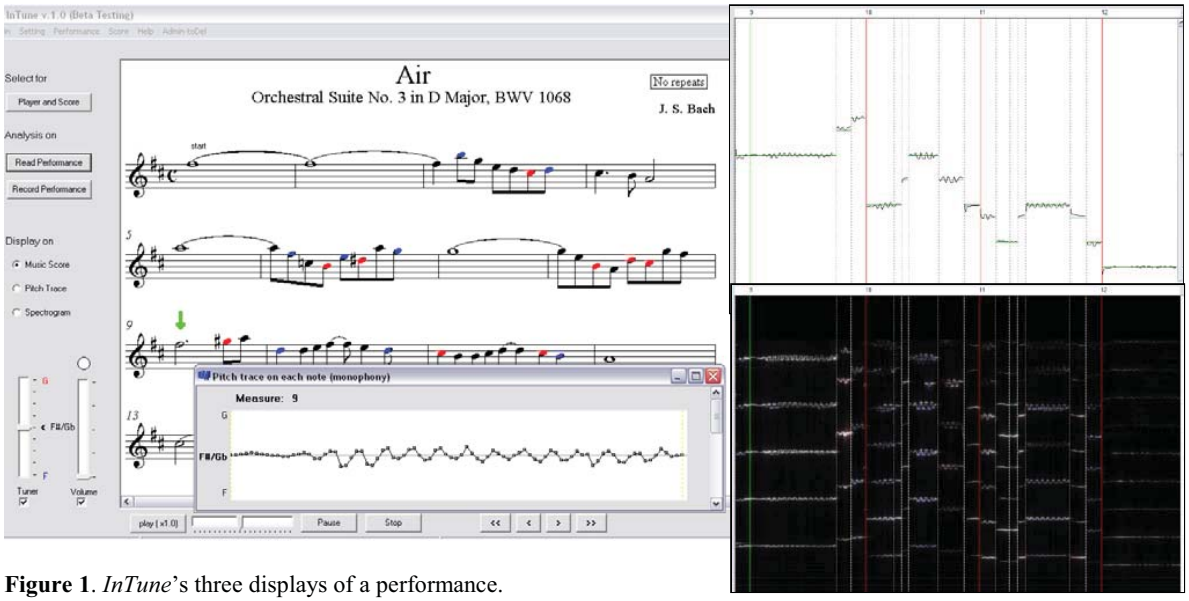

Figure 1. *InTune*'s three displays of a performance.
The score image (left) colors the heads of "suspicious" notes. The pitch trace (upper right) shows precise pitch evolving over time. The spectrogram (lower right) is the traditional display of frequency content evolving over time. The vertical lines show the measure or beat boundaries (red), the note boundaries (white), and the current position (green).

event having MIDI pitch m, then the pitch of the note is approximately

$$f(m) = 440 \times 2^{(m-69)/12} \quad (1)$$

Hz. Such knowledge makes it much easier to estimate the pitch more precisely. In fact, many pitch estimation and tracking approaches suffer more from coarse pitch errors on the octave level (misestimating by a factor of two), than from the fine tuning of pitch [6].

The frequency is defined here as the instantaneous rate of change of phase; we estimate the frequency for a particular frame by approximating this derivative. This is a timehonored and intuitive approach dating back to [8]. If $Y_n(k)$ is the windowed finite Fourier transform of y_n at frequency k, we estimate the frequency as

$$\hat{f}_n = \frac{k/r + [\phi(Y_{n+1}(k)) - E(k) - \phi(Y_n(k))]/2\pi}{\Delta t}$$

where r is the frame overlap rate, $\phi()$ is the argument or angle of a complex number, $E(k) = \frac{2k\pi}{r} \bmod 2\pi$ is the deterministic phase advance of frequency k between frames, and Δt is the time, in seconds between frames. The numerator in this calculation simply computes the fractional number of cycles that have elapsed for frequency k, which is then divided by the elapsed time to get cycles per second.

