# OFFLINE SCORE ALIGNMENT FOR REALISTIC MUSIC PRACTICE

**Yucong Jiang**[1] **Fiona Ryan**[12] **David Cartledge**[2] **Christopher Raphael**[1]
[1] School of Informatics, Computing, and Engineering [2] Jacobs School of Music
Indiana University Bloomington, USA
yujiang,fkryan@iu.edu docartle,craphael@indiana.edu

## ABSTRACT

In a common music practice scenario a player works with a musical score, but may jump arbitrarily from one passage to another in order to drill on difficult technical challenges or pursue some other agenda requiring non-linear movement through the score. In this work we treat the associated score alignment problem in which we seek to align a known symbolic score to audio of the musician's practice session, identifying all "do-overs" and jumps. The result of this effort facilitates a quantitative view of a practice session, allowing feedback on coverage, tempo, tuning, rhythm, and other aspects of practice. If computationally feasible we would prefer a globally optimal dynamic programming search strategy; however, we find such schemes only barely computationally feasible in the cases we investigate. Therefore, we develop a computationally efficient off-line algorithm suitable for practical application. We present examples analyzing unsupervised and unscripted practice sessions on clarinet, piano and viola, providing numerical evaluation of our score-alignment results on hand-labeled ground-truth audio data, as well as more subjective and easy-to-interpret visualizations of the results.

## 1. INTRODUCTION

### 1.1 Problem Description

Score alignment finds a correspondence between a symbolic representation of a musical score and an associated audio performance, identifying the positions of all note onsets. The subject was introduced through the early musical accompaniment systems of Dannenberg and Vercoe [1, 2], while notable contributions include [3–13]. Cuvillier [3] provides a thorough review on score alignment. This paper deals with a variation of the traditional score alignment problem: instead of aligning a performance, we align the audio of a music practice session of a given score, where the player is allowed to skip from one score location to another in an arbitrary fashion. Such a score-alignment problem is also called score-alignment-with-skips, dropping the constraint of linear movement in the score while playing.

Traditional score alignment is typically partitioned into two varieties: on-line and off-line. On-line recognition is appropriate for applications in which the audio source must be understood in real-time, such as musical accompaniment systems, automatic page-turners, or approaches that coordinate the display of supertitles for opera. Off-line recognition is appropriate for score alignment applications without a real-time component, such as note-level editing of an audio performance, quantitative analysis of musical performance from a stylistic viewpoint, or the automatic generation of large sample libraries from recordings.

In this effort, we treat the *off*-line recognition of a music practice session. In particular, we target typical instrumental practice, which usually involves a large degree of repetition with particular attention directed toward challenging passages, as well as frequently inaccurate playing. The score-alignment-with-skips problem can enable a number of meaningful applications, as follows.

### 1.2 Applications

Off-line score-alignment-with-skips allows the development of useful tools that provide a high-level view of a practice session. Our experiments present a fledgling version of such a tool, facilitating efficient navigation through a practice session while coordinating the audio and visual display of the score. We believe the understanding provided by such a tool is far superior to that gained by simply listening to a practice recording, and is particularly important to musicians working to develop effective practice techniques. Interacting with such a tool implicitly answers a variety of useful questions, such as how much time was spent on a particular passage, or what was the typical length of a repeated fragment. The underlying analysis also enables additional feedback about tuning, tempo and rhythm, providing a deeper level of pedagogical feedback. For example, most wind players will have particular notes that are consistently flat or sharp, while such *global* tuning characteristics could easily be computed from the results of our proposed score alignment. Another example would be identifying a common problem of the student learner — unconsciously reducing the tempo when technical difficulties are encountered. Perhaps one could even develop measures of improvement over the course of a sequence of practice sessions.

Tools that facilitate navigating a practice session efficiently, perhaps including useful summaries of the session, offer particularly engaging possibilities for the music teacher as well. Many teachers experience the interval between lessons as something of a "black box," where divining the difference between a student's self-perceptions and reality can be a challenge. Currently, the closest we can get to practice intervention is to directly observe practice, or to have students submit recordings of their practice, both of which are linear, and in real time. If a teacher were able to view a high-level representation of practice over the course of a week, he or she could intervene and correct the students at a new and productive level of granularity, with reference to a concrete analysis of practice over time.

