SCORE FOLLOWING FOR PIANO PERFORMANCES WITH SUSTAIN-PEDAL EFFECTS

Bochen Li **Zhiyao Duan**
Audio Information Research (AIR) Lab,
University of Rochester, Department of Electrical and Computer Engineering
bli23@ur.rochester.edu, zhiyao.duan@rochester.edu

ABSTRACT

One challenge in score following (i.e., mapping audio frames to score positions in real time) for piano performances is the mismatch between audio and score caused by the usage of the sustain pedal. When the pedal is pressed, notes played will continue to sound until the string vibration naturally ceases. This makes the notes longer than their notated lengths and overlap with later notes. In this paper, we propose an approach to address this problem. Given that the most competitive wrong score positions for each audio frame are the ones before the correct position due to the sustained sounds, we remove partials of sustained notes and only retain partials of "new notes" in the audio representation. This operation reduces sustain-pedal effects by weakening the match between the audio frame and previous wrong score positions, hence encourages the system to align to the correct score position. We implement this idea based on a state-of-the-art score following framework. Experiments on synthetic and real piano performances from the MAPS dataset show significant improvements on both alignment accuracy and robustness.

1. INTRODUCTION

1.1 Audio-Score Alignment

Audio-score alignment is the problem of aligning (synchronizing) a music audio performance with its score [8]. It can be addressed either offline or online. Offline algorithms may "look into the future" when aligning the current audio frame to the score. Online algorithms (also called *score following*), on the other hand, may only use the past and current audio data to align the current audio frame to the score. Provided with enough computational resources, online algorithms can be applied in real-time scenarios. As online algorithms utilize less input data than offline algorithms, they can support broader applications including those in offline scenarios. However, they are also more challenging to achieve the same alignment accuracy and robustness as offline algorithms do.

Audio-score alignment has many existing and potential applications. Offline algorithms have been used for audio indexing to synchronize multiple modalities (video, audio, score, etc.) of music to build a digital library [28]. Other applications include a piano pedagogical system [3] and an intelligent audio content editor [11]. Online algorithms further support online or even real-time applications, including automatic accompaniment of a soloist's performance [8], automatic coordination of audio-visual equipment [18], real-time score-informed source separation and remixing [11], and automatic page turning for musicians [1]. Potential applications of audio-score alignment include musicological comparison of different versions of musical performances, automatic lyrics display, and stage light/camera management.

1.2 Related Work

In this section, we briefly review existing approaches to audio-score alignment with an emphasis on score following for piano performances, which is the problem addressed in this paper.

Audio-score alignment has been an active research topic for two decades. Early researchers started with monophonic audio performances. Puckette [25], Grubb and Dannenberg [16], and Cano et al. [4] proposed systems to follow vocal performances. Orio and Dechelle [23] used a Hidden Markov Model (HMM)-based method to follow different monophonic instruments and voices. Raphael [26] applied a Bayesian network to follow and accompany a monophonic instrument soloist.

For polyphonic audio, a number of offline systems using Dynamic Time Warping (DTW) have been proposed for different kinds of instruments, including string and wind ensembles [24] and pop songs [17]. For online algorithms, Duan and Pardo [12] proposed a 2-dimensional state space model to follow an ensemble of string and wind instruments. All the abovementioned methods, however, have not been tested on piano performances.

There are a few systems that are capable of aligning piano performances. Joder and Schuller [20] proposed an HMM system with an adaptive-template-based observation model to follow piano performances. In [19], Joder et al. further improved the system by exploring different feature functions for the observation model and using a Conditional Random Field (CRF) as the alignment frame-

Attribution: Bochen Li, Zhiyao Duan. "Score Following for Piano Performances with Sustain-Pedal Effects", 16th International Society for Music Information Retrieval Conference, 2015.

work. Wang et al. [29] employed DTW to achieve alignment in three passes of the audio performance and used score-driven NMF to refine the audio and score representations in later passes. All the abovementioned systems have been systematically evaluated and shown with good performance on about 50 classical piano performances from the MAPS dataset [14], however, they are offline algorithms and require the entire audio piece to find the alignment. Dixo and Widmer [10] developed a toolkit to align different versions of music audio performances including piano based on an efficient DTW algorithm. However, this is again an offline algorithm, although an extension to online scenarios can be made through online DTW algorithms [9].

