A REAL-TIME SCORE FOLLOWER FOR MIREX 2010

Zhiyao Duan
Northwestern University
zhiyaoduan00@gmail.com

Bryan Pardo
Northwestern University
pardo@northwestern.edu

ABSTRACT

This abstract describes a real-time score follower that we submitted to MIREX 2010 "Real-time Audio to Score Alignment (aka. Score Following)" task.

1. INTRODUCTION

A real-time score follower is a program that synchronizes a performance with its score in real time. It estimates a score position for each input time frame of the performance and the estimation is made only using past frames. Researchers have proposed different methods for different score following situations [1–10]. However, the problem remains challenging when the performance is polyphonic audio and the tempo of the performance is not stable.

This extended abstract describes our proposed polyphonic music score follower which uses a state space model. The state space is a 2-d continuous set where the two dimensions are score position and tempo, respectively. State transition is modeled as two dynamic equations. Audio time frames are our observations. The observation model is built as the likelihood of observing the time frame given the pitches of the current state (i.e. at the current score position). The current score position is inferred using particle filtering.

In the following sections, we will describe the method and implementation in detail.

2. METHOD

The state space model we use is a hidden Markov process model. A hidden Markov process has two basic models: a process model and an observation model. The process model describes how the states transit and it satisfies Markov properties. The observation model is the likelihood of seeing an observation given a state.

We decompose the audio performance into time frames before feeding to the algorithm. For the n-th frame $\mathbf{y}_n$, its state is a 2-d vector $[x_n, v_n]^T$, where $x_n \in \mathcal{X}$ is its score position (in beat) and $v_n \in \mathcal{V}$ is its tempo (in Beat Per Minute (BPM)). $\mathcal{X} = [1, M]$ is a continuous set of all score positions, where M is the last beat in the score. $\mathcal{V} = [v_{min}, v_{max}]$ is a continuous set of all tempi from the lowest to the highest. The audio frame itself $\mathbf{y}_n$ is our observation. The algorithm is to infer the current score position x_n given current and previous observations $\mathbf{y}_1, \cdots, \mathbf{y}_n$.

2.1 Process Model

The process model we adopt here are two dynamic equations:

$$x_n = x_{n-1} + l \cdot v_{n-1} \tag{1}$$

$$v_n = \begin{cases} \hat{v}_{n-1} + n_v & \text{if } z_k \in [\hat{x}_{n-1}, \hat{x}_n] \text{ for some } k \\ v_{n-1} & \text{otherwise} \end{cases} \tag{2}$$

where l is the audio frame hop in minute; $n_v \sim \mathcal{N}(0, \sigma_v^2)$ is a Gaussian noise variable; z_k is the k-th note onset time in beat in the score; $\hat{x}_n$ and $\hat{v}_n$ are the estimates of current score position and tempo, respectively, where $\hat{x}_n$ is also the output of the algorithm at the n-th frame.

Eq. (1) presents that the score position of the current audio frame is determined by the score position of the previous audio frame and the tempo. Eq. (2) presents that if the current score position has just passed a note onset, then the tempo will be initialized around the current tempo estimate with a Gaussian distribution. Otherwise, the tempo will remain the same. We can see that randomness is only introduced in tempo instead of position. In this way, we can make sure that the score position estimates progress smoothly. In addition, the randomness is only introduced when the estimated score position has just passed a note onset. This is intuitive, since the only information that we can use to update tempo estimate is note onsets. If the audio performance is in the middle of a note, we have no clue to change the tempo estimate.

2.2 Observation Model

The observation model we use is the likelihood of observing the current audio frame given the current score position $p(\mathbf{y}_n|x_n)$. In a frame of polyphonic music played by harmonic instruments, multiple pitches are the most informative objects. Therefore, we use multi-pitch observation likelihood as our observation model. The multi-pitch observation likelihood was developed in our previous work [11, 12].

The frame of audio is first transformed to the frequency domain by Short Time Fourier Transform (STFT). Then significant peaks of the power spectrum are detected and represented as a frequency-amplitude pair (f_i, a_i). Non-peak regions of the power spectrum are also extracted. The

multi-pitch observation likelihood is defined as the likelihood of observing the peaks and the non-peak regions given the multiple pitches in this frame $\boldsymbol{\theta} = \{F0_1, \cdots, F0_J\}$.

