INTUNE: A System to Support an Instrumentalist's Visualization of Intonation

Kyung Ae Lim and Christopher Raphael

School of Informatics and Computing, Indiana University

901 E. 10th St. Bloomington, IN 47408, USA

kalim@alumni.iu.edu, craphael@indiana.edu

One of our most beloved music teachers emphasized the importance of "facing the music," by which he meant listening to recordings of our playing. As with the first hearing of one's voice on recording, many were both surprised and suspicious of this external perspective. While sometimes revealing more than we were ready to hear, the long-term effect of this exercise helps 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 would admire hearing from another player.

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 (among other things) commit 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.

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 Allen and Helga Winold, professors emeriti of music in the Jacobs School of Music at Indiana University, and is freely available for download at http://www.music.informatics.indiana.edu/programs/InTune.

Past Work

Two recent examples, (Robine et al. 2007), (Schoonderwaldt et al. 2005), are addressing computer-assisted music instruction on intonation in the computer music community. Multimedia and e-learning conferences also have music training and education works presented: (Ng 2008), (Ng et al. 2008), (Percival et al. 2007), (Nakano et al. 2007). (Robine et al. 2007) and (Schoonderwaldt et al. 2005) address several other performance aspects, including dynamics, rhythm, articulation, etc. (Robine et al. 2007) and (Percival et al. 2007) share some of the basic kinds of visualization as does our work, though the effort is restricted to technical exercises such as playing of long tones or scales, rather dramatically restricting its reach. (Robine et al. 2007) also does not relate the results to a musical score, thus shifting the burden of interpretation to the musician.

Although we target on any instruments, these works target on the limited range of instruments: (Ng 2008), (Ng et al. 2008) designed for strings, and (CantOvation 2008), (ChaumetSoftware), (Nakano et al. 2007), (VoiceVista 2007) for vocal training. (Hoppe et al. 2006) presents a review on four different software tools for vocal training with real-time feedback. 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. (Schoonderwaldt et al. 2005) 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.

The basic kinds of visual music display we use are found in these examples as well. (CantOvation 2008) and (VoiceVista 2007) use the spectrogram, (Nakano et al. 2007), (ChaumetSoftware), (Percival et al. 2007) and a commercial audio editor (Celemony) use pitch trace representation. (MakeMusic 2008) and (StarPlayIt 2000) annotate 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 of automated score alignment, which allows us to relate the music data directly to our score representation. While (MakeMusic 2008) and (StarPlayIt 2000) 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 (Schoonderwaldt et al. 2005), (MakeMusic 2008) and (StarPlayIt 2000) 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.

Implementation

InTune is written in C++ for the Windows platform. A score-following system is the backbone of the program, and simplifies the problem of precise pitch estimation. We designed and implemented the user-centered program by working with 33 musicians (including user study subjects), from second grade children to faculty members at the music school.

Score-following

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 the 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 (Raphael 1999). 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

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

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 (Raphael 2002).

Pitch Estimation

Pitch Estimation has a long and active history as outlined in (Kootsookos 1991). Among the many approaches are those based on statistical models, in which it is possible to develop an "optimal" estimator. Our approach is not based on any explicit model and is quite simple, in spirit, but performs well enough to serve our application.

Our score following approach tells us when the various notes of the musical score occur, thus giving the approximate pitch for each frame of audio. That is, if the score match designates a frame to belong to a score event having MIDI pitch m, then the frequency 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 (Kootsookos 1991).

The frequency is defined here as the instantaneous rate of change of phase (divided by 2π); we estimate the frequency for a particular frame by approximating this derivative. This is a time honored and intuitive approach dating back to (McMahon and Barret 1987). If $Y_n(k)$ is the windowed finite Fourier transform of y_n at "bin" 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 estimates 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 hth 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.

Interface Design

We focused on designing *InTune* so that musicians from a wide range of backgrounds, instruments and performing levels, could easily use the program for their daily practice. The main goals of its interface are: 1) to present intonation feedback in the context of a musical score, 2) to display, rather than judge, the data, and 3) to invite users to apply their musical preferences into the system. To accomplish these three goals, we created a system that can get its score from a MIDI file, that displays three different views linked with one another alongside the audio, and that allows the user to adjust tuning and display settings.

On bringing up the program, the musician begins by choosing a piece to work on, at which point standard music notation is displayed. 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 displayed to the musician in a collection of three linked views, as shown in Figure 1.