For online algorithms capable of following piano performances, Cont [5] proposed a hierarchical HMM approach with Nonnegative Matrix Factorization (NMF). However, this system was not quantitatively evaluated. Later, Cont [6] proposed another probabilistic inference framework with two coupled audio and tempo agents to follow general polyphonic performances. This algorithm has been systematically evaluated on 11 monophonic and lightly polyphonic pieces played by wind and string instruments, but just 1 polyphonic piano performance (a Fugue by J.S. Bach).

1.3 Our Contribution

In this paper, we are interested in following piano performances. Their specific properties, such as sustain pedal effects, the sympathetic vibration of strings, and the wide pitch range, may impose challenges to systems that are designed to follow ensembles of voices, strings, and wind instruments. In particular, we argue that the sustain-pedal effects are especially challenging. When the pedal is pressed, notes played will continue to sound until the string vibration naturally ceases. This makes the notes longer than their notated lengths and overlap with later notes, which causes potential mismatch between audio and score.

Note that Niedermayer et al. reported negligible influence of sustain-pedal effects on alignment results in their experimental study on audio-score alignment [22]. However, they further reasoned that this might be because the dataset used for evaluation contains only Mozart pieces, in which "the usage of pedals plays a relatively minor role". In fact, the sustain pedal has been commonly used since the Romantic era (after Mozart) in the Western music history, and is widely used in modern piano performances of many different styles. Another reason for Niedermayer et al.'s observation, we argue, is that the algorithm used for evaluation was an offline algorithm, which is more robust to the local mismatch between audio and score as a global alignment is employed. For online algorithms, however, they are more sensitive to local audio-score mismatch and they can be totally lost during the following process.

In this paper, we build a system to follow piano performances, based on the state-space framework proposed by Duan and Pardo [12]. More specifically, we propose an approach to deal with the mismatch issue caused by sustain-pedal effects. In each inter-onset segment of the audio, we remove partials of all notes extended from the previous segment and only retain partials of the new notes. This operation reduces sustain-pedal effects by weakening the match between an audio frame and the previous wrong score positions, which are the most competitive wrong candidates. But we need to mention another case that the match between this audio frame and the current correct score position may be also reduced, if notes in previous frames are actually extended because they are not released yet according to the score instead of due to the sustain pedal. Nevertheless, as explained in detail in Section 3.4, this operation still favors the correct position even in this case. We conduct experiments on 25 synthetic and 25 real piano performances randomly chosen from the MAPS dataset [14]. Results show that the proposed system significantly outperforms the baseline system [12] on both alignment accuracy and robustness.

2. SYSTEM FRAMEWORK

We build our system based on the state-space model proposed in [12], which follows polyphonic audio with its score. Music audio is segmented into time frames and fed into the system in sequence. Each frame $\mathbf{y}_n$ is associated with a 2-dimensional state vector $\mathbf{s}_n = (x_n, v_n)^T$, representing its underlying score position (in beats) and tempo (in beats-per-minute), respectively. The goal of score following is to infer the score position x_n from current and previous audio observations $\mathbf{y}_1, \cdots, \mathbf{y}_n$. This is formulated as an online inference problem of hidden states of a hidden Markov process, which is achieved through particle filtering. The hidden Markov process contains two parts: a process model and an observation model.

The process model describes state transition probabilities $p(\mathbf{s}_n \mid \mathbf{s}_{n-1})$ by two dynamic equations for x_n and v_n, respectively. The score position advances from the previous position according to the tempo. The tempo changes through a random walk or does not change at all, depending on where the position is.

The observation model $p(\mathbf{y}_n \mid \mathbf{s}_n)$ evaluates the match between an audio frame and the hypothesized state on the pitch content. A good match is achieved when the audio frame contains exactly the pitches described on the score at the hypothesized score position in the state. Otherwise, a bad match is achieved. This is calculated using the multi-pitch likelihood model proposed in [13], which evaluates the likelihood of a hypothesized pitch set in explaining the magnitude spectrum of an audio frame.

The multi-pitch likelihood model detects prominent peaks in the magnitude spectrum of the audio frame and represents them as frequency-amplitude pairs:

$$\mathcal{P} = \{\langle f_i, a_i \rangle\}_{i=1}^{K}, \tag{1}$$

where K is the total number of peaks detected in the frame. The likelihood would be high if the harmonics of the hypothesized pitch set match well with the detected peaks in terms of both frequency and amplitude. The likelihood would be low otherwise, for example, if many harmonics are far away from any detected peak.