$$p(\mathbf{y}_n|x_n) = \mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{peak region}}(\boldsymbol{\theta}) \cdot \mathcal{L}_{\text{non-peak region}}(\boldsymbol{\theta}) \quad (3)$$

where we assume spectral bins to be independent given multiple pitches, hence the peak region and non-peak region are conditionally independent. For the same reason, spectral peaks are also conditionally independent in the peak region likelihood:

$$\mathcal{L}_{\text{peak region}}(\boldsymbol{\theta}) = \prod_{k=1}^{K} p\left(f_k, a_k|\boldsymbol{\theta}\right) \quad (4)$$

$$\mathcal{L}_{\text{non-peak region}}(\boldsymbol{\theta}) \approx \prod_{F_0 \in \boldsymbol{\theta}} \prod_{\substack{h \in \{1 \cdots H\} \\ F_h \in \mathcal{F}_{\text{np}}}} 1 - P\left(e_h = 1|F_0\right) \quad (5)$$

where F_h is the frequency of the predicted h-th harmonic of F_0; e_h is the binary variable that indicates whether this harmonic is detected; $\mathcal{F}_{\text{np}}$ is the set of frequencies in the non-peak region; and H is the largest harmonic number we consider.

3. IMPLEMENTATION

Given the process model and the observation model, we want to infer the current score position from current and past observations, i.e. to estimate the posterior probability $p(x_n|\mathbf{y}_1, \cdots, \mathbf{y}_n)$.

We implement this by particle filtering with 1,000 particles. We initialize the particles to have the same starting score position (1st beat), and tempi assume a Gaussian distribution $\mathcal{N}(v_{\text{init}}, \sigma_v^2)$, where v_{init} is calculated as the average tempo of the MIDI score and σ_v is set to 40BPM. Now these particles represent the initial state distribution $p(x_0, v_0)$.

After seeing each audio frame, the particles will be updated according to Eq. (1) and (2). After this, the particles will represent the distribution $p(x_n, v_n|\mathbf{y}_1, \cdots, \mathbf{y}_n)$. Then the observation likelihood of each particle is calculated according to Eq. (3), (4) and (5). The particles are then resampled with replacement according to their likelihoods, which form a discrete distribution among the particles. After resampling, the new particles will represent the posterior probability $p(x_n, v_n|\mathbf{y}_1, \cdots, \mathbf{y}_n)$. Then the algorithm outputs their mean $\hat{x}_n$ and $\hat{v}_n$ as the estimate of current score position and tempo.

Audio frame length and hop are set as 46ms and 10ms, respectively. The minimum v_{min} and maximum tempo v_{max} are set to 30BPM and 300BPM, respectively.

Finally, as required by the output format of this task, we need to output the actual audio performance time of each note onset in the score. We calculate this value by reversing the audio frame to score position mapping.

4. RESULTS

The dataset used for evaluation has two groups. The first group are monophonic pieces or pieces of a monophonic melody with light accompaniment. There are in total 46 excerpts extracted from 4 distinct musical pieces in this group. The second group are polyphonic pieces. It consists of 10 pieces of four-part J.S. Bach chorales. The audio recordings were performed by a quartet of instruments: violin, clarinet, saxophone and bassoon. The audio recordings of both groups are in 44.1 KHz 16 bit wave format and the scores are in MIDI. The ground-truth alignment between audio and MIDI were generated by human annotation.

Evaluation measures are described in [13]. They are:

1. Missed Notes: notes in the reference file that are not recognized by the score follower. There are two cases: 1. these notes are not reported by the score follower; 2. the reported time of these notes are more than 2,000 ms away from their reference notes.
2. False Positive: notes of case 2 in Missed Notes.
3. Average Offset: average absolute-valued time offset between a reported note onset by the score follower and its real onset in the reference file.
4. Mean Offset: average sign-valued time offset.
5. Std Offset: standard deviation of sign-valued time offset.
6. Average Latency: Difference between detection time and the time the score follower sees the audio.

These measures above are calculated both globally (over the whole database) and locally (for each sound file). This way we will have two precision rates:

1. **Overall precision rate**: $(1 - \frac{\#\text{all missed notes}}{\#\text{all notes in the reference files}}) \times 100\%$
2. **Piecewise precision rate**: average for each piece of the value $(1 - \frac{\#\text{missed notes}}{\#\text{notes in the reference file}}) \times 100\%$.

The results are presented in Table 1. There are in total 5 score followers from 4 research groups. "DP1" is our system. It can be seen that in both measures, "SUROS1" gets the best result and its precision is significantly better than others. For "Total precision", our system ranks third, and is close to the second best system "AW1". For "Piecewise Precision", our system ranks second.

Figure 1 presents the histogram of piecewise precisions of each system. It is interesting that for each system, there are some pieces that can be perfectly followed. However, for the other pieces, the performances are quite uniform. This may indicate that once the score follower is lost, it is hard to catch up the performance again. From detailed results posted in the MIREX 2010 results webpage, we found that different systems are good at different pieces, and there seemed no particular piece that are easy or difficult to all systems. This may indicate that the content of the dataset biases the results. To get more meaningful results, we may need a larger dataset with a balanced coverage of different kinds of music pieces.

