AN EFFICIENT METHOD FOR POLYPHONIC AUDIO-TO-SCORE ALIGNMENT USING ONSET DETECTION AND CONSTANT Q TRANSFORM

Chun-Ta Chen
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan
chun-ta.chen@mirlab.org

Jyh-Shing Roger Jang
Department of Computer Science
National Taiwan University
Taipei, Taiwan
jang@mirlab.org

Wen-Shan Liu
Institute for Information Industry, Taiwan
wsliou@iii.org.tw

Chi-Yao Weng
Department of Computer Science
National Pingtung University
Pingtung, Taiwan
cyweng@mail.nptu.edu.tw

ABSTRACT

This paper proposes an innovative method that aligns a polyphonic audio recording of music to its corresponding symbolic score. In the first step, we perform onset detection and then apply constant Q transform around each onset. A similarity matrix is computed by using a scoring function which evaluates the similarity between notes in the music score and onsets in the audio recording. At last, we use dynamic programming to extract the best alignment path in the similarity matrix. We compared two onset detectors and two note matching methods. Our method is more efficient and has higher precision than the traditional chroma-based DTW method. Our algorithm achieved the best precision, which are 10% higher than the compared traditional algorithm when the tolerance window is 50 ms.

Index Terms**—**Music synchronization, audio-to-score alignment

1. INTRODUCTION

The goal of audio-to-score alignment is to find a mapping between an audio performance and its symbolic musical score [1, 2]. In other words, the alignment analyzes the content of input audio at some time point and maps it to a corresponding time point on its score with similar music structure. In general, audio-to-score alignment can be implemented as either a real-time or offline system. A real-time system, also called score follower, can be used to align a musical score to a live performance and, further, can take part in musical interactions with human musicians in real-time in order to achieve goals such as automatic accompaniment [3] or automatic page turning [4]. On the other hand, in an off-line scenario, there is no real-time constraint and the complete information of the given audio signals can be used in the alignment process. Therefore, the alignment performance of off-line cases is usually better than that of real-time cases. The offline alignment system can be used to retrieve the best matching MIDI file in a database for an audio query [5]. It would also allow indexing timestamps of an audio according to the desired measure or passage on the musical score [6]. Moreover, alignment system provides the information of performance errors and temporal deviation of note events, so it can be used for music performance evaluation or music education.

The standard approach to audio-to-score alignment involves three steps: feature extraction, distance/similarity computation, and alignment. Step 1 aims to extract from audio signal informative features that characterize the music contents, like pitch, chord, or onset. In step 2, we need to define a distance or similarity function to measure the difference between features of audio recording and the note events in the score. Step 3 employs an alignment algorithm to find the best match between the feature sequence and the score. Note that the tempo of audio performance may be unstable and deviate from the tempo of its score. Besides, there may be minor inconsistence in the notes between audio performance and score. For example, musician possibly loses notes inadvertently or add ornaments intentionally. As a result, a good alignment algorithm should take these issues into consideration. Dynamic programming based alignment, such as dynamic time warping (DTW), can cope with tempo fluctuation of audio signals. Therefore, DTW is extensively used in the audio-to-score alignment system.

The problem of audio-to-score alignment was first introduced in 1984 by R. B. Dannenberg [7] who presented algorithms for following a monophonic soloist in a score and synchronizing the accompaniment. Early audio-to-score alignment techniques mostly dealt with monophonic scores since polyphonic music alignment was much harder at that time. Most polyphonic audio-to-score alignment algorithms adopt a similar procedure which converts a symbolic score into audio, performs feature extraction on the original and converted audio pieces, and then align two sequences of features. Orio and Schwarz [8] use DTW to align polyphonic audio file to another audio file that is synthesized from a MIDI file. They use a measure called peak structure distance, which is derived from the spectrum of both audio. In contrast, Dannenberg and Hu [5, 6] proposed a similar scheme, which computes a simple 12-dimensional pitch chroma as input feature and align the performance audio with the synthesized audio by DTW

978-1-4799-9988-0/16/$31.00 ©2016 IEEE

approach. These simple techniques give good results on polyphonic audio signal and do not require a training phase.