3. PROPOSED METHOD

3.1 Properties of Piano Music

There are many specific properties of piano music, such as the wide pitch range and the inharmonicity of note partials. In this section, we discuss two properties considered in the proposed approach: strong onset with exponential decay of the note waveform, and the sustain-pedal effects.

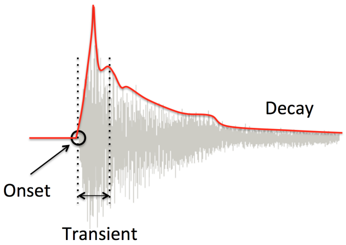


Figure 1. Waveform and energy envelope of a piano note.

Figure 1 shows the waveform and energy envelope of a piano note. We see a sudden energy increase at the onset followed by an exponential decay. When a piano key is pressed, its damper is released and its hammer strikes the strings, which yields an impulse-like articulation. The damper continues to be released as the key is being pressed. This lets the string vibration decay naturally, which may take as long as 10 seconds. The damper comes back to the strings when the key is released, and the string vibration ceases quickly. However, when the sustain pedal is pressed, all dampers of all keys are released no matter if a key is pressed or not. This allows all active notes to continue to sound, and even activate some inactive notes due to sympathetic vibrations, which enriches the sound timbre.

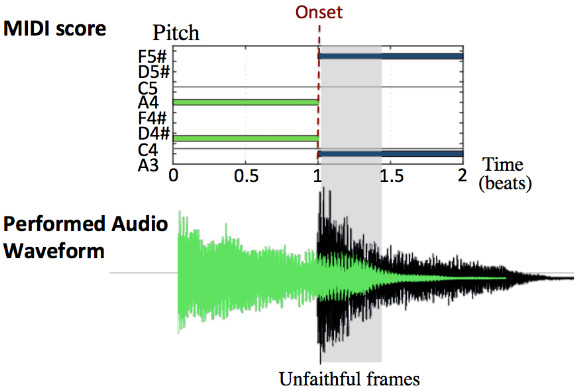


Figure 2. Mismatch between audio and score caused by the sustain-pedal effects.

A detailed analysis of the sustain-pedal effects is given by Lehtonen et al. in [21]. Here we focus on its resulted mismatch problem between audio and score. Figure 2 shows the MIDI score (in pianoroll) and waveforms of four notes. According to the score, the first two notes are supposed to end when the latter ones start. However, due to the sustain pedal, the waveforms of the first two notes are extended into those of the latter. This causes potential mismatch between the audio and the score, especially in frames right after the onset of the latter notes. In other words, the audio is unfaithful to the score in those frames. The degree and the length of the unfaithfulness, however, is not notated in the score. It depends on the the notes being played as well as how hard the performer presses the pedal. If the pedal is pressed partially, then the damper will slightly touch the strings and the effects are slighter. While some composers and music editors use pedal marks to notate it, appropriate use of the sustain pedal is more often left to the performer.

The main idea of the proposed approach to deal with the sustain-pedal effects is to first detect audio onsets to locate the potentially unfaithful frames. Then partials of the extended notes are removed in the peak representation of these frames. We describe the two steps in the following.

3.2 Onset Detection

Although not all frames right after an onset are unfaithful, as notes could be extended because their keys are still pressed according to the score, many unfaithful frames do appear right after onsets. Therefore, onset detection helps to locate potentially unfaithful frames. Many onset detection methods have been proposed in the literature [2]. In this paper, we adopt the widely used spectral-based approach, since it is effective for polyphonic signals. We adapt it to online scenarios for our score following system.

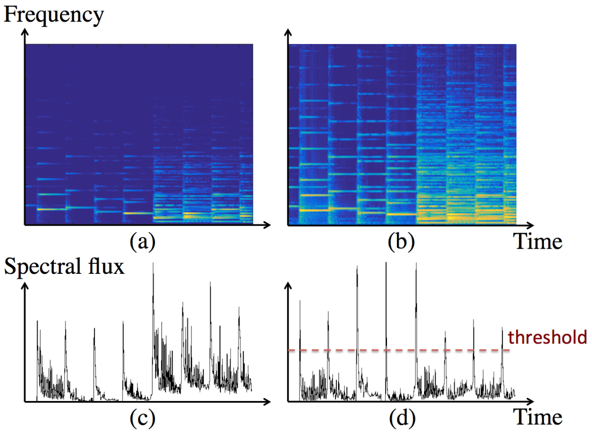


Figure 3. Illustration of onset detection. (a) Spectrogram. (b) Spectrogram after compression. (c) Spectral flux. (d) Normalized spectral flux by signal energy.

