ALIGNING SEMI-IMPROVISED MUSIC AUDIO WITH ITS LEAD SHEET

Zhiyao Duan and Bryan Pardo
Northwestern University
Department of Electrical Engineering & Computer Science
zhiyaoduan00@gmail.com, pardo@northwestern.edu

ABSTRACT

Existing audio-score alignment methods assume that the audio performance is faithful to a fully-notated MIDI score. For semi-improvised music (e.g. jazz), this assumption is strongly violated. In this paper, we address the problem of aligning semi-improvised music audio with a lead sheet. Our approach does not require prior training on performances of the lead sheet to be aligned. We start by analyzing the problem and propose to represent the lead sheet as a MIDI file together with a structural information file. Then we propose a dynamic-programming-based system to align the chromagram representations of the audio performance and the MIDI score. Techniques are proposed to address the chromagram scaling, key transposition and structural change (e.g. a performer unexpectedly repeats a section) problems. We test our system on 3 jazz lead sheets. For each sheet we align a set of solo piano performances and a set of full-band commercial recordings with different instrumentation and styles. Results show that our system achieves promising results on some highly improvised music.

1. INTRODUCTION

In this work we investigate the problem of aligning an audio recording of semi-improvised music to a lead sheet. This problem belongs to a more general research problem called *score alignment*, i.e. finding the time mapping between a musical performance and its score. The fulfillment of this task would be very useful for a number of applications like synchronizing multiple sources (video, audio, score, etc.) of music in a digital library and automatically accompanying a musical performance.

In the last two decades, many methods have been proposed for score alignment in different problem settings: MIDI to MIDI, audio to MIDI, monophonic or polyphonic audio performances, online or offline, etc. [4]. However, most methods assume faithful performances to a fully-notated score, with at most a tempo change and key transposition.

We call modern jazz *semi-improvised*, because many significant elements of the music are improvised but deeper-level structural aspects remain relatively fixed. The score for semi-improvised music is called a *lead sheet*. A lead sheet specifies only essential elements like a basic melody, harmony, lyric and a basic musical form. A performer typically improvises all the notes in a solo, changes in tempo, accompaniment figuration and even some structural elements of a piece (e.g. repeating a chorus). The nature of semi-improvised music makes the alignment to a lead sheet very challenging. Even for an educated musician it is sometimes difficult to align an improvisation to the lead sheet when the improvisation has high degree of freedom.

For aligning such performances, a few methods have been proposed. Dannenberg and Mont-Reynaud [5] aligned a jazz solo performance with the chord progression on the score. Pardo and Birmingham [10] aligned a polyphonic semi-improvised MIDI performance with its lead sheet. They also proposed a method [11] to follow a performance with possible structural variations, i.e., deviating from the expected path written on the score by skipping or repeating a section. The above-mentioned methods have loosened the faithful performance assumption, however, they are either limited to deal with MIDI performances [10,11], or can only follow a solo performance under a 12-bars blues form [5]. Arzt and Widmer [1] also proposed an alignment system to handle structural variations, but only for non-improvised (classical) music. To our knowledge, there is no existing methods that align a semi-improvised (polyphonic) audio performance under an arbitrary form with its lead sheet.

This problem is in some ways similar to Cover Song Identification (CSI), i.e. identifying different performances (usually by different artists) of the same song [7]. However, variations of these performances are generally much less than those in what we called semi-improvised music such as modern jazz. In addition, the alignment methods used in CSI only serve as an intermediate step for similarity calculation, and no precise time mappings are required.