There are still other audio-to-score alignment systems that employ probabilistic models for better performance, such as hidden Markov models (HMMs) and conditional random field (CRF) models. These models take into account the uncertainty of the matching to achieve better performance. In such systems, the hidden variable represents the current position in the score. Flexible transition probabilities can also permit possible structure changes. Viterbi algorithm is usually used to search the best alignment path in these models. For example, Cont [9] presented a polyphonic score following system using hierarchical HMM using previously learned pitch templates for multiple fundamental frequency matching. CRF model was first applied to the audio-to-score alignment problem by Joder [10], which incorporates both frame-level and segment-level features, including chroma, onset and tempo information. Joder proposed 3 different CRF models for different choices of tradeoff between accuracy and complexity. The probabilistic models brought more efficient and accurate alignment techniques, and can compete favorably with DTW based methods.

The disadvantage of using synthesized audio (from MIDI or music score) is that its music properties might differ from real-world signals. The input audio would not always have the same timbre and harmonic property as the synthesized audio music. On the other hand, probabilistic models need a time-consuming training phase to fine tune model parameters. Instead, this paper proposes an improved scheme which bypasses the process of music synthesis and the training phase. It observes multiple pitches around onsets based on constant Q transform, and compute the similarity with the notes in the score. Our method is rather robust to variations in harmonic series and the interference of music accompaniments according to our experimental results.

In the next section, we shall describe the proposed algorithm in more detail. Section 3 describes evaluation criteria and experimental results of the proposed algorithm. Finally, Section 4 gives conclusions and future work.

2. POLYPHONIC AUDIO-TO-SCORE ALGORITHM

This section introduces the proposed alignment method, which involves 4 steps (onset segmentation, constant Q transform, note matching, and dynamic programming), as shown in Figure 1. We shall introduce these steps one-by-one in the following sub-sections.

2.1. Onset detection

Onset detection is commonly used as a basic step for further music analysis tasks, such as beat tracking and music transcription. The general procedure of onset detection is to compute an onset strength function (OSF) from the input audio, and then pick local maxima from OSF as onsets. In our implementation, we tested two onset detection methods. First one is spectral flux proposed by S. Dixon [11]. Spectral flux captures onsets with changes in the magnitude spectrum of short-time Fourier transform. The other one is based on a probabilistic model proposed by F. Eyben, et al [12]. It incorporates a recurrent neural network with the spectral magnitude and its first time derivative as input features. It shows good performance on all type of music categories. In our implementation, the frame size is 25 ms and the hop size is 5 ms. To pick reliable peaks in an OSF, we applied a median filter as threshold to remove spurious peaks, and then pick the maximum in a sliding window of 50 ms.

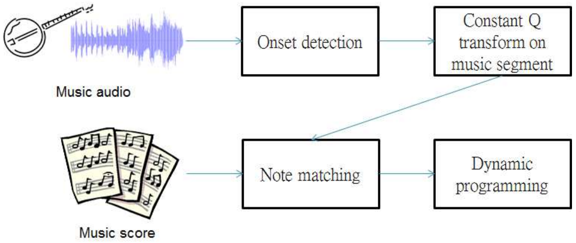


Figure 1. Flowchart of the proposed audio-to-score alignment algorithm

2.2. Constant Q transform

Constant Q transform has been used extensively in music analysis, where the “Q” means the "quality factor", which is the ratio of center-frequency to its bandwidth. Therefore, constant Q means that ratio is constant in all frequency bins of the spectrum. The transform mirrors the human auditory system, where spectral resolution is better at lower frequencies, and temporal resolution improves at higher frequencies [13]. The frequency scale of constant Q transform is logarithmic, so it is particularly useful in music processing since musical transposition corresponds only to translation of frequency bins. The constant Q transform of a signal $x(n)$ can be expressed as follows:

$$C(k) = \frac{1}{N_k}\sum_{n=1}^{N_k} x(n) w_{N_k}(n) e^{\frac{-2\pi jQn}{N_k}} \tag{2}$$

where N_k is the number of samples used to calculate constant Q transform at the frequency f_k. The definition of N_k is $\frac{f_s}{f_k}Q$, where f_s is the sampling rate of the signal $x(n)$. $w_{N_k}(n)$ is a window function with length N_k. We use Hamming window in our implementation.