The score-alignment-with-skips variant has an on-line version as well, untreated here, though discussed in Nakamura et al. [6]. Such technology would be appropriate for a system that interacts with a musician during practice. For instance, after having identified the section that is currently being rehearsed, a system may provide an accompaniment that follows and supports the player. Fertile possibilities also exist for musical tutoring systems. For instance, at any time during a practice session we may add a metronome whose rate and phase initially synchronize with the live player, proceeding either deterministically or in an adaptive manner. We may also periodically suggest interventions over the course of the session, such as slow practice, or directing the subject's attention toward unmet challenges.

### 1.3 Related Work

One version of the score-alignment-with-skips problem is treated by Müller and Appelt [13], where the authors seek to compare different versions of a piece of music, perhaps with different choices of repeats, though with a preference for matching long sections of the two audio recordings, unlike what would be encountered with instrumental practice.

More recently, Nakamura [6] treats a version more oriented to our vision of practice analysis. This method performs online analysis by computing the filtered distribution without approximation through the usual "forward" iteration. With this approach, as with globally optimal dynamic programming computation of the most likely path, the computation is $O(NS)$, where $N$ is the length of the data and $S$ is the number of notes in the score. Nakamura observes that their algorithm is feasible in real time; however, from the computational complexity one can see that this depends on the particular score chosen. We imagine practice scenarios where the "score" might be a concatenation of all the scores in a player's library, thus nearly ruling out globally optimal approaches with no approximation. Furthermore, we expect that a more fine-grained approach will increase the number of states that must be devoted to each score note. For these reasons we pursue approaches that relax the guarantee of global optimality in exchange for both computational efficiency and extensibility to more complex graph topologies.

In our experiments we don't see a way to make direct comparisons with these approaches as Nakamura's work is on the filtering problem, thus not appropriate the off-line score alignment problem, while Müller considers a version of the problem that is far more constrained, thus not workable for kind of unconstrained practice considered in our experiments.

In what follows we explicitly describe our score alignment methodology. In addition, we present results on about two hours' worth of audio data on clarinet, viola and piano (polyphonic), collected from various members of the Jacobs School of Music at Indiana University, both in numerical fashion and through our audio-visual "practice browser."

## 2. MODELING

In this section, we first describe a hidden Markov model for the score alignment problem, which serves as the foundation of our approach to the score-alignment-with-skips problem. Then, we explain the motivations behind our approach. Lastly, we explain using a pitch tree and beam search to accommodate the computation burden.

### 2.1 HMM for Score Alignment

Score alignment has been cast as a hidden Markov model (HMM) problem by several authors [3, 5, 6, 9, 12]. Here, we use the framework in [9].

As the HMM views time as discrete, we model time as a sequence of "frames" of about 30 ms. in length. We denote the hidden Markov chain as $X = X_1, \ldots, X_N$ where $N$ is the number of frames in the audio excerpt, and $X_n$ is the hidden state associated with the $n$th frame, taking values in a state graph. A simple construction of the state graph models the $k$th note as a chain of states, $s_{k,1}, \ldots, s_{k,M}$, where $M$ is the maximum length of the $k$th note, in frames. Figure 1 shows a topology where each state, $s_{k,m}$, either connects to $s_{k,m+1}$, the next state of the same note, or to $s_{k+1,1}$, the first state of the next note.

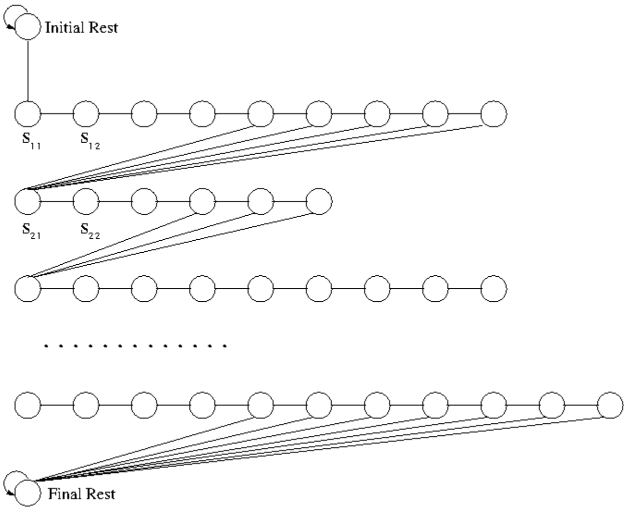

Figure 1. A possible left-to-right graph topology for score alignment.