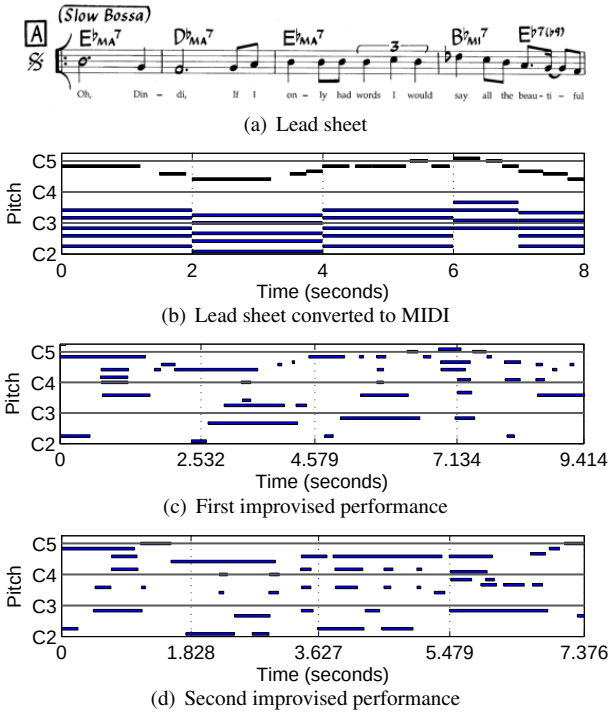
In this paper, we attempt to address the semi-improvised

music audio-score alignment problem, without prior training on example performances of the lead sheet to be aligned. We first analyze the problem's unique properties in Section 2, then propose an alignment system regarding these properties in Section 3. In Section 4 we describe experiments to test the system on real performances of solo piano and jazz combo. Section 5 concludes this paper.

2. PROBLEM ANALYSIS

2.1 Basic Properties

The problem considered in this paper is aligning an audio recording of a semi-improvised music performance to its lead sheet. A lead sheet usually only specifies a basic melody, harmony, lyric and a basic musical form (structure). Take Figure 1(a) as an example. The melody is indicated by note heads. Harmony is indicated by chord symbols above the staff. Lyrics are indicated as text below the staff. The text "A" with a square indicates the start of Section A, and the repeat sign besides it suggests that this section is often repeated in a performance. We can translate this lead sheet into a MIDI file by setting a tempo (e.g. 120BPM), rendering harmony as block chords with root notes in the C2-C3 octave and discarding the lyric and music structure information. The piano-roll representation of this MIDI is shown in Figure 1(b). We mark measures with vertical dash lines.



(a) Lead sheet

(b) Lead sheet converted to MIDI

(c) First improvised performance

(d) Second improvised performance

Figure 1. Four measures of the lead sheet for *Dindi* by Antionio Carlos Jobim, and its two semi-improvised piano performances.

In semi-improvised performances, the performer views the lead sheet as a reference and continuously creates new musical elements that are not on the score. Figures 1(c) and 1(d) show the piano-rolls of two semi-improvised piano performances by two different pianists of the lead sheet, with measure times marked by vertical dash lines. We can see that the two performances have different tempi from the lead sheet. Also, harmony is rendered in free rhythmic patterns. We also notice that the melody contour of the lead sheet remains in the first performance, while is significantly altered in the second performance.

2.2 Representing Harmonic Content

Harmonic content is the most similar feature that an semi-improvised performance and its lead sheet shares. We need to find a representation of harmonic content, robust to variations among different performances, on which to do the alignment. The chromagram is a good representation which has been used in many audio-score alignment methods [4]. In these methods, chroma features are usually calculated for every short time frame (e.g. 46 ms), so that the alignment can be precise at the millisecond level. However, this choice is not suitable in our problem, as we can see in Figure 1 that performed notes can be significantly different from the notes written on the lead sheet at any one 46 ms frame. In fact, chord labels on the lead sheet are more like sets of high-likelihood notes to be played over given time periods (e.g. two beats of D minor 7), and aggregating performed notes across larger time spans (e.g. two beats) makes for a clearer correspondence to the score. Therefore we choose to calculate chroma features in this scale.

2.3 Utilizing Structural Information

Structural information on the lead sheet is also important for an alignment system. Performers often modify the basic musical form, but not arbitrarily. For example, the basic form of *Dindi* is "Intro-[A-A-B-C]", where the bracket represents a repeat sign. Performers may skip the Intro section at the beginning but play it at the end. They may change the repeat bracket by including the Intro section or excluding the A sections. Basically, they view musical sections as toy bricks, selecting and shuffling them during a performance. However, it is not common to make other structural changes such as making a jump at the middle of a section.

However, structural information on the lead sheet is not encoded in the MIDI representation shown in Figure 1(b). Therefore, we encode it in an additional file, as shown in Table 1. Basically, this file stores two kinds of information: 1) musical section definitions and boundaries; 2) possible jumps that an semi-improvised performance might make.