For each onset, we take the frames immediately before and after the onset for constant Q transform. We choose Q factor in a way such that the pitch range from 27.5 Hz to 22050 Hz is divided into 116 bands, as there is 12 frequency bins in an octave and then each frequency bin directly corresponds to a musical note. After obtaining 2 constant Q spectrums for the each onset, we can estimate the pitches of note concurrence from them as described next.

2.3. Similarity measure

After obtaining the onsets, we have to determine which note-on event triggers an onset according to the spectrum of constant Q transform. We have developed two scoring functions to evaluate the similarity of how the variation of spectrums near an onset is correlated with a note on the score.

2.3.1. Note matching method 1

As described in section 2.2, we can obtain 2 constant-Q spectrums right before and after an onset. Assume dA is the absolute difference of these 2 magnitude spectrum, we can then define a vector T as

$$T(k) = \begin{cases} dA(k), & if\ dA(k) > \epsilon \\ 0, & otherwise \end{cases} \tag{3}$$

where ϵ is a noise margin, which is set to 1/4 of the average of dA. We define that a pitch is matched if its frequency bin k is a local maximum in vector T; otherwise it is unmatched. We can use a binary value ϕ_k to indicate whether a pitch in bin k is matched or not. That is, it is 1 if matched, or 0 otherwise. Here we also take into account the harmonic series of a note by introducing an overtone vector Ω equal to [0, 12, 19, 24, 28, 31], which are the frequency bin index differences between a note and its overtones of constant Q spectrum. Elements of the vector Ω correspond to the 1st to 6th harmonic partials of a note since we divide an octave into 12 frequency bins. For example, if a note has a pitch value of 440Hz, then 37th bin of the spectrum is its first harmonic partial (i.e., fundamental frequency), 59th bin is its 2nd harmonic partial, 66th bin is its 3rd harmonic partial, and so on. We use a scoring function to evaluate how well a note is matched, as shown next:

$$s_i = \sum_{j=1}^{6} \frac{1}{j} \phi_{i+\Omega(j)} \tag{4}$$

where i is the frequency bin corresponding to a note pitch and the term $\frac{1}{j}$ is the weight of its j-th harmonic partial. If there is more than one note with the same onset time, we just sum output of their scoring functions.

2.3.2. Note matching method 2

We can further improve the note matching algorithm by inducing non-negative matrix factorization (NMF) [13] to decompose harmonics of notes. We extend the vector dA into a 116×88 matrix V by replicating dA 88 times. We use NMF to factorize V into a nonnegative matrix W and a nonnegative matrix H such that $V \approx WH$. W is the feature matrix of which dim is 116×88 where each column is a note pitch and its harmonics, which cover from A1 to A8. H is the coefficient matrix of size is 88×88, where all columns are identical to each other. The column h of H is the result of decomposition.

We build chord templates for each chord (i.e., notes with the same onset time) in the score. Chord template t is a vector in which its index corresponds to a note pitch in the chord. If there is a note with pitch i, the corresponding index of the chord template t will be set to 1, and 0 otherwise. We define a scoring function to evaluate similarity between a chord template t and an onset vector v by calculating their dot product $\sum_{i=1}^{116} t_i \times v_i$.

2.4. Dynamic programming

According to the above scoring function, we can build a similarity matrix $S(i,j)$, where i is an onset index of the input audio and j is a chord index of the music score. Each cell in a matrix is the output of scoring function. We use dynamic programming (DP) to find the best path that has the overall maximum similarity. The recursive formula of DP is as follows:

$$D(i,j) = max \begin{Bmatrix} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) + S(i,j) \end{Bmatrix} \tag{5}$$

The alignment path is a sequence of adjacent cells, where each cell indicates a correspondence between an onset in audio performance and a note-on event in the music score. After computing the maximum similarity, we can derive the best alignment path by back tracking the path with the highest accumulated value in matrix D.