Figure 3 illustrates the onset detection process. We first calculate the audio magnitude spectrogram $\mathbf{Y}(n,k)$ through Short-time Fourier Transform (STFT) in Figure 3(a), where n and k are frame and frequency bin indices, respectively. We then apply logarithmic compression on it to enhance the high-frequency content by

$$\tilde{\mathbf{Y}}(n,k) = \log\left(1 + \gamma \cdot \mathbf{Y}(n,k)\right), \qquad (2)$$

where γ controls the compression ratio. This is because high frequency content is indicative for onsets but relatively weak in the original spectrogram [27]. Figure 3(b) shows the enhanced magnitude spectrogram with $\gamma = 0.2$. We then compute the spectral flux $\Delta_{\mathbf{Y}}(n)$ by summing positive temporal differences across all frequency bins as

$$\Delta_{\mathbf{Y}}(n) = \sum_k \left| \tilde{\mathbf{Y}}(n,k) - \tilde{\mathbf{Y}}(n-1,k) \right|_{\geq 0}, \qquad (3)$$

where $|\cdot|_{\geq 0}$ denotes half-wave rectification, i.e., keeping non-negative values while setting negative values to 0. The calculated spectral flux is shown in Figure 3(c). We can see that all onsets in the example are associated with a clear peak, however, peak heights vary much. Spurious peaks in the middle of louder notes are as high as true peaks of softer notes. One could set an adaptive threshold which varies with the moving average of the spectral flux, but this would make the onset detection algorithm offline. Instead, we normalize the spectral flux by the energy of the audio signal in the current frame by

$$\tilde{\Delta}_{\mathbf{Y}}(n) = \Delta_{\mathbf{Y}}(n)/E(n), \qquad (4)$$

where $E(n)$ is the Root-Mean-Squre (RMS) value of the n-th frame of the audio. After this operation, a simple threshold can detect the onsets, as shown in Figure 3(d).

Note that onset detection has been used in several online [5] and offline [15] alignment algorithms, where a special matching function is used to match audio and score onsets. In our system, however, onset detection is to locate potentially unfaithful audio frames. Their audio representations are modified but no special matching function is defined.

3.3 Reduce Pedal Effects by Spectral Peak Removal

Frames within a period after a detected onset are potentially unfaithful frames due to the sustain pedal. Conservatively, without knowledge of the degree and length of the effects, we just reduce them in the first 200ms (i.e., 20 frames) following an onset. As described in Section 2, each audio frame is represented by a set of significant spectral peaks in Eq. (1). The match between the audio frame and a hypothesized score location is evaluated through the multi-pitch likelihood model on how well the harmonics of the score notes match with spectral peaks in the audio. As the spectrum of an unfaithful audio frame contains unexpected peaks corresponding to partials of notes extended by the sustain pedal, we propose to remove these peaks to reduce the mismatch between audio and score.

Figure 4 illustrates the idea. For each potentially unfaithful frame (e.g., the n-th frame), we compare its spectral peaks with those in a frame before the onset (e.g., the m-th frame), and remove peaks that seem to be extended from the earlier frame. Let $\mathcal{P}_m = \{\langle f_i^m, a_i^m \rangle\}_{i=1}^{K_m}$ be the total K_m peaks detected in the m-th frame, and $\mathcal{P}_n = \{\langle f_j^n, a_j^n \rangle\}_{j=1}^{K_n}$ be the total K_n peaks detected in the n-th frame. A peak in the n-th frame whose frequency is very close to and whose amplitude is smaller than those of a peak in the m-th frame is considered as an extension and is removed. Note that repeated notes will not be removed in this way as the amplitude criterion is not met. Extended partials that are overlapped with a partial of a new note will not be removed either due to the same reason. After peak removal, a new spectral peak representation of the n-th frame is obtained as