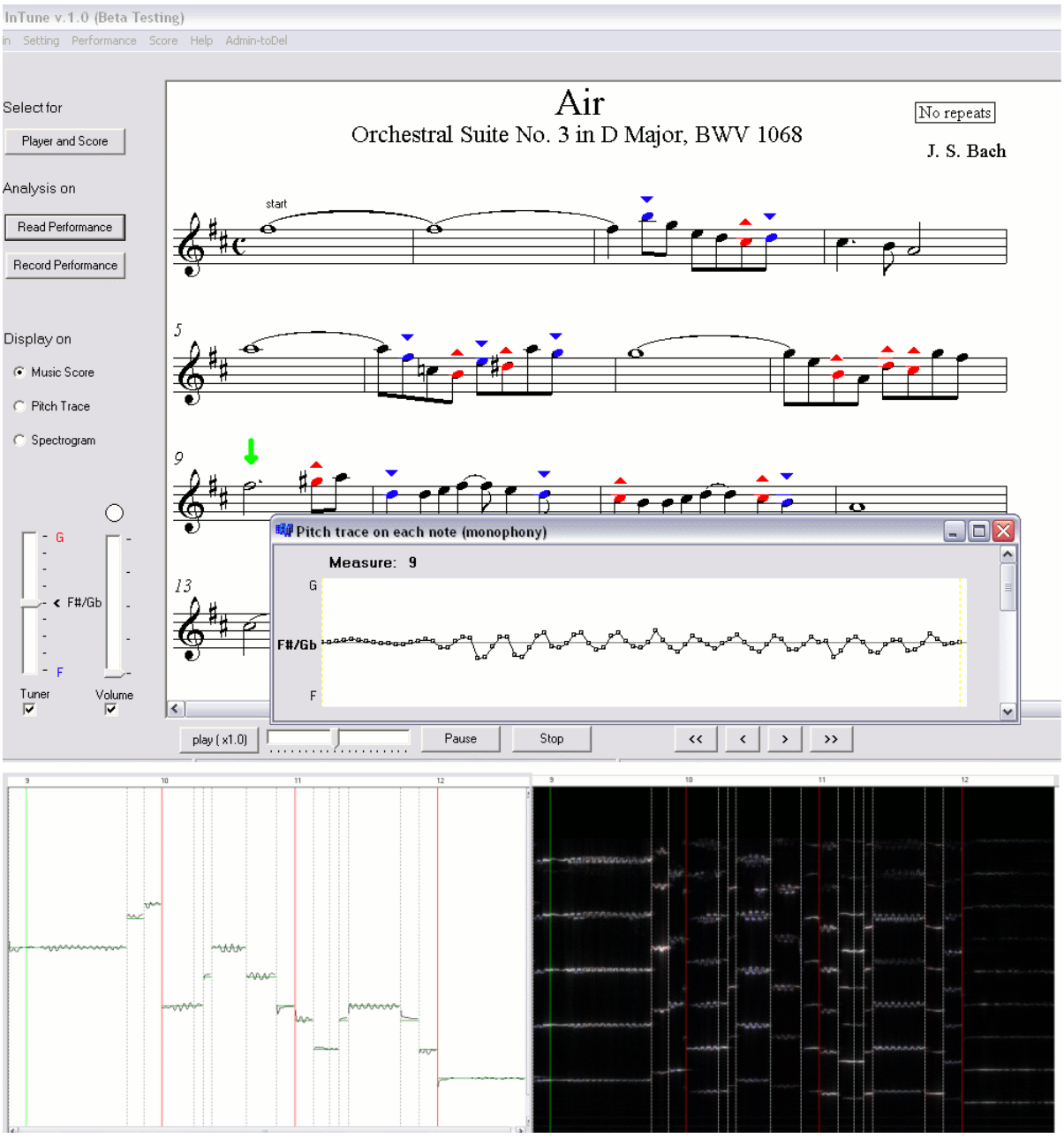
InTune v.1.0 (Beta Testing)
Setting
Performance
Score
Help
Admin-toDel
Select for
Player and Score
Analysis on
Read Performance
Record Performance
Display on
Music Score
Pitch Trace
Spectrogram
Tuner
Volume
G
F#/Gb
F
Air
Orchestral Suite No. 3 in D Major, BWV 1068
No repeats
J. S. Bach
start
Pitch trace on each note (monophony)
Measure: 9
play (x1.0)
Pause
Stop

Figure 1: *InTune*'s three displays of a performance. The score image (top) colors the heads of "suspicious" notes (two symbols ▲(sharp) and ▼(flat) are manually added to distinguish colors in case this paper is printed black and white.). The pitch trace (bottom left) shows precise pitch evolving over time on a log scale. The spectrogram (bottom right) is the traditional time-frequency display, showing frequency content evolving over time. The vertical lines show the measure or beat boundaries (red), the note boundaries (white), and the current position (green, ↓ above the current note). Users can navigate freely between these three images by clicking radio buttons.