#### 2.1.1 Transition Probability

Suppose we let $Q(x, x')$ be the transition probability matrix for $X$, $Q(x, x') = P(X_{n+1} = x' | X_n = x)$, where we assume time homogeneity — these probabilities don't depend on $n$. Suppose $L_k$ is the random length of the $k$th note and that we have some desired distribution for this length, $P(L_k = l)$, which is indicated from the music score. Since visiting state $s_{k,m}$ means that the realization of the $k$th note is at least $m$ frames long, we have

$$\begin{aligned} Q(s_{k,m}, s_{k,m+1}) &= P(L_k \geq m+1 | L_k \geq m) \\ &= \frac{\sum_{l=m+1}^{M} P(L_k = l)}{\sum_{l=m}^{M} P(L_k = l)}, \end{aligned}$$

and $Q(s_{k,m}, s_{k+1,1}) = 1 - Q(s_{k,m}, s_{k,m+1})$. As Figure 1 shows, we append a start state and an end state, both with self-loops, to the beginning and end of the graph as the simple state-space model for the Markov chain, $X$. For longer notes, we also allow their states to have self-loops. This model can also be regarded as a hidden semi-Markov model [15] if we view all the "micro states" of a note as one super state.

### 2.1.2 *Data Model*

We only briefly describe the data model here because it is not the most important part of our proposed idea — one can easily replace our data model with a new one while using our framework. For each frame, $n$, we observe a short burst of audio data, $y_n$. We model the data likelihood in terms of the normalized magnitude spectrum of $y_n$,

$$e_n(\omega) = \frac{|z_n(\omega)|}{\sum_{\omega'} |z_n(\omega')|} \tag{1}$$

for $\omega = 1, \ldots, \Omega$ where $z_n$ is the windowed finite Fourier transform of $y_n$, and $\Omega$ is the number of *bins* in the frequency domain. We then model the data likelihood as

$$P(y_n | X_n = x) = \prod_{\omega=1}^{\Omega} q_x(\omega)^{e_n(\omega)} \tag{2}$$

where $q_x$ is the probability distribution over frequency we associate with state $x$ (the template of state $x$). Refer to Raphael [16] for detailed description of this data model.

### 2.1.3 *Inference*

With our HMM in place, it is possible to compute a number of quantities relevant to inference about the audio performance [17]. For instance, one can compute the forward model, giving the evolving state of knowledge on score position, $P(X_n = x_n | y_{1:n})$ where $y_{1:n} = y_1, \ldots, y_n$; $P(X_n = x_n | y_{1:N})$, the state distributions given the entire data $y_{1:N}$; or the most likely sequence of states, $\hat{x} = \arg\max_{x_{1:N}} P(X = x_{1:N} | y_{1:N})$. All of these computations use dynamic programming or dynamic-programming-like algorithms.

## 2.2 Score-alignment-with-skips and Motivations

Models like the one depicted in Figure 1 assume the player will play the score as written, from the beginning of the excerpt to the end — the usual assumption of score alignment, which is appropriate for many applications, but not reasonable for the "free practice" case at hand. In score-alignment-with-skips, we expect that the player will play *sections* of the score, perhaps repeating them numerous times, before moving on to other sections. When a particular section is practiced, we assume the player will play the score notes in order (just as in traditional score alignment), according to the notated rhythm. Therefore, we do not wish to completely abandon the basic model of Figure 1.

In our approach, we want to allow occasional skips in which our Markov model, $X$, jumps from one score position to another. In practice, the overwhelming majority of these skips are "do overs" — cases where the player repeats a group of notes that are *most recently* played because he or she is unsatisfied with the sound, perhaps repeating numerous times. Therefore, small backward skips are the most likely possibilities. However, we cannot constrain the model to *only* allow such local skips, because occasionally the player will shift to a completely new section of the score, or restart from the beginning. If our model is to be genuinely useful, it must allow for such non-local skips as well. We extend the model of Figure 1 to allow for score skips by adding a "hub" state that communicates in both directions with each of the note models, as proposed in Nakamura et al. [6]; any note can "jump" to or from this hub state.

