Score Following by Particle Filtering for Music Robots

Takuma Otsuka[†] Kazuhiro Nakadai[‡] Toru Takahashi[†]
Kazunori Komatani[†] Tetsuya Ogata[†] Hiroshi G. Okuno[†]
†Graduate School of Informatics, Kyoto University ‡Honda Research Institute Japan, Co., Ltd.

1. Introduction

As an increasing number of robots are expected to get more involved in our human society, the means of interaction between humans and robots, such as verbal dialogue, music, or eyecontacts, have great importance. Music is especially a promising medium among many kinds of interaction. This is because music has been an essential and common factor in most human cultures beyond the region and race. Even people who do not share a language can share a friendly and joyful time through music although natural communications by other means are difficult. Therefore, *music robots* that is capable of, for example, dancing, singing or playing an instrument with humans will play an important role in the symbiosis between robots and humans.

We believe that robots must listen to the music in an acoustic manner like humans do without using symbolic data such as MIDI signals for natural interaction. To achieve such a capability, music robots at least require the ability to synchronize with the music in addition to presenting musical expressions. Previously, the music robots that listens to the music synchronizes with the music using only beats in the music [1] [2]. This tends to make the robot's musical expression repetitive such as drumming or stepping.

If music robots synchronize with the music using melodies on top of musical beats, robots become capable of singing a song with an accompaniment by a human musician, for instance. This paper presents a score following algorithm that enables robots to understand what part of the music is being performed. Figure 1 outlines our score following method. This algorithm acquires an acoustic signal incrementally and identifies the current score position.

2. Score Following for Music Robots

Music robots have to not only *follow* the music but also *predict* coming musical notes. This is because there exists some temporal overhead when music robots present a musical expression. For example, it takes the robot some time to make a step, or Murata *et. al.* [2] reports that it takes around 200 (ms) to generate singing voice using singing voice synthesizer VOCALOID [4]. Therefore, the mechanism for the prediction of future musical events are necessary.

To predict the future musical notes, the current score position as well as tempo (the speed of the music) should be estimated. With these two types of information, the future score position can be derived by using extrapolation or probabilistic model presented in Sec 3.. Here, the problem is specified as follows:

Input: incremental audio signal and the corresponding musical score,
Output: predicted score position,
Assumption: the tempo is unknown; only pairs of pitch and length (e.g., quarter note) are given.

2.1 Existing Score Following Methods

Many of existing score following methods are based on Hidden Markov Model (HMM) [5]. Score following problem is naturally formulated using state space model which is compatible with HMM. Typically, each musical note is modeled as hidden states. However, HMM provides no prediction scheme because their purpose is automatic accompaniment, where no temporal overhead exists.

2.2 Predictive Score Following using Particle Filter

We model this simultaneous estimation as a state space model and obtain the solution with a particle filter. The particle filter approximates the probability distribution of score positoin and tempo by the density of many particles with state transition model and observation model. With incremental audio input, the particle filter updates the distribution and the estimation is carried out.

3. Algorithm

3.1 Overview of Particle Filter

Let $X_{f,t}$ be the amplitude of input audio signal in time frequency domain with frequency bin f and time t and k be score frame. The score is divided into frames such that the length of quater note is 12. Musical notes $\mathbf{n}_k = [n_k^1 ... n_k^{r_k}]^T$ are placed at frame k and r_k is the number of musical notes. Each particle p_i has the value of score position, beat interval and weight: $p_i = (\hat{k}_i, \hat{b}_i, w_i)$, and N is the number of particles. Although the actual score position k is discrete, the value held by particles $\hat{k}_i$ is continuous.

Every ΔT time, the following procedures are carried out: (1) observation, (2) resampling, (3) state transition (prediction). Each procedure is as follows.