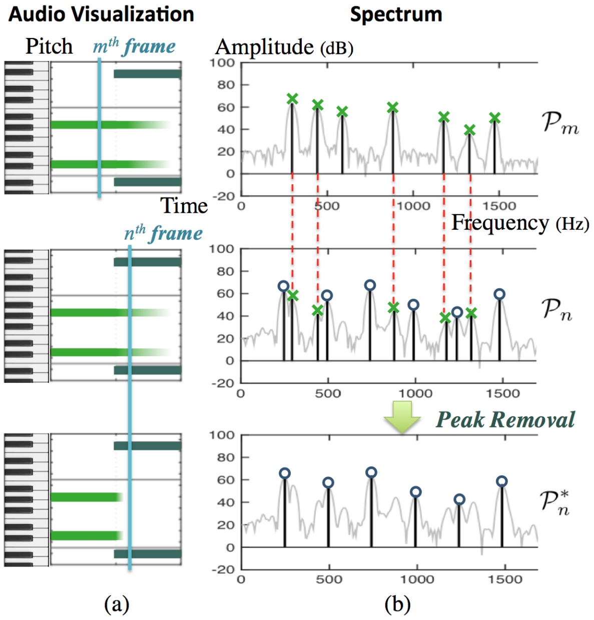


Figure 4. Illustration of the spectral peak removal idea. (a) Audio performance representation before and after peak removal. (b) Magnitude spectra with spectral peaks in the m-th and n-th frames. Peaks marked by crosses correspond to the first two notes. Peaks marked by circles correspond to the latter two notes.

$$\mathcal{P}_n^* = \mathcal{P}_n - \left\{ \langle f_i^n, a_i^n \rangle : \exists j \text{ s.t. } |f_i^n - f_j^m| < d, a_i^n < a_j^m \right\}, \qquad (5)$$

where $\langle f_i^n, a_i^n \rangle \in \mathcal{P}_m$. d is the threshold for the allowable frequency deviation, which is set to a quarter tone in this paper. Finally, the match between the n-th frame and a hypothesized score position is evaluated through the multi-pitch likelihood of score-indicated pitches in explaining the modified peak representation of the spectrum. Note that this operation only modifies the peak representation of the audio instead of the audio itself.

The peak removal operation emphasizes new notes in the representation and discards old ones. This is in accordance to music perception, as we always pay more attention to new notes even though the old notes are as loud.

3.4 New Mismatch Introduced by Peak Removal

The peak removal operation removes notes extended by the sustain pedal in the audio representation, however, it also removes notes that should remain according to the score, e.g, D4 in Figure 5(a). This causes new mismatch between audio and score. Ideally, we could differentiate these two kinds of notes from the note offset information in a well-aligned score, which we do not have during score following. Nevertheless, we explain in the following that the new mismatch actually still helps with score following.

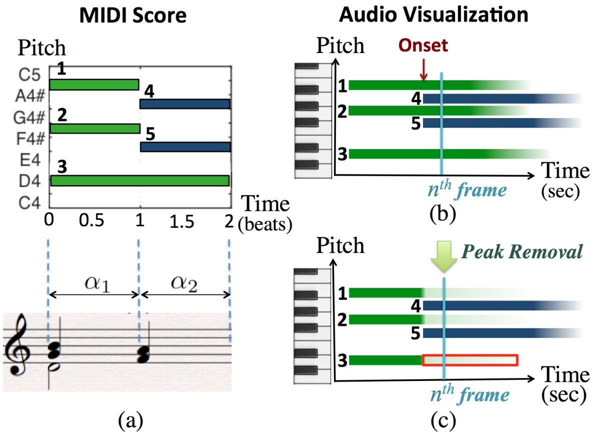


Figure 5. Illustration of mismatch reduced and introduced by the peak removal operation. (a) MIDI score and its piano-roll representation. (b) Audio performance representation before peak removal. (c) Audio performance representation after peak removal.

Figure 5 illustrates the mismatch reduced and introduced by the peak removal operation. A MIDI score with two inter-onset segments α_1 and α_2 is shown in Figure 5(a). Notes 1 and 2 are supposed to end when Notes 4 and 5 start, while Note 3 is supposed to span both segments. For an audio frame right after the onset (e.g., the n-th frame) in Figure 5(b), we can see that it contains all the five notes, including Notes 1 and 2 due to the sustain pedal. It is therefore unfaithful to the correct segment α_2 in the score. Which segment is a better match to this audio frame? Out of the 5 notes in the n-th frame, α_1 contains 3 (Notes 1, 2, and 3) and α_2 also contains 3 (Notes 4, 5, and 3). The correct segment α_2 does not show a better match than α_1.