As discussed earlier, we believe globally optimal "full-fledged" dynamic programming approaches, either for online or offline versions of this problem don't leave sufficient headroom for exploring the space of possible graph topologies or expanding the search space to model *collections* of scores the musician is studying. Thus we focus on *beam search* methods — algorithms that retain a fixed-size list of the currently-best hypotheses at each frame. Typically the beam is several hundred hypotheses in our experiments. Considerations for beam search models are different from those for full-fledged dynamic programming, since a hypothesis must look attractive at *every* stage of the computation in order to avoid being pruned. In addition, we use a "pitch tree" to further help with computation, as will be discussed in what follows.

## 2.3 Pitch Tree and Beam Search

Figure 2 introduces our *global skips* model which allows the player to jump from any score location to any other score location at any time. In the bottom of this figure, the linear sequence of states is a compact description of the original model of Figure 1. In this linear graph each note has been compressed into a single state for simplicity's sake. Each of these states can either remain in the current state, move forward in the score, or "escape" to the "wait" state at the root of the tree in the top of the figure. The escape probability is chosen to be small enough so that our model is disinclined to recognize one- or two-note (super short) excerpts, but still capable of identifying them. In essence, we use the tree structure to "sort out" the player's score position in a computationally efficient manner when a jump is made.

The root of the tree is a state with a self loop, modeling the typical pause that occurs as one stops playing and resumes again at a new location. Therefore, the data model for this state is the silence model. The rest of the tree is illustrated in the case of the short "toy" score represented by the pitch sequence $a, b, c, a, b, a, b$ given in Figure 2. The

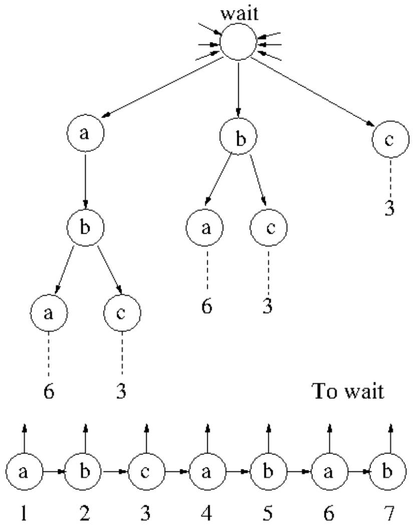

Figure 2. The global skips model in which a "pitch tree" allows the efficient sorting out of the score location after a jump.

possible pitches in the score are $a$, $b$ and $c$, so these define the first level of the tree. $c$ appears only once in the score so unambiguously defines the position as score note 3. On the other hand, $b$ has two possible successors in the score ($bc$ and $ba$), thus two children in the tree. Because each of the children unambiguously identifies a score position (6 or 3), they terminate their branches. The construction continues in this manner until the score position is uniquely identified.

In general, we define a tree construction as follows. Suppose for now we treat a monophonic instrument, although we generalize to polyphonic cases too in our experiments. Let $M$ be the distinct possible pitches that occur in the score, expressed as the letter names $\{a, b, c\}$ in Figure 2. We denote the finite-length sequences of such pitches by $M^*$. We let $\tilde{M} \subset M^*$ be the collection of all subsequences of pitches, without regard for rhythm, that appear *multiple* times in the score. $\tilde{M}$ indexes the non-terminal nodes of our tree: $\{t_c : c \in \tilde{M}\}$. We let $\tilde{M}_0$ denote the pitch subsequences that appear *only once* in the score whose prefixes are in $\tilde{M}$ — these are the shortest sequences that uniquely determine the score position. $\tilde{M}_0$ indexes the terminal nodes of our tree: $\{t_c : c \in \tilde{M}_0\}$. If $c \in \tilde{M}$ and $c' \in \tilde{M} \cup \tilde{M}_0$, with $c' = c \circ m$ for $m \in M$ (sequence $c'$ is $c$ concatenated by pitch $m$), then $t_c \to t_{c'}$ in the tree graph (non-terminal node $t_c$ has a successor node, $t_{c'}$). The terminal nodes in the tree are really proxies for the score notes in the linear graph. That is, if $t_c \to t_{c'}$ with $c' \in \tilde{M}_0$, then $t_c$ really connects to the score note uniquely identified by the string $c'$. This association is made explicit by the dotted lines in Figure 2. For example, in the left branch (out of the three branches), the second level $b$ really connects to score positions 6 and 3.