User Settings

While *InTune* includes 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, as shown in Figure 2. Here the user will choose a single track from the MIDI file to form the score. In this process the program computes pitch spellings using the ideas of (Teodoru and Raphael, 2007).

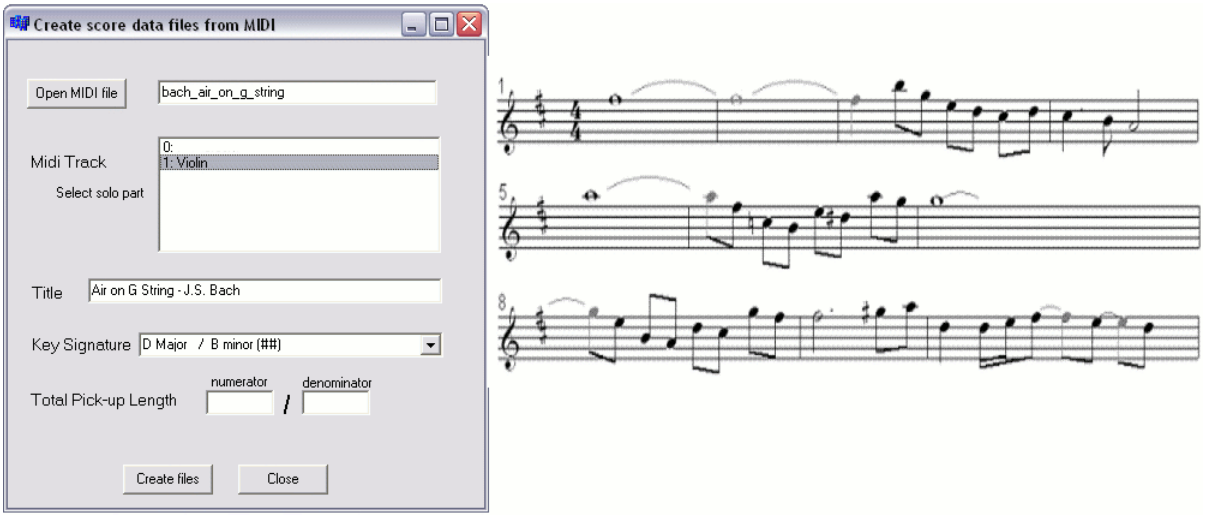

Figure 2: *InTune* generated score image. Window for importing MIDI score (left) and

example of *InTune's* automatic notated score (right). While our notation is rather spare (c.f. Figure 1) it is completely serviceable.

The system uses 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, as shown in Figure 3. Of course there is no single "correct" view of tuning --- in some situations tuning based on simple integer ratios may be preferable. 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 make their own accuracy judgments for their preferred notion of tuning based on this reference point without necessarily "buying in" to equal temperament. In fact, the importance of displaying, rather than judging, the tuning result was a basic tenet of ours, due to the lack of any single agreed-upon yardstick. *InTune* provides a setting for users to adjust the tolerance beyond which their notes are flagged as suspicious, as well as various display settings (Figure 3).