3. EXPERIMENTS

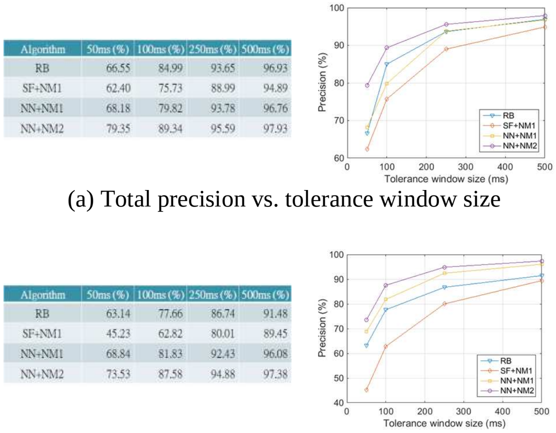
3.1. Dataset and evaluation metrics

The dataset used to evaluate the audio-to-score alignment task contains 56 recordings of human played performance. Each recording has a corresponding MIDI representation of the score. Recordings are in 44.1 KHz 16 bit wav format. The total number of notes is slightly more than 11,000 and the total duration is 54 minutes. The dataset contains two subsets. Subset 1 [14] has 46 recordings extracted from 4 distinct pieces of classic music, which are performed in a monophonic or slightly polyphonic manner. Subset 2 [15] consists of 10 human played J.S. Bach four-part chorales, with 30-sec audio files sampled from real music performances by a quartet of instruments: violin, clarinet, tenor saxophone and bassoon.

Evaluation metrics is based on the proportion of correctly matched notes in the score. A note is said to be correctly matched if its estimated onset does not deviate from the real one by more than a tolerance window (i.e., 50 ms). There are 2 types of figures of merit, piecewise and total precision. Piecewise precision is based on the average of individual precision of each recording. Total precision counts the total matched pairs to derive the overall proportion.

3.2. Performance evaluation

We shall compare the proposed algorithm with some methods of the audio-to-score alignment. The results of performance evaluation are listed in Figure 2. There are total 4 algorithms: “RB” is derived from R. B. Dannenberg’s algorithm [5], which uses chroma feature and DTW alignment. This algorithm is chosen as a baseline in our experiment. The remaining three are based on the proposed methods. “SF+NM1” is an implementation, which employs spectral flux and note matching method 1. “NN+NM1” is based on the neural-network onset detector and the note matching method 1. “NN+NM2” is based on the neural-network onset detector and the note matching method 2. In this table, there are 4 different tolerance windows to evaluate the performance, which are described in the section 3.1.



Algorithm	50ms (%)	100ms (%)	250ms (%)	500ms (%)
RB	66.55	84.99	93.65	96.93
SF+NM1	62.40	75.73	88.99	94.89
NN+NM1	68.18	79.82	93.78	96.76
NN+NM2	79.35	89.34	95.59	97.93

(a) Total precision vs. tolerance window size

Algorithm	50ms (%)	100ms (%)	250ms (%)	500ms (%)
RB	63.14	77.66	86.74	91.48
SF+NM1	45.23	62.82	80.01	89.45
NN+NM1	68.84	81.83	92.43	96.08
NN+NM2	73.53	87.58	94.88	97.38

(b) Piecewise precision vs. tolerance window size

Figure 2. Experimental results: The algorithm RB is a used as a baseline. The remains are our proposed methods.

From the Figure 2, it is obvious that “NN+NM2” can achieve the best performance in both total precision and piecewise precision in all tolerance windows. Besides, the precision of “SF+NM1” is lower than baseline. This is because the spectral flux does not capture well soft onsets. When we change to the onset detector based on neural network, which learned all types of music onset cases, the performance of “NN+NM1” is obviously improved. However, the total precision of “NN+NM1” is still lower than the baseline around the window size 100 ms, as shown in Figure 2(a). Therefore, we improved the note matching method as described in section 3.2, and the total precision 79.35% and the piecewise precision 73.53% of “NN+NM2” are 10% higher than the baseline in the tolerance window 50 ms. As a whole, the algorithm “NN+NM2” achieves the best result in our audio-to-score alignment experiments.