Suppose the audio onset of Note 4 and 5 is detected, then the peak removal operation will remove spectral peaks corresponding to Notes 1, 2, and 3 in the n-th frame. The mismatch between the n-th frame and the correct segment α_2 due to the sustain pedal is reduced, while new mismatch is introduced as Note 3 is supposed to stay in α_2 in the score but is removed in the audio. This leaves 2 notes (Notes 4 and 5) shared by the score and the audio, although the score has 1 more note (Note 3). The mismatch between the n-th frame and α_1, on the other hand, is increased significantly. There becomes no intersection at all between notes remained in the n-th frame (Notes 4 and 5) and notes in α_1 (Notes 1, 2, and 3). Therefore, the correct segment α_2 is clearly a better match to the n-th frame.

In general, the peak removal operation may introduce mismatch between an audio frame and its correct score location as it may remove peaks that are supposed to stay, but the mismatch between the audio frame and the previous wrong score location will be increased much more. In fact, there will be no match at all. This is true even if all notes in α_1 stay in α_2 according to the score. Therefore, the mismatch introduced by the peak removal operation is not harmful to but actually helps with score following.

In Figure 5, we only consider the previous segment α_1 as a wrong segment to compete with α_2. This is because it is the most common error caused by the sustain pedal in score following. The peak removal operation, however, can help eliminate non-immediate segments that are prior to the current segment as well.

4. EXPERIMENTS

4.1 Data Set and Evaluation Measures

We use the MAPS dataset [14] to evaluate the proposed approach. In this dataset, performers first play on a MIDI keyboard, then the MIDI performances are rendered into audio by a software synthesizer or a Yamaha Disklavier. The former are synthetic recordings while the latter are real acoustic recordings. Both have exactly the same timing as the MIDI performances. We randomly select 25 synthetic pieces and 25 real pieces from the dataset. The synthetic pieces simulate the “Bechstein D 280” piano in a concert hall, and the real pieces are recorded with an upright Disklavier piano. Approximately 18 synthetic pieces and 10 real pieces are played with substantial sustain pedal usage. We then download their MIDI scores from `http://piano-midi.de/`. Note that the MIDI performances have minor differences from the MIDI scores besides their tempo difference. These include occasionally missed or added notes, different renderings of trills, and slight desynchronization of simultaneous notes. We therefore perform an offline DTW algorithm to align the MIDI performances to the MIDI scores and then manually correct minor errors to obtain the ground-truth alignment.

We calculate the time deviation (in ms) between the ground-truth alignment and the system’s output alignment of the onset of each score note. This value ranges from 0ms to the total length of the audio. We define its average over all notes in a piece as the *Average Time Deviation (ATD)*.

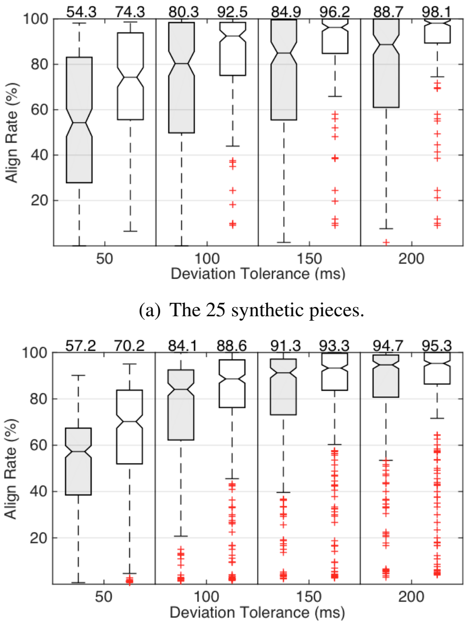
We also calculate the *Align Rate (AR)* [7] for all pieces. It is defined as the percentage of correctly aligned notes, those whose time deviation is less than a threshold. Commonly used thresholds range from 50ms to 200ms depending on the application. For an automatic accompaniment system, a deviation less than 50ms would be required, while for an automatic page turner, 200ms would be fine.