The purpose of the tree is efficient computation while maintaining accuracy. After a jump is made, our data model usually argues strongly for a small number of world of possible pitches. Especially in the context of beam search where only the best several hundred hypotheses are kept at each frame, it seems wasteful to maintain an individual hypothesis for each separate place a pitch occurs in the score, so our tree efficiently maintains only a single hypothesis for all such score positions. When the next note is played, we drop down a level in the tree, reaching a state associated with all of the score positions where the pair of pitches occur in order. We continue this process until the score position is unambiguously determined, at which point it "joins" our original score model (as in Figure 1).

In reality, we would not move down the tree deterministically, but, rather, would consider a range of possible tree positions supported by the data model, as is always the case when finding the most likely state sequence of an HMM using dynamic programming and a beam search.

Although beam search does not guarantee a global optimal result, in practice the correct hypotheses usually survives pruning because the data model is strong. The pitch tree also helps with avoiding unwanted prunings, since when a jump is performed we represent the possible score positions compactly, refining our representation as more information becomes available. Even if all correct hypotheses are pruned out at some unfortunate frame, the search can "recover" at any time by jumping to the wait state and then to the correct score position. This behavior is verified in the following experiments.

| Composer | Piece | Meas. | Min. |
|---|---|---|---|
| Mozart | Clarinet Concerto, Mvmt 1 | 1-154 | 9 |
| Mozart | Clarinet Concerto, Mvmt 2 | 1-59 | 6 |
| Mozart | Clarinet Concerto, Mvmt 3 | 1-112 | 13 |
| Brahms | Sonata for Viola in Eb, Mvmt 1 | 1-97 | 14 |
| Hoffmeister | Concerto for Viola, Mvmt 1 | 1-150 | 11 |
| Bartok | Concerto for Viola | 1-200 | 15 |
| Mozart | Piano Sonata K. 330, Mvmt 1 | 1-88 | 14 |
| Beethoven | Piano Sonata op. 110, Mvmt 1 | 1-70 | 15 |
| Debussy | *La Fille aux Cheveux de Lin* | 1-39 | 17 |

Table 1. List of repertoire used in the experiments.

## 3. EXPERIMENTS

### 3.1 Data

We collected practice audio from a number of students and faculty mostly in the Jacobs School of Music at Indiana University. These consisted of three undergraduate clarinet majors, three undergraduate viola majors, one faculty pianist, and one student pianist who was not a music major. The data together account for a little less than two hours of practice audio. Part of our goal in collecting these data was to understand the range of variation encountered in real-life practice sessions. In particular, we want to know if the score-alignment-with-skips model of a practice session is tenable — can musicians naturally confine their practice to the score as our model assumes? We also want to know the accuracy of our approach in tracking the players' score trajectories.

We instructed our subjects to practice an agreed-upon piece of music and only this piece for the duration of their recorded practice session, generally 10 to 20 minutes in a single "take." Table 1 lists the pieces we tested, along with the associated measure ranges and practice session lengths. Aside from requesting that the pianists practice with both hands together throughout, we tried to give subjects a minimum of direction and did not supervise their practice or try to otherwise constrain their practice beyond the initial instructions. We observed a number of departures from the basic playing-with-skips assumption, including, for example, deliberately distorted rhythms, testing of reeds, playing significantly slower than the generally-accepted tempo, one-hand piano practice, and brief forays into interval tuning practice only loosely related to the score.

Pieces with verbatim repetition of passages pose problems for both recognition and evaluation: if a player jumps to a passage that appears multiple times in a piece, how can we say for sure which version is being played? However, this distinction doesn't seem especially important from the standpoint of evaluation. For instance, consider the 3rd movement (*Rondo*) of the Mozart Clarinet Concerto, where the refrain repeats six times with almost no variation. It doesn't seem reasonable to penalize our algorithm for failing to ascertain which repetition is being practiced, nor does it seem feasible to make this determination while creating ground truth. For simplicity sake, we chose excerpts from the pieces where there was no direct phrase-level repetition of musical material (as in Table 1).