Sections	from	to	Jumps	from	to
Intro	1	16		48	1
A	17	24		48	17
A	25	32		48	33
B	33	40			
C (A)	41	48			

Table 1. Structural information extracted from the lead sheet for *Dindi*. Section C is very similar to Section A.

3. PROPOSED SYSTEM

Based on the above analysis, we design our system as shown in Figure 2. We represent both the audio and MIDI with a chromagram where chroma vectors are extracted at the 2-beats scale, then use a modified string alignment algorithm that can handle structural changes to align the chromagrams.

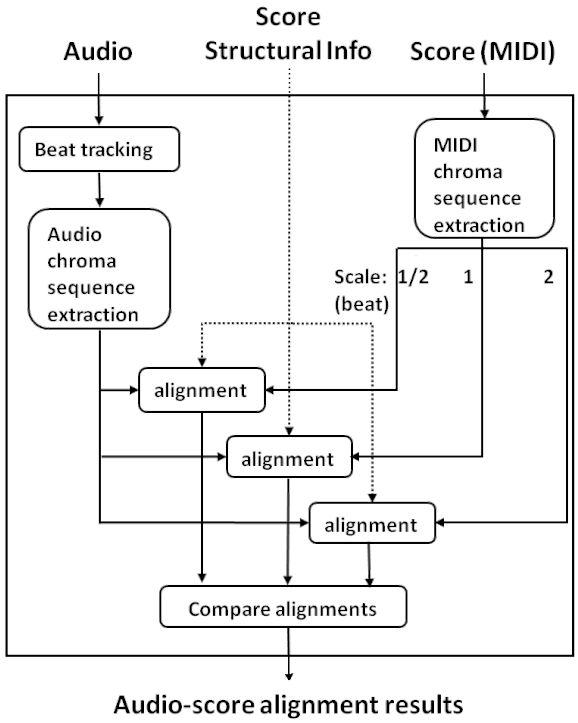


Figure 2. Overview of the proposed system.

3.1 Audio Beat Tracking

In order to extract chroma features from audio at the 2-beat scale, we need audio beat times of the performance. We use the original implementation of the beat tracking algorithm proposed by Ellis [6]. While this is a high-quality beat tracker, the estimated tempo often has halfing/doubling errors, as described in [6]. In addition, when the performance has an unstable tempo, the algorithm may find extra beats or miss some beats.

3.2 Audio Chroma Feature Extraction

We first chop the audio signal into 46 ms long time frames with a 23 ms hop size and calculate a chroma vector for each frame. The frame-level chroma vector is 12-d, and is calculated by "folding" the local maxima of the hamming-windowed Short Time Fourier Transform (STFT) spectrum to the 12-pitch classes. This tends to suppress the non-harmonic part of the spectrum.

As discussed in Section 2.2, the ideal analysis unit is not the 46 ms frame, but something on the order of 2 musical beats. We therefore average the chroma vectors of the frames into segments of length l and a hop size h, where these values are measured in beats. The resulting chromagram is a sequence of the segment-level chroma vectors. In our experiments, we set l and h to 2 beats and $\frac{1}{4}$ beats, respectively. A segment size of two beats worked well for the harmonic rhythm of the music analyzed, with the shortest duration chords typically being 2 beats. For the hop size h, theoretically a smaller h leads to a more precise alignment. However, the computational complexity increases quickly as h shrinks ($O(1/h^2)$). We investigate the influence of different parameters on the alignment result in Section 4.

3.3 MIDI Chroma Feature Extraction

As with the audio chromagram, we segment the MIDI representation of the lead sheet into segments of length l and hop size h, and calculate a chroma vector for each segment. We simply sum up the lengths of notes in each segment to their corresponding pitch-class bins. We generate 12 transposed MIDI chromagrams to cope with the possible key transposition of the audio performance.

3.4 Chromagram Scaling Problem

In Section 3.1, we note that the estimated tempo of the audio might be half or twice the true tempo. Therefore the audio and MIDI chromagrams might be on temporal different scales, which will strongly influence the alignment result.