4.2 Implementation Details

Our score following system is built upon the system proposed in [12], whose source code can be downloaded at the authors’ website. We therefore take it as the baseline system for comparison. We use the authors’ original code and parameter settings in both the baseline system and the proposed system. The multi-pitch likelihood model in [12] was trained on thousands of randomly mixed chords using notes of 16 kinds of Western instruments excluding piano. We stick with this model in the proposed system for a fair comparison. For unique parameters of the proposed system, we set γ to 0.2 in Eq. (2), the threshold in Figure 3 to 225, the length of unfaithful region to 200ms after each detected onset, the frame to compare with to the 5-th frame before the onset, and the peak frequency deviation d in Eq.

(5) to a quarter tone. All these parameters are fixed for all pieces. Due to the probabilistic nature of the baseline system and the proposed system, we run 10 times of each system on each piece for the comparison.

4.3 Results



(a) The 25 synthetic pieces.

(b) The 25 real pieces.

Figure 6. Align Rate comparisons between the baseline [12] (grey) and the proposed (white) systems using different time deviation tolerrances. Numbers above the figures show medians of the boxes.

Figure 6 shows box plots of align rates of the two systems with different onset deviation tolerance values on both synthetic and real pieces. Each box in Figure 6(a) represents 250 data points (10 runs on 25 pieces) and each box in Figure 6(b) represents 250 data points. We can see that for the synthetic pieces, the median align rate is significantly improved for all tolerance values. The dispersion of the distribution is also significantly shrunk, making the improvement on some low-performing piece-runs especially significant. For the real pieces, the median align rate is significantly improved for all tolerance values except 200ms. The dispersion of the distribution is shrunk significantly for all tolerances except 50ms. This shows that the proposed approach improves the alignment accuracy and robustness significantly on both synthetic and real pieces. The improvement on synthetic pieces is more remarkable because there are more synthetic pieces with a substantial pedal usage. However, the proposed system also has more low-performing outliers on the real pieces, some of which correspond to piece-runs when the system is lost.

Figure 7 compares the Average Time Deviation (ATD) between the two systems on all piece-runs. Again, each box in the synthetic setting contains 250 points and each box in the real setting contains 250 points. We can see that the median ATD in both cases is reduced by the proposed system. The reduction on the synthetic pieces is even more significant. The dispersion of the distribution is also shrunk significantly, reducing the worst ATD (excluding outliers) from 200-300ms to the range under 200ms. After the improvement, a fair amount of synthetic and real piece-runs have ATD under 50ms, which would enable real-time applications such as automatic accompaniment.

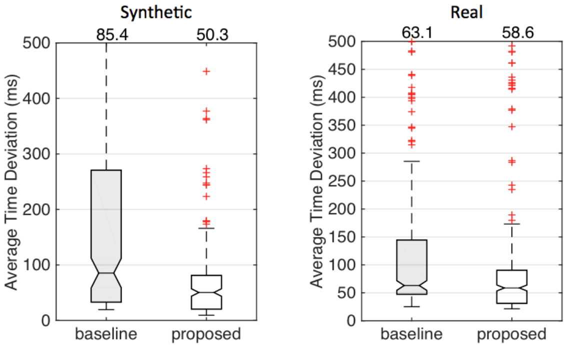


Figure 7. Average time deviation comparison between the baseline [12] and the proposed system. Outliers that exceed 500ms are not shown in this figure. Several outliers are higher than 3 seconds. Numbers above the figure show medians of the boxes.

Examples of alignment results can be found at `http://www.ece.rochester.edu/users/bli23/projects/pianofollowing`.

5. CONCLUSIONS

In this paper we proposed an approach to follow piano performances with sustain-pedal effects. The usage of the sustain pedal extends notes even if their keys have been released, hence causes mismatch between audio and score, especially in frames right after note onsets. To address this problem, we first detect audio onsets to locate these potentially unfaithful frames. We then remove spectral peaks that correspond to the extended notes in these frames. This operation reduces the mismatch caused by the sustain-pedal effects at the expense of introducing potential new mismatch caused by the removal of notes whose keys have not been released. However, we analyzed that this operation still helps the system to favor the correct score position even in this case. Experimental results on both synthetic and real piano recordings show that the proposed approach improved the alignment accuracy and robustness significantly over the baseline system.

For future work, we plan to consider other specific properties of piano music to improve the alignment performance. For example, alignment of audio and score onsets can provide "anchors" for the alignment, and we can define a special matching function that models the transient-like property to align onsets. In addition, for the sustain part, a time-varying matching function that considers the exponential energy decay would improve the alignment accuracy within a note.