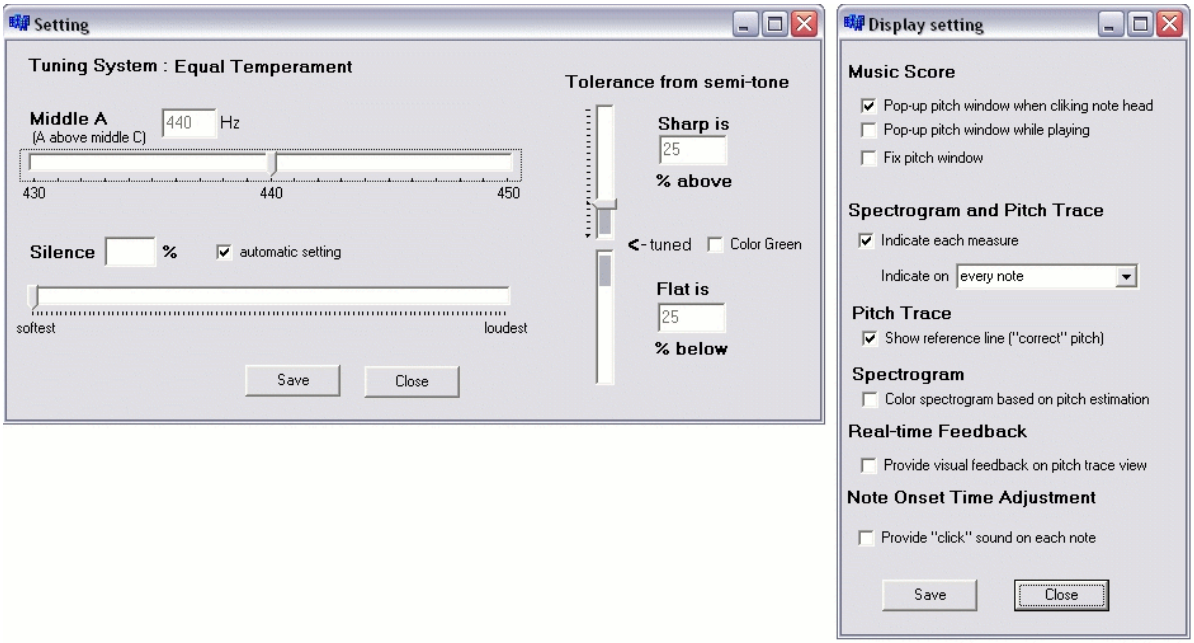

Figure 3: Setting. *InTune* provides both tuning (left) and display setting (right).

The Three Views

The *score view* (top of Figure 1) 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 (center right of Figure 1) 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.

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 (Flanagan and Golden 1966) 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* (bottom left 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 (right window of Figure 3), with vertical lines, computed from 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.

User Study

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

The students first chose and performed an excerpt from 10 ready-made pieces in *InTune*, while the program recorded their performance. Afterward, we played back their audio, without showing any feedback from *InTune*, while the subjects noted any intonation problems that they heard. Next, they were directed to use the program to review their performance, involving both visualizing and hearing their audio data. The students responded to a questionnaire assessing their beliefs about the usefulness of *InTune,* and their interest in incorporating the program into their practice. Overall, the students we quite positive about the program, with most saying they would incorporate *InTune* into their practice if it were available. Figure 4 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?

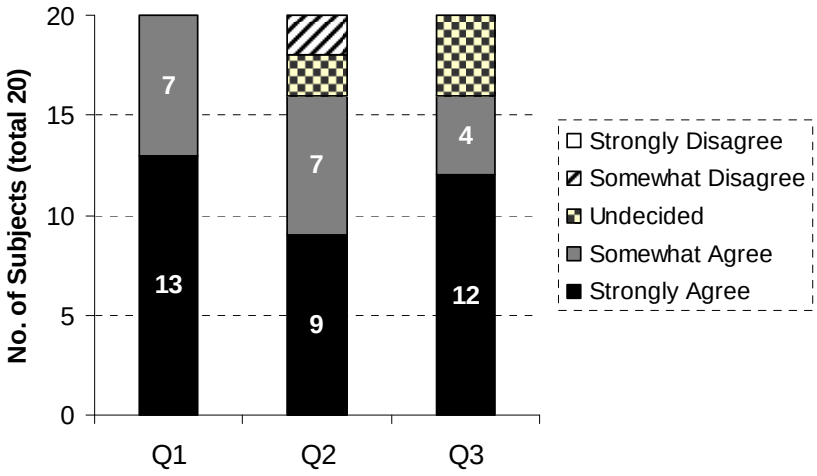

Figure 4. User Study. Answers to questions Q1, Q2, and Q3.

We were most pleased with the musicians' professed willingness to use the program in actual practice, and hope that this is followed with action.

The following set of questions compares the three views as means of conveying the program's intonation feedback. The subjects rated each view according to the perceived transparency, information content, and interest provided by each one. Figure 5 shows the responses from the users.