To address this problem, we also segment the MIDI file and calculate the chromagram in three ways, with segment length and hop size of (l, h), $(2l, 2h)$ and $(\frac{1}{2}l, \frac{1}{2}h)$, respectively. Therefore, for each audio-MIDI pair, we have 1 audio chromagram and 36 MIDI chromagrams, corresponding to 3 scales and 12 key transpositions. It is noted that the idea of time scaling and key transposition has been used in other music information retrieval systems such as [3].

3.5 Aligning Chromagrams

Let $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_m)$ be the audio chromagram, $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n)$ be the score chromagram, where $\mathbf{a}_i$ is the

chroma vector of the i-th audio segment and $\mathbf{s}_j$ is the chroma vector of the j-th score segment. We describe a dynamic-programming algorithm to align them. Unlike standard string alignment algorithms, this algorithm utilizes structural information provided by the lead sheet (as shown in Table 1) to handle possible structural changes in the semi-improvised performance. To do so, we define a *parent-index set* $\mathcal{P}(j)$ for each score segment index j. Each element k of $\mathcal{P}(j)$ is a score segment index, from which a semi-improvised performance might transition to j. This transition can be a smooth progression i.e. $k = j - 1$, or a forward/backward jump. In the latter case, the pair (k, j) is a possible jump listed in the structural information file as Table 1.

Now we recursively define a $(m + 1) \times (n + 1)$ alignment cost matrix $\mathbf{C}$, where the value $\mathbf{C}(i, j)$ is the lowest cost of the alignment between the initial sub-chromagrams $(\mathbf{a}_1, \cdots, \mathbf{a}_i)$ and $(\mathbf{s}_1, \cdots, \mathbf{s}_j)$. For all $i = 1, \cdots, m$ and $j = 1, \cdots, n$, $\mathbf{C}(i, j)$ are calculated as follows:

$$\mathbf{C}(0,0) = 0, \mathbf{C}(i,0) = i \cdot c_1, \mathbf{C}(0,j) = 0 \tag{1}$$

$$\mathbf{C}(i,j) = \min \begin{cases} \mathbf{C}(i, j-1) + c_1 \\ \mathbf{C}(i-1, j) + c_2 \\ \min_{k \in \mathcal{P}(j)} \mathbf{C}(i-1, k) + d(\mathbf{a}_i, \mathbf{s}_j) \end{cases} \tag{2}$$

where c_1 and c_2 are constants specifying the costs of skipping one segment of audio and score in the alignment, respectively. $d(\mathbf{a}_i, \mathbf{s}_j)$ specifies the cost of mismatching the i-th audio segment with the j-th score segment.

Note that Eq. (1) is not symmetric, i.e. $\mathbf{C}(i, 0)$ is set to $i \cdot c1$, but $\mathbf{C}(0, j)$ is set to 0 instead of $j \cdot c2$. This means that we penalize skipping audio segments at the beginning but do not penalize skipping score segments, i.e. we assume that the performance can start anywhere but must be on the lead sheet. Although sometimes performers play several measures that are unrelated to the lead sheet at the beginning, this is short compared to the whole performance and we ignore this case. In addition, the third line in Eq.(2) is calculated from $\mathbf{C}(i-1, k)$ for all possible parents k of the j-th score segment, while in an standard string alignment algorithm it is only calculated from $\mathbf{C}(i-1, j-1)$. This allows the performance to play to the j-th score segment in all possible ways, either progress smoothly from the previous segment $j - 1$ or jumping from other segments.

The mismatch cost function $d(\mathbf{a}_i, \mathbf{s}_j)$ is defined as:

$$d(\mathbf{a}_i, \mathbf{s}_j) = \arccos\left(\frac{\mathbf{a}_i^T \mathbf{s}_j}{\|\mathbf{a}_i\| \|\mathbf{s}_j\|}\right) \tag{3}$$

We use cosine angle distance instead of Euclidean distance to make it loudness insensitive. This is because the loudness of the audio may vary from the loudness calculated from the score differently in different performances. Since angle distance between an arbitrary audio-score chroma vector pair is around 1, we set $c_1 = c_2 = 1$ to match the three penalties.