Since we know the nominal score pitch of the current note from our score match, our choice of k is not too difficult. If there is sufficient energy around the fundamental frequency we take k to be the frequency "bin" in the neighborhood of the fundamental having greatest energy. Otherwise we scan the neighborhoods of the lowest 4 or 5 harmonics seeking the bin having the greatest amplitude. If this bin corresponds to the *h*th harmonic, we must divide our frequency estimate by *h* to estimate the fundamental frequency. Thus our pitch estimation algorithm functions well when several of the lowest harmonics have little or no energy. When no harmonic seems to have any significant amount of energy, we assume the player is not generating any sound at the moment, and do not estimate frequency in this case.

4. THE THREE VIEWS OF INTUNE

On bringing up the program, the musician begins by choosing a piece to work on, at which point standard music notation is displayed. While *InTune* begins with a small collection of ready-made pieces, MIDI files can be imported, thus extending the program's range to nearly anything playable by a single instrument. The player then selects a range or excerpt from the piece and records a performance. The audio is then automatically aligned to the score, followed by pitch estimation, as described above. This information is then displayed to the musician in a collection of three linked views, as shown in Figure 1.

All three views use the notion of equal tempered tuning as reference point. For instance, if we choose A = 440 Hz as our pitch level, then the reference frequency of MIDI pitch *m* would be as given in Eqn. 1, (69 is the MIDI pitch for the "tuning" A). The location of the tuning A is adjustable by the user. We acknowledge here that there is no single "correct" view of tuning. For example, in many situations it is common to prefer tuning based on simple integer ratios, such as 3:2 for a perfect 5th. In addition, some players advocate various kinds of "expressive tuning" such as the raising of leading tones, or bending pitches in the direction of future notes. We choose equal temperament as our reference due to its simplicity and wide acceptance—not to assert its correctness. Users of the program can easily make their own judgments of the desirability or accuracy of the tuning based on this reference point without necessarily "buying in" to equal temperament. In fact, the importance of displaying, rather than judging, the tuning results was a basic tenet of ours, due to the lack of any single agreed-upon yardstick.

The *score view* is immediately presented by the program after a recording is made. This view employs a mark-up of the music notation, coloring notes whose mean frequency differs by more than a (user-adjustable) threshold from the equal tempered standard. We use red for high or "sharp" notes and blue for low or "flat" ones, due to their implications of hot and cold. The coloring of notes gives an easy-to-assimilate overall view of the performance that may show tendencies of particular notes or parts of phrases, such as the undesirable change in pitch that can accompany a change in loudness on some wind instruments. Clicking on any note in the score view opens a window that graphs the pitch trajectory over the life of the note. This aspect gives higher-resolution pitch detail, allowing one to see the tuning characteristics of vibrato, as well as variation associated with the attack or release of a note. Visualization of vibrato was of particular interest to the music faculty with which we developed this project.

Of course, one cannot appreciate the most important dimension of the performance without sound, so the score view (as well as the others) allows audio playback that is mirrored as a moving pointer in the image display. Variable-rate playback through phase-vocoding [5] allows the truly brave user to hear details of the performance often lost at the original speed. Since we have aligned the audio to our musical score the user can play back the performance beginning with any note, and at any speed, thus allowing random access to the audio and enabling more focused listening than normally possible with audio.

A second view of the audio data is called the *pitch trace* (top right of Figure 1). This representation is analogous to a piano roll graph in which notes are represented as horizontal lines whose height describes the note's pitch and whose horizontal extent shows the time interval where the note sounds. Typically, one uses a log scaling of frequency in a piano roll graph so that each octave (or any other interval) corresponds to a constant amount of vertical distance. We modify this graph simply by allowing the lines to "wiggle" with changing pitch. To make the graph more intelligible we mark measures, beats, or some other musical unit of the user's choosing, with vertical lines, courtesy of the score alignment. As with the score view, the user is free to page through the notes and to play the audio starting from the current location.