(%)	AW1	DP1	RVCC3	RVCC4	SUROS1
Total Precision	50.84	49.11	32.17	32.44	73.97
Piecewise Precision	50.33	67.14	62.79	64.50	73.93

Table 1. Evaluation results

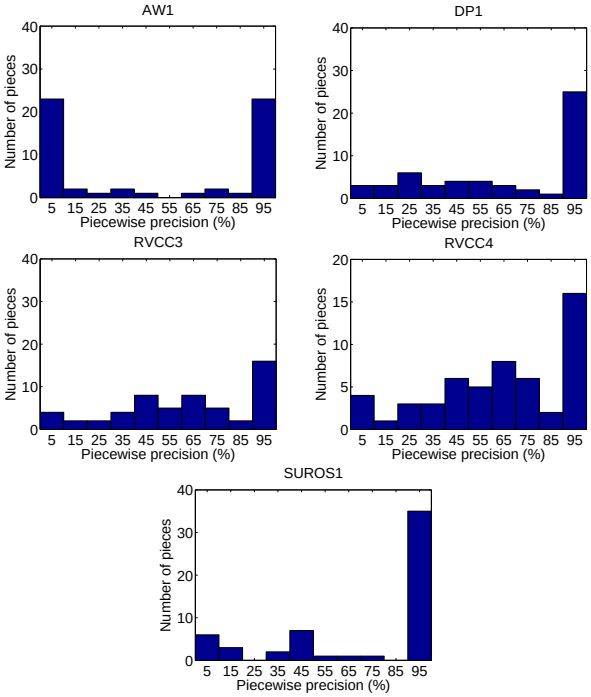

Figure 1. Histograms of piecewise precisions.

5. CONCLUSION

In this extended abstract, we describe our real-time score following algorithm for polyphonic audio performance. This algorithm is based on a hidden Markov process model, where the process model is defined by two dynamic equations and the observation model is the multi-pitch observation likelihood. Particle filtering is used to infer the current score position given current and past observations. Our system ranks third for overall precision and second for piecewise precision. However, detailed results suggest that a more meaningful result will need a larger and more balanced dataset.

6. ACKNOWLEDGEMENT

We would like to thank IMIRSEL. Without their hard work in organization, the score following task and all other tasks in MIREX would be impossible! Our research described in this extended abstract is supported by National Science Foundation grant IIS-0643752.

7. REFERENCES

[1] J. J. Bloch and R. B. Dannenberg: "Real-time computer accompaniment of keyboard performances," *Proc. International Computer Music Conference (ICMC)*, pp. 279–289, 1985.

[2] R. B. Dannenberg: "An on-line algorithm for real-time accompaniment," *Proc. International Computer Music Conference (ICMC)*, pp. 193–198, 1984.

[3] R. B. Dannenberg and B. Mont-Reynaud: "Following an improvisation in real time," *Proc. International Computer Music Conference (ICMC)*, pp. 241-248, 1987.

[4] S. Dixon: "Live tracking of musical performances using on-line time warping," *Proc. of the 8th International Conference on Digital Audio Effects (DAFx'05)*, 2005.

[5] L. Grubb and R. B. Dannenberg: "A stochastic method of tracking a vocal performer," *Proc. International Computer Music Conference (ICMC)*, 1997.

[6] N. Orio, S. Lemouton and D. Schwarz: "Score following: state of the art and new developments," *Proc. 2003 Conference on New Interfaces for Musical Expression (NIME)*, pp. 36–41, 2003.

[7] B. Pardo and W. Birmingham: "Modeling form for on-line following of musical performances," *Proc. Twentieth National Conference on Artificial Intelligence (AAAI)*, 2005.

[8] C. Raphael, "A Bayesian network for real-time musical accompaniment," *Proc. International Conference on Neural Information Processing Systems (NIPS)*, 2001.

[9] D. Schwarz, A. Cont and N. Orio: "Score following at IRCAM," *Proc. International Conference on Music Information Retrieval (ISMIR)*, 2006.

[10] A. Cont: "A coupled duration-focused architecture for real-time music-to-score alignment," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 974–987, 2010, 2010.

[11] Z. Duan, J. Han and B. Pardo: "Harmonically informed multi-pitch tracking," *Proc. ISMIR*, 2009.

[12] Z. Duan, B. Pardo and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio Speech Language Proc.*, in press.

[13] A. Cont, D. Schwarz, N. Schnell and C. Raphael, "Evaluation of real-time audio-to-score alignment," *Proc. International Conference on Music Information Retrieval (ISMIR)*, 2007.