4. CONCLUSIONS AND FUTURE WORK

We have proposed an algorithm that aligns a polyphonic audio recording of music to its corresponding symbolic score. The algorithm involves 4 steps, including onset detection, constant Q transform around onsets, similarity between onsets in audio and note-on in score, and dynamic programing to find the optimum mapping path. Since the proposed method does not rely on timber features of all frames for alignment, it is more efficient in terms of computation. It is also more reliable when the timbres (or the instruments) in audio are different from those specified in the music score. We compared two onset detectors and two note matching methods in the experiments. Our method is more efficient and has higher precision than traditional chroma-based DTW method. When the tolerance window is 50 ms, the best total precision and the piece-wise precision are 79.35% and 73.53%, respectively, which are 10% higher than the compared baseline algorithm.

Although the proposed method performs well in the experiments, it still has room for further improvement. Future work of this research will be focused on improving the alignment algorithm in various aspects. For one thing, the scoring function we used is straightforward that it does not consider the complicated case where different notes coincidence. Moreover, we shall use machine learning to analyze the spectral pattern of constant Q transform in order to come up with a better similarity matrix. Finally, we shall develop a real-time score following system of polyphonic music performances to demonstrate the feasibility of the proposed algorithm.

5. REFERENCES

[1] Alexander Lerch, An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics, John Wiley & Sons, Inc., New Jersey, 2012.

[2] N Orio, S Lemouton, D Schwarz, “Score following: State of the art and new developments,” Proceedings of the 2003 conference on New interfaces for musical expression), Montreal, Canada, pp. 34-41, 2003.

[3] Shinji Sako, Ryuichi Yamamoto, Tadashi Kitamura, “Ryry: A Real-Time Score-Following Automatic Accompaniment Playback System Capable of Real Performances with Errors, Repeats and Jumps,” Active Media Technology: 10th International Conference, AMT 2014, Warsaw, Poland, pp134-145, August 2014.

[4] Arzt, A., Widmer, G., Dixon, S., “Automatic page turning for musicians via real-time machine listening,” Proc. of European Conference on Artificial Intelligence (ECAI), pp. 241–245, 2008.

[5] Hu, N., Dannenberg, R., Tzanetakis, G., “Polyphonic audio matching and alignment for music retrieval,” Proc. IEEE WASPAA, New Paltz, NY, October 2003.

[6] Dannenberg and Hu, “Polyphonic Audio Matching for Score Following and Intelligent Audio Editors,” Proceedings of the 2003 International Computer Music Conference, San Francisco: International Computer Music Association, pp. 27-34, 2003.

[7] Dannenberg, R, “An on-line algorithm for real time accompaniment,” Proceedings of the 1984 International Computer Music Conference, pp. 193-198, 1984.

[8] Orio, N., & Schwarz, D., “Alignment of Monophonic and Polyphonic Music to a Score,” Proc. 2001 ICMC, pp. 155-158, 2001.

[9] Arshia Cont, “Realtime Audio to Score Alignment for Polyphonic Music Instruments, using Sparse Non-Negative Constraints and Hierarchical HMMS,” ICASSP 2006: 245-248, 2006.

[10] Joder, C., Essid, S., Richard, G., “A conditional random field framework for robust and scalable audio-to-score matching,” IEEE Transactions on Audio, Speech and Language Processing 19 (8), 2385–2397, 2011.

[11] S. Dixon, “Onset Detection Revisited,” Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06), pp. 133–137, September 2006.

[12] F. Eyben, S. Böck, B. Schuller, and A. Graves: “Universal onset detection with bidirectional long short-term memory neural networks,” Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR), pp. 589–594, 2010.

[13] Judith C. Brown, “Calculation of a constant Q spectral transform,” J. Acoust. Soc. Am., 89(1), pp. 425–434, 1991.

[14] Arshia Cont, Diemo Schwarz, Norbert Schnell, Christopher Raphael, “Evaluation of Real-Time Audio-to-Score Alignment,” International Symposium on Music Information Retrieval (ISMIR), 2007, Vienna, Austria. 2007.

[15] Zhiyao Duan and Bryan Pardo, “Soundprism: an online system for score-informed source separation of music audio,” IEEE Journal of Selected Topics in Signal Process., vol. 5, no. 6, pp. 1205-1215, 2011.