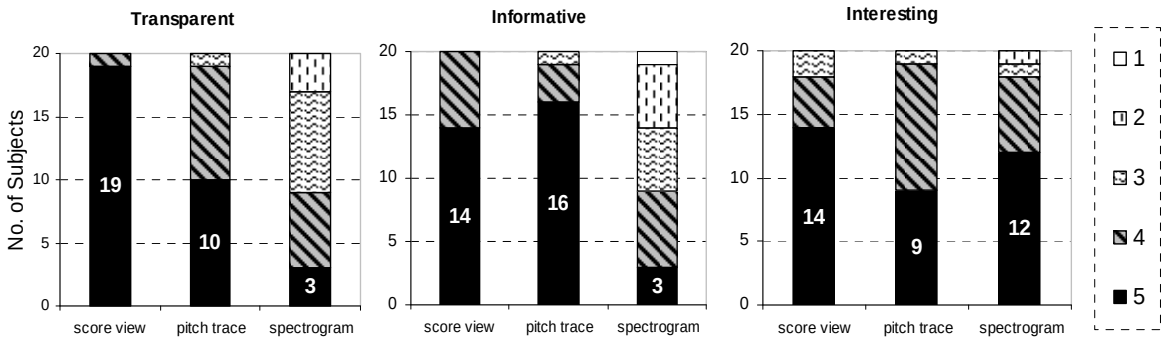

Figure 5: User ratings on a scale of 5 to 1 (highest to lowest) for each of the program's views.

Most students strongly agreed that the score view and the pitch trace view were transparent and informative, but the spectrogram view was scored significantly lower. While not reflected in our data, we found a number of musicians, especially brass players, to be particularly interested in the spectrogram view's ability to support concrete assertions about the seemingly intangible world of timbre.

Several themes emerged through the written and spoken comments accompanying 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 (shown in Figure 6), by allowing one to clearly see the nature of pitch excursions. Visualization of vibrato was also of particular interest to the music faculty with which we developed this project; throughout this collaboration we revisited, though never resolved, the issue of where within the vibrato's pitch range the perceived pitch lies.

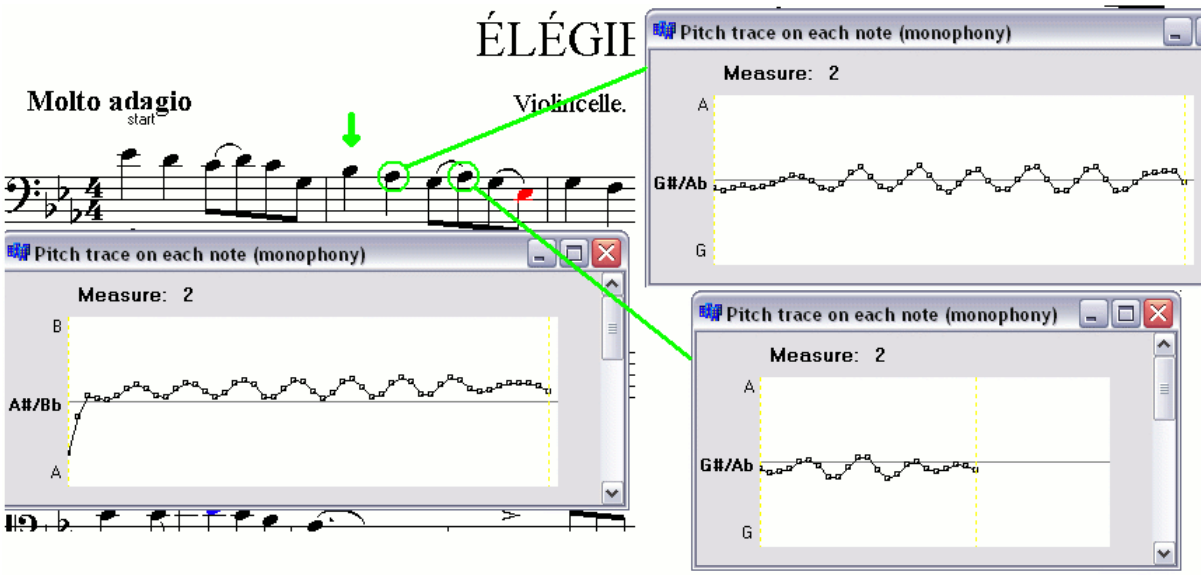

Figure 6: Example displaying vibrato on three notes from the same measure.