The final view (bottom right of Figure 1) is a traditional *spectrogram*, in which we show frequency energy on the vertical axis evolving over time on the horizontal axis. Except for the use of color to denote notes with suspicious tuning, and vertical lines to mark musical time units, this view presents an uninterpreted view of the raw data. To some extent, one can make judgments about timbre by the proportions of energy in the various "harmonics" of a note (integral multiples of the fundamental frequency). In user tests we have found a number of musicians to be particularly fascinated with this data view, since it seems to support concrete assertions about the seemingly intangible world of timbre.

5. USER STUDY

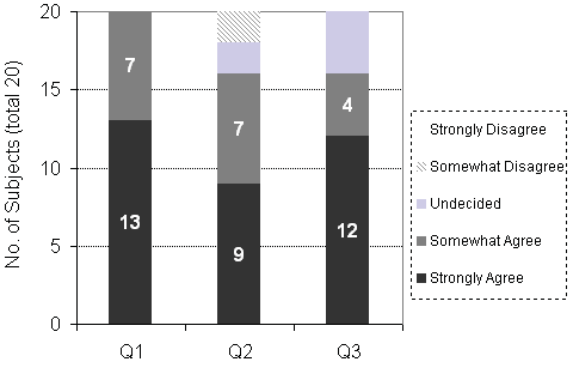

Figure 2. User study response.

We performed a small 20-subject user study using undergraduate and graduate performance majors from the School of Music. The school is one of the very best music conservatories in the US, yielding a musically sophisticated collection of subjects. The subjects consisted of a mixture of woodwind, brass, and string players, as well as two vocalists.

The students were directed to perform some simple tasks using the program, involving playing their instruments, recording, as well as visualizing and hearing their audio data. The students then responded to a questionnaire assessing their belief about the usefulness of *InTune* and their interest in incorporating the program into

their practice. Overall, the students we quite positive about the the program, with most saying they would incorporate *InTune* into their practice if it were available. Figure 2 summarizes the response in the most illuminating questions:

Q1 Did *InTune* help you recognize inaccuracies you did not hear?
Q2 Is *InTune*'s sense of intonation consistent with your own?
Q3 Would you use *InTune* with your practice when it is available?

We were most pleased with the musician's willingness to use the program in actual practice, and hope that professed willingness holds true.

Several themes emerged through the written and voiced comments that accompanied the study. Virtually all perceived the program as an improvement over the tuner, though acknowledging the difficulty of carrying a laptop to the practice room. This improvement was primarily due to the possibility of scanning and studying past pitch histories while making these data accessible by relating them to the musical score. Players also commented on the program's facile and informative handling of fast notes. Some players found the program gave especially useful feedback on vibrato, by allowing one to clearly see the width of pitch excursions.

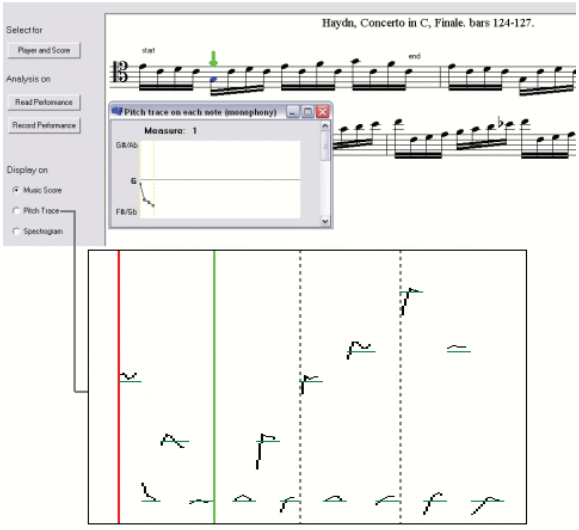

Figure 3. Example showing pitch estimation for fast notes.