While calculating $\mathbf{C}$, we fill another $m \times n$ matrix $\mathbf{P}$, where $\mathbf{P}(i, j)$ stores the index pair (i', j') from which $\mathbf{C}(i, j)$ is calculated in Eq. (2). When the calculation of $\mathbf{C}$ is finished, the *final alignment cost* is calculated as $\min_j \mathbf{C}(m, j)$. Let $j_1 = \arg\min_j \mathbf{C}(m, j)$. We than trace back from the index pair (m, j_1) through $\mathbf{P}$ to some index pair $(1, j_2)$. The sequence of index pairs $(1, j_2), \cdots, (m, j_1)$ give the alignment between $\mathbf{A}$ and $\mathbf{B}$. Note that the last pair is (m, j_1) instead of (m, n). This allows the audio performance to end at any position of the score.

If we view each score segment as a state, each audio segment as an observation, then the proposed algorithm is essentially equivalent to the forward-backward algorithm for a Hidden Markov Model (HMM) [12]. The transition matrix T has a positive value t_1 on the diagonal, corresponding to the penalty of skipping an audio segment c_1. It also has a positive value t_2 on the superdiagonal (elements $(j-1, j)$) and elements (k, j) for all $k \in \mathcal{P}(j)$, corresponding to the penalty of skipping a score segment c_2 by smooth progressions and jumps, respectively. If $c_1 = c_2$, then $t_1 = t_2$. We also notice that this algorithm is equivalent to the one proposed by Fremerey et al. [8], which also handles jumps and repeats in synchronizing a score with a performance.

Finally, for each audio-MIDI pair, we do the alignment 36 times corresponding to the 36 MIDI chromagrams. The alignment that achieves the lowest final alignment cost is selected as the output of the system.

4. EXPERIMENT

4.1 Dataset

Our dataset consists of 36 semi-improvised performances of 3 jazz lead sheets: *Dindi* by Antonio Carlos Jobim, *Nicas's Dream* by Horace Silver and *Without A Song* by Vincent Youmans, selected from commonly used jazz fake books. For each song, the performances consist of two subsets. The first subset contains MIDI recordings performed by professional Chicago jazz pianists obtained from [9]. In [9], four pianists each gave three different performances scaled to three subjective levels of difficulty, ranging from a performance closely adhering to the given lead sheet to a more "free" interpretation. After recording, these pianists also annotated their own performances with beat, measure and structural branch point information, encoded as MIDI data. We include the two less difficult levels into our dataset (denoted as *easy* and *medium*), totalling 8 jazz piano performances for each song. We render these MIDI performances into audio recordings with the Logic Audio software using Grand Piano sound samples. We use the pianists' annotations to generate the ground-truth audio-score alignment.

The second subset contains 4 commercially released recordings for each lead sheet. Table 2 shows basic information for them. To generate the ground-truth audio-score alignment,

two musicians listened to these recordings, marked beat and measure time points and identified the score position (score measure number) of each measure of the audio. Audio measures that are unrelated to the lead sheet (e.g. an improvised cadenza) were labeled score measure number 0.

	ID	Performer(s)	Instruments
Dindi	1	Astrud Gilberto	female, violin, guitar
	2	Charlie Byrd	guitar, saxophone
	3	Ohta San	guitar
	4	Sadao Watanabe	string, saxophone
Nica's...	1	Art Farmer	trumpet, trombone, brass
	2	Benjamin Koppel Quintet	saxophone, piano, conga
	3	Cal Tjader	vibraphone, piano
	4	The Hot Club	violin, guitar
Without...	1	Diane Schuur	female, piano, bass
	2	Joe Henderson	saxophone, brass, piano
	3	Oscar Peterson	piano, brass
	4	Sonny Rollins	saxophone, brass, guitar

Table 2. Improvised performances played by jazz bands.

For each improvised performance, we use two experimental settings. In the first setting, we align the *whole* performance with the lead sheet. This is to observe our system's behavior on a larger time scale (usually several minutes). In the second setting, we randomly select 10 *excerpts* of the performance and align them with the lead sheet. The length of each excerpt ranges from 16 measures to 48 measure. This is to observe our system's behavior on a smaller scale (usually 30 seconds to 2 minutes) and would be representative of the task of selecting a portion of audio in a music player and asking to be shown the corresponding place on the lead sheet. The second setting is in general more challenging, as there is less context information.