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 not to correct the users, it seems inevitable that some may interpret the program's output this way. One musician suggested the usefulness of simultaneously visualizing dynamics and intonation for

our future consideration, due to the common (and undesirable) connections between them.

Several subjects thought *InTune* would best serve beginning musicians and their teachers. IU's String Academy program teaches students ranging from age 5 to 18, usually with close involvement of the parents in early years. We found the program was useful in acquainting these young musicians (and some of the parents) with this important concept. The idea of coloring notes particularly interested some of the children, who received the program's feedback in a positive way.

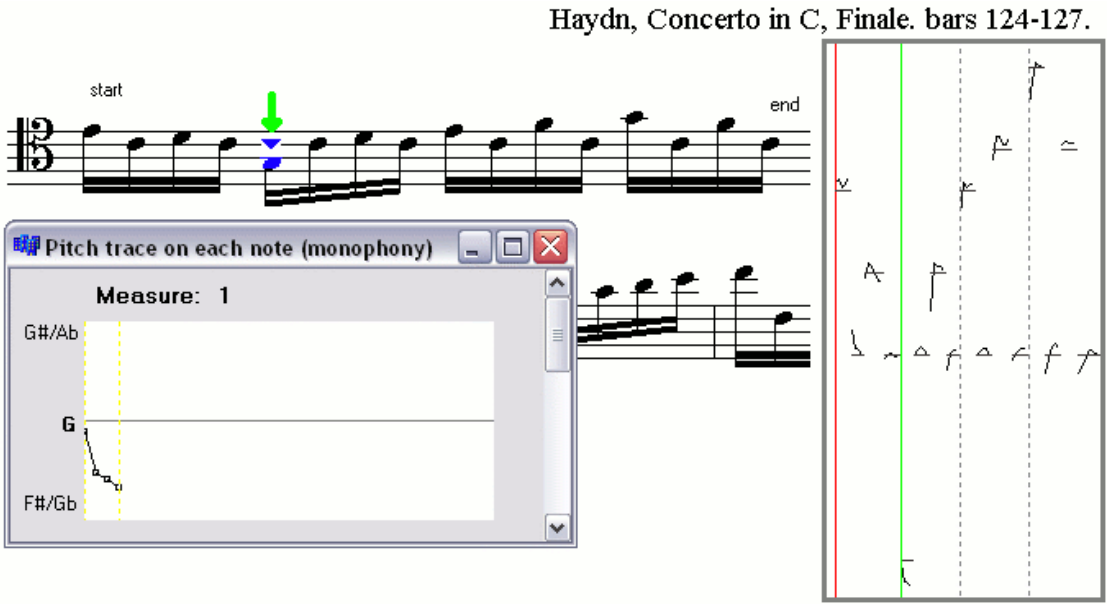

Figure 7: Example showing pitch estimation for fast notes (Pitch trace on the right).

Figure 7 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.

The most gratifying aspects of our user study came when our program enabled players' *hearing* of intonation issues they were previously unaware of. For example,

one horn player observed that he was consistently sharp throughout an entire register, as shown in Figure 8. In this example the use of score notation makes this tendency rather obvious, though the player may not have been aware of this beforehand.

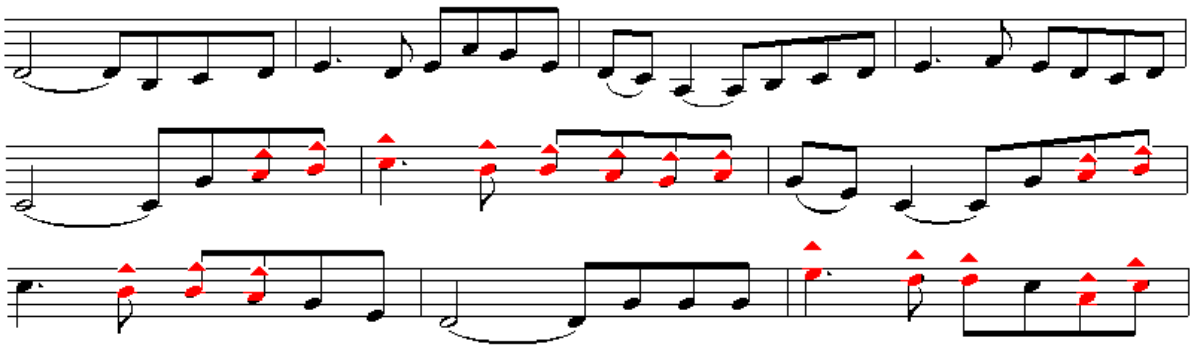

Figure 8: Example showing that a group of notes (above middle G) are sharp (red).

