ALGORITHM FOR SCORE FOLLOWING

Chunta 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

***Abstract*—This paper proposes a method that aligns a polyphonic audio recording of music to its corresponding symbolic score.**

I. 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. The standard approach to audio-to-score alignment involves three steps, including feature extraction, distance/similarity computation, and alignment. Note that the tempo of audio performance may deviate from the tempo of its score. Besides, there may be minor inconsistence of notes between audio performance and score. For example, musician possibly losses notes inadvertently or add ornaments intentionally. As a result, a good alignment algorithm should take these issues into consideration.

II. ALGORITHM

The algorithm involves 4 steps, including onset detection, constant Q transform around onsets, similarity comparison between onsets in audio and note-on in score, and dynamic program to find the optimum mapping path.

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 use spectral flux as our onset strength function, as proposed by Dixon [3]. Spectral flux captures onsets with changes in the magnitude spectrum of short-time Fourier transform. To pick reliable peaks, we need to apply a median filter as threshold to remove spurious peaks, and then pick the maximum in a sliding window.

Constant Q transform: 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 A1 (55 Hz) to A9 (14080 Hz) is divided into 96 bands, as there is 12 frequency bins in an octave and then each frequency bin directly corresponds to a musical note.

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 use a scoring function to evaluate the similarity of how the variation of spectrums near an onset is correlated with a note on the score. We define that a pitch is matched if its frequency bin is a local maximum in the constant Q vector, or unmatched if not. Because we deal with polyphonic music, there might be several notes played at the same time. If there are more than one notes with the same onset time, we just sum output of their scoring functions.

Dynamic programming: According the above scoring function, we can build a similarity matrix S. 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:

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

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.

REFERENCES

[1] 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, 34-41, 2003.

[2] 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..

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