4.2 Evaluation Measures

A commonly used measure for audio-score alignment is *Align Rate (AR)* as proposed in [2]. It is defined as the percentage of correctly aligned notes in the score, where "correct" means that the note onset is aligned to an audio time which deviates less than a short time (e.g. 250 ms) from the ground-truth audio time. In our problem, however, there is no bijective correspondence between score notes and audio notes, hence it is very hard to define the ground-truth audio time for each score note and AR is not suitable.

We formulate our problem as a classification problem, by assigning to each audio frame a score measure number. Given this, we simply use *Accuracy* as our measure. It is calculated as the proportion of audio frames which are correctly assigned score measure numbers as the ground-truth. We exclude those audio frames where the performance is unrelated to the score. This measure ranges from 0 to 1.

4.3 Results

Figure 3 shows overall results of aligning whole performances. Among the 36 performances, 11 have accuracies higher than 75%, 13 between 50% and 75%, while 6 lower than 10%. Their average is 54.8%. It is noted that a random guess alignment would get an accuracy as the reciprocal of the number of measures on the lead sheet, about 2%.

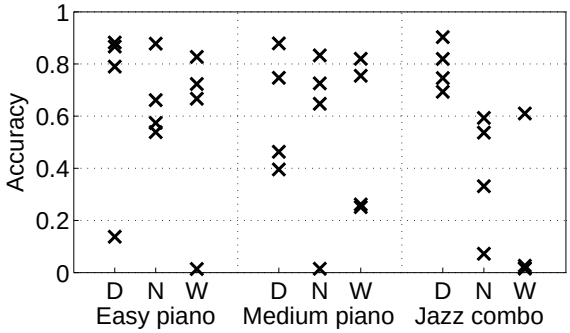


Figure 3. Alignment accuracies of all the 36 whole performances. 'D', 'N' and 'W' represents the lead sheet names *Dindi*, *Nica's Dream* and *Without A Song*, respectively.

We show three examples with different alignment accuracies in Figure 4. In the upper panel, the system's output alignment matches with the ground-truth perfectly except in two parts (51-58 seconds, 193 seconds - end). In both parts the performance is unrelated to the lead sheet. It is noted that the accuracy measures always underestimate the performance of the system, because the audio beat boundaries estimated by the beat tracking module are not perfectly aligned with the ground-truth beat boundaries, hence the assigned score measure numbers of the audio frames that are close to these boundaries are often off for ± 1 measures.

In the middle panel, the performance sometimes repeats from the Intro section and sometimes from Section A. Our system handles this uncertain structural change well. However, it incorrectly identifies the two B sections around 150 seconds (also the two B sections around 250 seconds) as only one B section with about half the tempo. Interestingly, it comes back to the right position after this error. In addition, after incorrectly identifying Section A (175-192 seconds) as C and B, the system identifies another A section (192-210 seconds) as Section C. Since Section A and C are almost the same on the lead sheet, this error is reasonable. Excluding this error causes accuracy to increase to 65.8%.

In the bottom panel, our system fails totally. Audio frames are constantly skipped after about 16 seconds. This example played by Diane Schuur, however, is very difficult. First, there are four parts (0-12, 91-97, 162-165 seconds and 179 seconds - end) that the performance is unrelated to the lead sheet. Second, the performance plays at half the tempo

in Section C (142-162 seconds). Third, the performance switches to a new key at 165 seconds till the end. The audio, MIDI and alignment results of these and other examples can be accessed at `http://www.cs.northwestern.edu/~zdu459/ismir2011/examples`.