6. REFERENCES

[1] A. Arzt, G. Widmer, and S. Dixon. Automatic page turning for musicians via real-time machine listening. In *Proc. European Conference on Artificial Intelligence (ECAI)*, 2008.

[2] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler. A tutorial on onset detection in music signals. *IEEE Trans. Speech and Audio Processing*, 13(5):1035–1047, 2005.

[3] E. Benetos, A. Klapuri, and S. Dixon. Score-informed transcription for automatic piano tutoring. In *Proc. European Signal Processing Conference (EUSIPCO)*, 2012.

[4] P. Cano, A. Loscos, and J. Bonada. Score-performance matching using HMMs. In *Proc. ICMC*, 1999.

[5] A. Cont. Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical HMMs. In *Proc. ICASSP*, 2006.

[6] A. Cont. A coupled duration-focused architecture for real-time music-to-score alignment. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(6):974–987, 2010.

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

[8] R. Dannenberg and C. Raphael. Music score alignment and computer accompaniment. *ACM Communications*, 49(8):39–43, 2006.

[9] S. Dixon. Live tracking of musical performances using online time warping. In *Proc. International Conference on Digital Audio Effects (DAFx)*, 2005.

[10] S. Dixon and G. Widmer. Match: A music alignment tool chest. In *Proc. ISMIR*, 2005.

[11] Z. Duan and B. Pardo. Soundprism: An online system for score-informed source separation of music audio. *Journal of Selected Topics in Signal Processing*, 5(6):1205–1215, 2010.

[12] Z. Duan and B. Pardo. A state space model for online polyphonic audio-score alignment. In *Proc. ICASSP*, 2011.

[13] Z. Duan, B. Pardo, and C. Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Trans. Audio Speech and Lang. Process*, 18(8):2121–2133, 2010.

[14] V. Emiya, R. Badeau, and B. David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Trans. Audio, Speech, and Language Process.*, 18(6):1643–1654, 2010.

[15] S. Ewert, M. Muller, and P. Grosche. High resolution audio synchronization using chroma onset features. In *Proc. ICASSP*, 2009.

[16] L. Grubb and R.B. Dannenberg. A stochastic method of tracking a vocal performer. In *Proc. ICMC*, 1997.

[17] N. Hu, R.B. Dannenberg, and G. Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003.

[18] T. Itohara, K. Nakadai, T. Ogata, and H.G. Okuno. Improvement of audio-visual score following in robot ensemble with human guitarist. In *Proc. IEEE-RAS International Conference on Humanoid Robots*, 2012.

[19] C. Joder, S. Essid, and G. Richard. Learning optimal features for polyphonic audio-to-score alignment. *IEEE Trans. Audio, Speech, Language Process.*, 21(10):2118–2128, 2013.

[20] C. Joder and B. Schuller. Off-line refinement of audio-to-score alignment by observation template adaptation. In *Proc. ICASSP*, 2013.

[21] H. M. Lehtonen, H. Penttinen, J. Rauhala, and V. Valimaki. Analysis and modeling of piano sustain-pedal effects. *Journal of the Acoustical Society of America*, 122(3):1787–1797, 2007.

[22] B. Niedermayer, S. Bck, and G. Widmer. On the importance of real audio data for mir algorithm evaluation at the note-level - a comparative study. In *Proc. ISMIR*, 2011.

[23] N. Orio and F. Dechelle. Score following using spectral analysis and hidden markov models. In *Proc. ICMC*, 2001.

[24] N. Orio and D. Schwarz. Alignment of monophonic and polyphonic music to a score. In *Proc. ICMC*, 2001.

[25] M. Puckette. Score following using the sung voice. In *Proc. ICMC*, 1995.

[26] C Raphael. A bayesian network for real-time musical accompaniment. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2001.

[27] X. Rodet and F. Jaillet. Detection and modeling of fast attack transients. In *Proc. ICMC*, 2001.

[28] V. Thomas, C. Fremerey, D. Damm, and M. Clausen. Slave: a score-lyrics-audio-video-explorer. In *Proc. ISMIR*, 2009.

[29] T. M. Wang, P.Y. Tsai, and A.W.Y. Su. Score-informed pitch-wise alignment using score-driven non-negative matrix factorization. In *Proc. IEEE International Conference on Audio, Language and Image Processing (ICALIP)*.