To create ground truth for an audio example, we first split it by hand into contiguous sections, each section containing music played without skips. We then performed score alignment on each of these sections individually. The results include the section information (the starting note and the starting frame of a section), and onsets of all played notes. The results were then meticulously corrected by hand to be as precise as we could get them.

### 3.2 Evaluation Method and Results

We propose a simple way to evaluate the score-alignment-with-skips problem that is easy to implement and useful for comparing with other approaches. In both ground turth and recognized results, all frames between two consecutive notes are associated with the former note. In other words, any given frame is associated with the note whose onset is the most recent. For every frame, we calculate the "musical distance" between the *recognized* note of this frame and the *ground truth* note of this frame. For example, assuming the time signature is $4/4$ and the score has a quarter note at every beat, the musical distance between the second quarter note of measure one and the first quarter note of measure two is $3/4$. Such distances tell us the quality of our recognition — how far away the recognized note is from the actual played note. We evaluate our algorithm by counting the number (proportion) of frames that have different levels of musical distances (errors). Our goal is to have more frames in the "0" category (accurate), and fewer frames in the "$>$ 1" category (larger error). Figure 3-5 show this kind of evaluation result for nine pieces in our experiments.

Figure 3 gives these frame-by-frame position errors for the three sessions from the Mozart Clarinet Concerto. This histogram, as well as those of the other two instruments presented later, bins the errors into several categories generated with split points given as 0, one eighth note, one quarter note, one half note, and one whole note. We use the same binning procedure regardless of the tempo of a piece or its time signature, so, for instance, a whole note error in 6/8 time corresponds to 1+1/3 measures. The most important categories are the two extreme ones: "0", where the score position has been identified as accurately as possible, and "$>$1 (whole note)", where the recognizer is essentially lost. The clarinet is perhaps the easiest instrument to recognize due to its comparative pitch stability. Their results were the best that we measured, with the recognizer lost (error $>$ 1) no more than 3% of the time in all cases. It was interesting to note that the deliberately distorted rhythms observed occasionally in the first movement did not create any problems for recognition.

Figure 4 shows analogous results for the viola data, in which we observed the recognizer being lost from 6% to

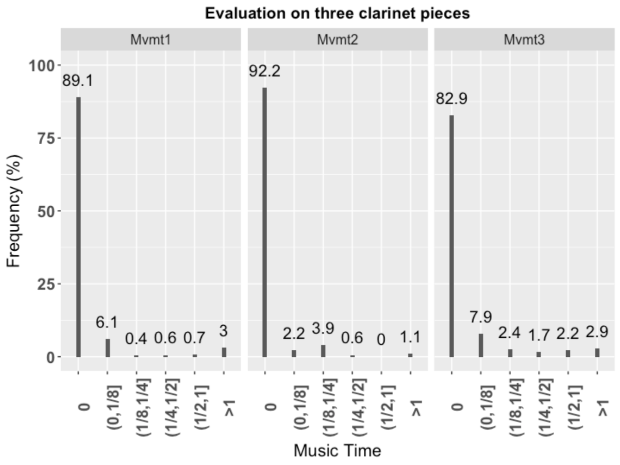

Figure 3. Histograms of frame-by-frame errors for the three practice sessions taken from the three movements of the Mozart Clarinet Concerto, as described in Table 1.

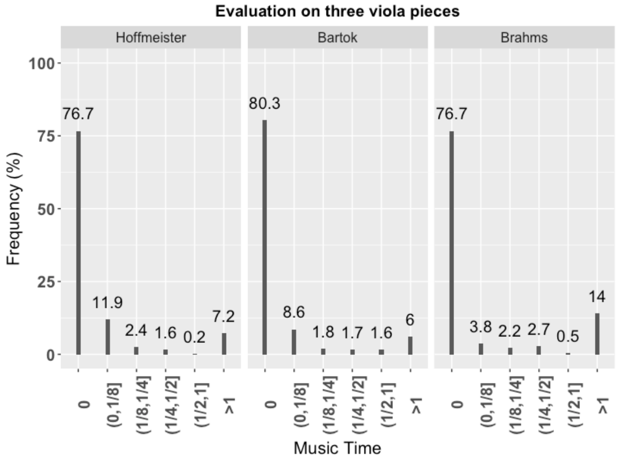

Figure 4. Histograms of frame-by-frame errors for the viola data as described in Table 1.