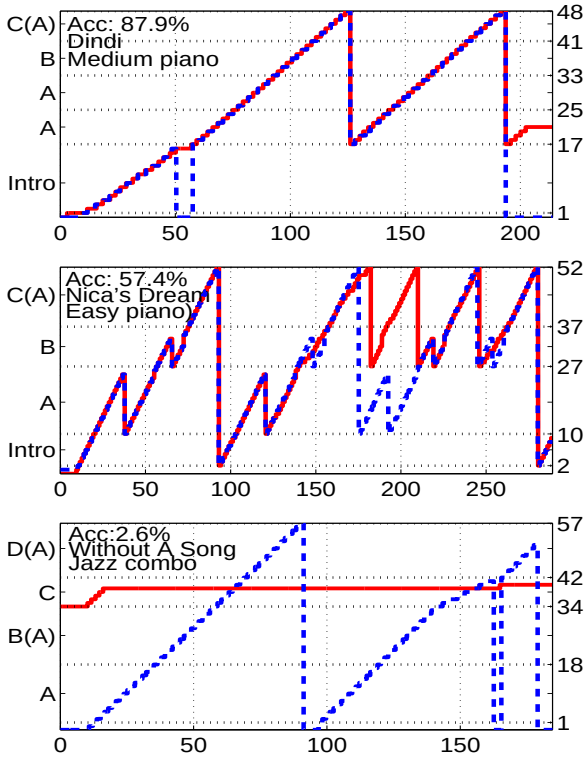


Figure 4. Three alignment examples. The horizontal axis is audio time in seconds. The left vertical axis shows section names of the lead sheet. The right vertical axis and the horizontal dash lines show the boundaries of the sections in measure numbers. Red solid lines show the system's alignments. Blue dash lines show the ground-truth alignments.

Figure 5 shows the average alignment accuracies over all 360 performance excerpts with different chroma length l and hop size h settings. Our choice of $l = 2, h = 1/4$ achieves an accuracy of 49.3%, which is one of the highest among all the parameter settings. This is in accordance to the analysis in Section 2.2. This result shows that with much less contextual information, our system still works well on some highly improvised audio excerpts.

5. CONCLUSION

In this paper, we attempted to align semi-improvised music audio with its lead sheet. We proposed a simple system to align chromagram representations of audio and score based on a modified string alignment algorithm, which utilizes structural information of the lead sheet. Experiments

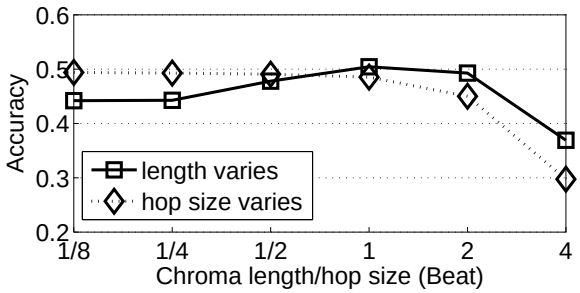


Figure 5. Average accuracies over all 360 excerpt performances, versus chroma length (fix hop size = 1/4) or hop size (fix chroma length = 2).

on 36 audio performances and their 360 excerpts of 3 lead sheets showed promising results. This work is supported by NSF grant IIS-0643752.

6. REFERENCES

[1] A. Arzt and G. Widmer, "Towards Effective 'Any-Time' Music Tracking," in *Proc. of the Starting AI Researchers Symposium (STAIRS)*, 2010.

[2] A. Cont, D. Schwarz, N. Schnell and C. Raphael, "Evaluation of real-time audio-to-score alignment," in *Proc. ISMIR*, 2007.

[3] R.B. Dannenberg, W.P. Birmingham, B. Pardo, N. Hu, C. Meek, 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. 3, 2007.

[4] R.B. Dannenberg, C. Raphael, "Music score alignment and computer accompaniment," *Commun. ACM*, vol. 49, no. 8, pp. 38–43, 2006.

[5] R.B. Dannenberg and B. Mont-Reynaud, "Following an improvisation in real time," in *Proc. ICMC*, 1987, pp. 241–248.

[6] D. Ellis, "Beat tracking by dynamic programming," *J. New Music Research, Special Issue on Beat and Tempo Extraction*, vol. 36 no. 1, pp. 51–60, 2007.

[7] D. Ellis and G. Poliner, "Identifying 'cover songs' with chroma features and dynamic programming beat tracking," in *Proc. ICASSP*, 2007.

[8] C. Fremerey, M. Müller, M. Clausen, "Handling repeats and jumps in score-performance synchronization," in *Proc. ISMIR*, 2010.

[9] J. Moshier and B. Pardo, "A database for the accommodation of structural and stylistic variability in improvised jazz piano performances," *ISMIR, Late-Breaking/Demo Session*, 2008.

[10] B. Pardo and W. Birmingham, "Following a musical performance from a partially specified score," in *Proc. IEEE Multimedia Technology and Applications Conference*, 2001.

[11] B. Pardo and W. Birmingham, "Modeling form for on-line following of musical performances," in *Proc. AAAI*, 2005.

[12] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition." in *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.