The program did not perform as well with the vocalists who generally sing with accompaniment, thus giving an external pitch reference. The singers' overall pitch level tended to drift and all agreed that they should not be "marked down" for this. Another criticism repeated by several musicians addressed the note-oriented view of pitch. They observed that musicians often spend time "between" notes, and found the program's pitch estimation wanting in this scenario. We admit that our view of pitch estimation simply did not take this phenomenon into account, most common with singers and string players. In essence, we gain a significant advantage by assuming the player's pitch is close to the notated pitch, allowing accurate handling of otherwise difficult situations, though this gain does not come without cost. Several players argued for the value of expressive *context-dependent* tuning, not recognized by the program. In spite of our efforts no prescribe the correct answers, it seems inevitable that some may interpret the program's output this way.

Figure 3 shows an example from the user study of a horn playing a section of the Haydn Cello Concerto. The notes here are fast enough so that a tuner would provide little use, while accurate recognition from pure audio would be challenging and, likely, unreliable.

One especially interesting example occurred with a graduate flute major whose pitch data are shown in Figure 4 on the 2nd movement of the Mozart Clarinet Quintet, K. 581. In these data she observed a rising pitch trend in the early life of many notes. Our teacher's "face the music" maxim seemed to reverberate when she commented that the program had pointed out a tendency that she was unaware of, but could now hear. This example demonstrates the utility of automatic score alignment. The audio for this and all other examples can be heard at http:// (removed for review, files can be provided upon request).

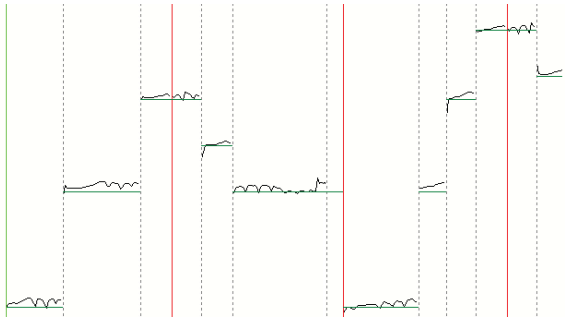

Figure 4. Example showing the rising pitch tendency, first made clear to the player by the program.

6. REFERENCES

[1] Agin, Gerald J. Intonia, http://intonia.com/index.shtml, 2008-2009.

[2] CantOvation *Sing&See*, http://www.singandsee.com/, 2008.

[3] ChaumetSoftware *Canta*, http://www.singintune.org/.

[4] Removed for review

[5] Flanagan, J. L. and Golden, R. M. *Phase Vocoder*. Bell System Technical Journal, November 1966.

[6] Kootsookos, Peter J. *A Review of the Frequency Estimation and Tracking Problems.* 1991.

[7] MakeMusic, Inc. *SmartMusic*, http://www.smartmusic.com/, 2008.

[8] McMahon, D. R. A. and Barrett, R. F. *Generalization of the Method for the Estimation of the Frequencies of Tones in Noise from the Phasese of Discrete Fourier Transforms.* Signal Processing Vol.12, 1987.

[9] Pygraphics *Interactive Pyware Assessment System*, http://www.pyware.com/ipas/, 2008.

[10] removed for review

[11] removed for review

[12] Robine, Matthias and Percival, Graham and LaGrange, M. "Analysis of Saxophone Performance for Computer-Assisted Tutoring", *Proceedings of the International Computer Music Conference (ICMC07)*, Copenhagen, Denmark, 2007.

[13] Schoonderwaldt, E. and Askenfelt, A. and Hansen, K. F. "Design and implementation of automatic evaluation of recorder performance in IMUTUS", *Proceedings of the International Computer Music Conference (ICMC05)*, Barcelona, Spain, 2005.

[14] StarPlayit *StarPlay*, http://www.starplaymusic.com/index.php, 2000.

[15] VoiceVista *VoiceVista* http://www.vocevista.com/, 2007.