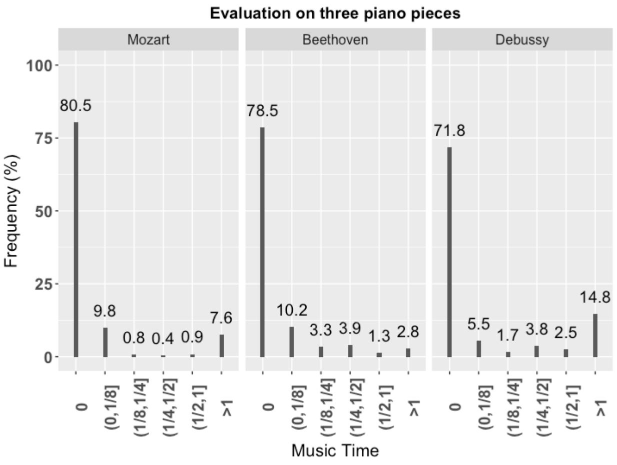

Figure 5. Histograms of frame-by-frame errors for the piano data as described in Table 1.

14% of the time. These results are not quite as good as with the clarinet data, for which we conjecture several reasons. First of all, the viola is simply harder to recognize, since (viola jokes aside) the instrument does not commit itself as clearly to pitch as the clarinet does. In addition, the viola plays double stops (quite a few in some sections of the Bartok and Hoffmeister), while our pitch (data) models tend not to discriminate as well between such chords. Finally, the practice session for the Brahms included many one- or two-note excerpts, and in a couple of cases seemed to be only *inspired* by the score, rather than directly from the score. However, as can be seen from the numerical results, the problems caused by all of these factors were only local.

Figure 5 describes the results for the three piano examples in Table 1. The piano constitutes a significantly harder challenge than the two mostly monophonic instruments. We take a *homophonic* view of the piano, regarding the score as a sequence of chords without regard for voice. That is, whenever the score indicates that a note enters or exits, we create a new chord at the appropriate musical position. This allows the piano to be recognized in the same fashion as used for the other instruments. It should be noted that the "pitch tree" approach of Figure 2 may be less effective as there is far less repetition of chord sequences than of individual pitch sequences.

Generally speaking, the piano is much more challenging than monophonic instruments, since, as mentioned before, the data model discriminates less well between chords than single notes. In addition, the nature of the instrument produces much more overlap between notes, either through pedaling, which is not reflected in our scores, or through the fact that the addition of new notes has no damping effect on preceding notes. In contrast, the essential physics of string, wind, and brass instruments cause the new note to damp the previous note (except in the case of different strings). However, these challenges seem to manifest mostly themselves in terms of small onset inaccuracies, rather than causing the recognition to become lost any more than with the viola. Again, the "lost" percentages were in the 2% to 14% range. The highest error rate was from the slow Debussy piece, where far less is known about the timing of a performance than with fast music.

A less numerical, but perhaps more illuminating example can be seen at *http://music.informatics.indiana.edu/papers/smc19-skips/*, where the video highlights the player's current position in a musical score as the practice audio plays. Similar videos for all of the practice sessions can be found at the same web site. In addition to providing an easily digestible demonstration of the heart of this research, the videos also foreshadow the kinds of tools we envision developing for musicians to help review practice effectively.

The results presented here are "exploratory" (on a small dataset), so we obviously cannot claim broad coverage of the world of possible practice habits — such a data collection would be a large undertaking in and of itself. Still, we encountered a good deal of variation within the sampled population. We believe the results show that our essential practice assumptions are reasonable, in the sense that our subjects were, for the most part, able to follow our model of score-constrained practice without much difficulty, while cases that departed from our model created only local problems or no problems at all. We believe the accuracy of our score alignment is also promising. In short, the algorithm occasionally gets lost but always finds its way back to the correct score position. Furthermore, we believe the accuracy of note onset estimates on the individual identified sections is certainly good enough for many kinds of pedagogical feedback.