One especially interesting example of this variety occurred with a graduate flute major whose pitch data are shown in Figure 9 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. The audio for this and all other examples can be heard at `http://www.music.informatics.indiana.edu/papers/InTune`.

Figure 9: Example showing the rising pitch tendency, first made clear to the player by the program.

Acknowledgement

The authors thank Professors emeriti Allen and Helga Winold and all the students from the Jacobs School of Music at Indiana University, who contributed to the *InTune* project. This work was supported by NSF grant _IIS-0739563.

References

Agin, G. J., 2008-2009. "Intonia." `http://intonia.com/index.shtml`.

CantOvation, 2008. "Sing&See." `http://www.singandsee.com/`.

ChaumetSoftware, "Canta." `http://www.singintune.org/`.

Celemony, "Melodyne Studio." `http://www.celemony.com/`.

Dannenberg, R. B., and Raphael, C. 2006. "Music score alignment and computer accompaniment." *Commun. ACM* 49, 8, 38 - 43.

Flanagan, J. L., and Golden, R. M. 1966. "Phase vocoder." *Bell System Technical*

Journal (Nov.), 1493 - 1509.

Hoppe, D., Sadakata, M., and Desain P. "Development of Real-Time Visual Feedback Assistance in Singing Training: A Review, *Journal of Computer Assisted Learning*, Vol.22, pp.308 - 316, 2006.

Kootsookos, P. J., 1991. "A review of the frequency estimation and tracking problems." CRASys Technical Report, Systems Engineering Department, Australian National University

MakeMusic, I., 2008. "SmartMusic." `http://www.smartmusic.com`.

McMahon, D. R. A., and Barrett, R. F. 1987. "Generalization of the method for the estimation of the frequencies of tones in noise from the phases of discrete fourier transforms." *Signal Processing* 12(4), 371 - 383.

Nakano, T., Goto, M., and Hiraga, Y. "MiruSinger: A Singing Skill Visualization Interface Using Real-Time Feedback and Music CD Recordings as Referential Data", *Proceedings of the IEEE International Symposium on Multimedia (ISM 2007) Workshops*, pp.75 - 76, 2007.

Ng, K., Interactive Feedbacks with Visualisation and Sonification for Technology-Enhanced Learning for Music Performance, *in Proceedings of the 26th ACM International Conference on Design of Communication, SIGDOC 2008*, Lisboa, Portugal, 22-24 September 2008.

Ng, K., Nesi, P., and McKenzie, N. Technology-Enhanced Learning for Music Theory and Performance, *in Proceedings of IADIS International conference e-Learning 2008*, Amsterdam, The Netherlands, 22-25 July 2008

Percival, G., Wang, Y., Tzanetakis, G. Effective use of multimedia for computer-assisted musical instrument tutoring, *in Proceedings of the international workshop on Educational multimedia and multimedia education*, September 28-28, 2007, Augsburg, Bavaria, Germany

Pygraphics, 2008. "Interactive pyware assessment system."

http://www.pyware.com/ipas/.

Raphael, C. 1999. "Automatic segmentation of acoustic musical signals using hidden markov models." *IEEE Trans. on PAMI* 21, 4, 360 - 370.

Raphael, C. 2002. "A bayesian network for real-time musical accompaniment. In Advances in Neural Information Processing Systems." *NIPS* 14, MIT Press, T. Dietterich, S. Becker, and Z. Ghahramani, Eds.

Robine, M., Percival, G., and LaGrange, M. 2007. "Analysis of saxophone performance for computer-assisted tutoring." *In Proceedings of the International Computer Music Conference (ICMC07)*, vol. 2, 381 - 384.

Schoonderwaldt, E., Askenfelt, A., and Hansen, K. F. 2005. "Design and implementation of automatic evaluation of recorder performance in imutus." *In Proceedings of the International Computer Music Conference (ICMC05)*, 97 - 103.

StarPlayIt, 2000. "Starplay." http://www.starplaymusic.com/index.php.

Teodoru, G., and Raphael, C., 2007. "Pitch Spelling with Conditionally Independent Voices." *in Proceedings of ISMIR*, Vienna, 2007.

VOICEVISTA, 2007. "Voicevista." http://www.voicevista.com.