## 4. FUTURE WORK

The basic recognition ideas developed here can be embedded into thought-provoking and illuminating tools to help instrumentalists review their practice, along the lines of the demonstrated video. To achieve this goal we must both improve the basic recognition on which these "practice browsers" will rely, as well as identify forms of feedback of interest to students, and ways of expressing that feedback visually.

The current approach assumes one cannot tell the difference between identical passages in a piece of music, though this is only partly true. Typically we can make rather strong assumptions about the way in which the musician will visit the piece of music. In particular, nearly all jumps are local ones, and the overwhelming majority of jumps move backwards. Nakamura et al. [18] also analyzed the skipping property on three piano pieces. These assumptions do not fit naturally with our sorting trees of Figure 2 since the non-terminal nodes of the tree are associated with multiple score positions. Therefore, we can't assess their distance from the jump origin. In contrast, a simple model that allows only local backward jumps performs surprisingly well for the overwhelming majority of cases. However, when the local backward assumption is violated this model becomes completely lost, thus is too fragile. We anticipate that it is possible to find a modeling framework that can express the likelihood of various jumps, while also retaining the computational efficiency of our sorting tree. This is a project for future study.

Tuning is certainly an obvious candidate for visualization, perhaps by coloring score notes according to the tuning error. Coverage of a practice session could be similarly represented, using color to denote the number of times a particular passage has been repeated. Giving feedback on rhythm is more challenging, partly because there will always be some degree of error in the identification of note onsets, but also because good rhythm depends both on timing and stress (or lack of stress). We anticipate considerably challenge here, though clearly there is much fertile ground to explore.

## 5. REFERENCES

[1] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," in *ICMC*, vol. 84, 1984, pp. 193–198.

[2] B. Vercoe, "The synthetic performer in the context of live performance," in *Proceedings of International Computer Music Conference*, 1984, pp. 199–200.

[3] P. Cuvillier, "On temporal coherency of probabilistic models for audio-to-score alignment," Ph.D. dissertation, Université Pierre et Marie Curie-Paris VI, 2016.

[4] A. Cont, D. Schwarz, and N. Schnell, "Training ircam's score follower [audio to musical score alignment system]," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 3. IEEE, 2005, pp. iii–253.

[5] D. Schwarz, A. Cont, and N. Schnell, "From boulez to ballads: Training ircam's score follower," in *International Computer Music Conference (ICMC)*, 2005, pp. 1–1.

[6] T. Nakamura, E. Nakamura, and S. Sagayama, "Real-time audio-to-score alignment of music performances containing errors and arbitrary repeats and skips," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 2, pp. 329–339, 2016.

[7] M. Puckette, "Score following using the sung voice." in *ICMC*, 1995.

[8] L. Grubb and R. B. Dannenberg, "A stochastic method of tracking a vocal performer." in *ICMC*, 1997.

[9] C. Raphael, "Automatic segmentation of acoustic musical signals using hidden markov models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 360–370, 1999.

[10] R. J. Turetsky and D. P. Ellis, "Ground-truth transcriptions of real music from force-aligned midi syntheses," 2003.

[11] N. Orio and F. Déchelle, "Score following using spectral analysis and hidden markov models," in *ICMC: International Computer Music Conference*, 2000, pp. 1–1.

[12] P. Cano, A. Loscos, and J. Bonada, "Score-performance matching using hmms." in *ICMC*, 1999.

[13] M. Müller and D. Appelt, "Path-constrained partial music synchronization," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 65–68.

[14] C. Raphael, "Music plus one and machine learning." in *ICML*, 2010, pp. 21–28.

[15] K. P. Murphy, "Hidden semi-markov models (hsmms)," *unpublished notes*, vol. 2, 2002.

[16] C. Raphael, "A hybrid graphical model for aligning polyphonic audio with musical scores." in *ISMIR*. Citeseer, 2004, pp. 387–394.

[17] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[18] E. Nakamura, T. Nakamura, Y. Saito, N. Ono, and S. Sagayama, "Outer-product hidden markov model and polyphonic midi score following," *Journal of New Music Research*, vol. 43, no. 2, pp. 183–201, 2014.