3.2 Observation Model and Weight Calculation

At time t, spectrogram $X_{f,\tau}, t - L < \tau \le t$ is used for the weight calculation. L denotes the window length of the spectrogram. The weight of each particle $w_i, 1 \le i \le N$ is a product of three weights as follows:

$$w_i \quad = \quad w_i^{ch} \times w_i^{sp} \times w_i^{t}. \tag{1}$$

The two weights, chroma vector weight w_i^{ch} and spectrogram weight w_i^{sp}, are measures of pitch information.

$$w_i^{ch} \quad = \quad \left(\sum_{\tau} \mathbf{c}_\tau^a \cdot \mathbf{c}_{k_\tau^i}^s \right) / L, \tag{2}$$

$$w_i^{sp} \quad = \quad (1 + Q) \exp(-Q), \tag{3}$$

$$Q \quad = \quad \frac{1}{L} \sum_f X_\tau \log \frac{Xf,\tau}{\hat{X}_{f,k_\tau^i}}, \tag{4}$$

$$\hat{X}_{f,k_\tau^i} \quad = \quad \sum_{r=1}^{r_{k\tau^i}} \sum_{g=1}^{G} h(g) N(f; gF_{n_{k\tau^i}^r}, \sigma^2). \tag{5}$$

Note that the particle p_i's score index k_τ^i corresponding to time τ can be derived using estimate score position $\hat{k}_i$ and tempo (beat interval) $\hat{b}_i$. $\mathbf{c}^a$ and $\mathbf{c}^s$ is chroma vector derived from the audio signal and the score, respectively. This vector has 12 elements that correspond to the power of $C, C\sharp, ..., B$ chromatic notes. This is calculated with band-pass filters [3] from audio signal. The value of each element in the score chroma vector is 1 when the score has

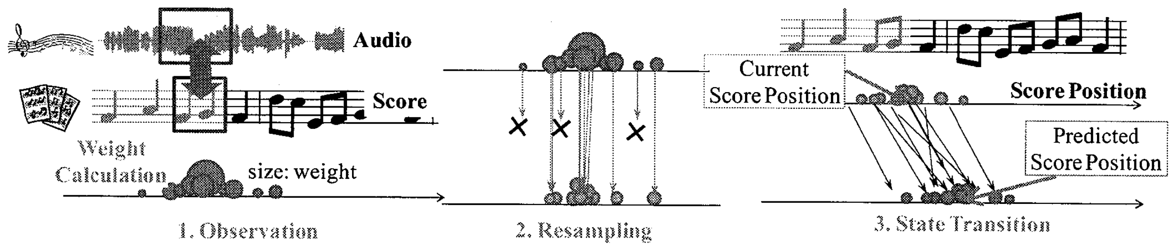

Figure 1: Overview of Particle Filter based Score Following

the corresponding note, and otherwise 0. For spectrogram weight calculation, the spectrum is generated from the musical score using the harmonic gaussian mixture model in Eq (5). g is the harmonic index, G is the number of harmonics, and $h(g) = 0.01^g$ is the height of each harmonics. $F_{n^r_{k\tau^i}}$ is the fundamental frequency of note $n^r_{k\tau^i}$.

The weight w^t_i is the measure of beat interval and obtained through normalized cross correlation of the spectrogram with $\hat{b}_i$ shift:

$$w^t_i = \frac{\sum_{f,\tau} X_{f,\tau} X_{f,\tau-\lfloor \hat{b}_i+0.5 \rfloor}}{\sqrt{\sum_{f,\tau} X^2_{f,\tau} \sum_{f,\tau} X^2_{f,\tau-\lfloor \hat{b}_i+0.5 \rfloor}}}, \qquad (6)$$

where $\lfloor x \rfloor$ is the floor function.

3.3 Resampling Based on the Weights

After calculating the weight of all particles, particles are resampled. In this procedure, particles with a large weight are selected many times, whereas those with little weight are discarded because their score position is unreliable. A particle p is drawn independently N times from the following distribution:

$$P(p = p_i) = \frac{w_i}{\sum_{i=1}^{N} w_i} \qquad (7)$$

3.4 State Transition Model

The future score position is predicted by updating particles with the following state transition model:

$$\hat{k}_i \leftarrow \hat{k}_i + \Delta T / \hat{b}_i + u, \qquad (8)$$

$$\hat{b}_i \leftarrow \hat{b}_i + v, \qquad (9)$$

where u and v are gaussian random variables with means 0 and variances σ^2_u, σ^2_v, respectively. The prediction of score position ΔT ahead is calculated by taking the mean value of densely distributed particles. Further prediction can be obtained by applying Eq. (8, 9) enough times, or simply extrapolating the score position with current tempo.

3.5 Initial Probability Distribution

Initial particles are set as follows: (1) draw N samples of beat interval $\hat{b}_i$ value from the uniform distribution ranging from 60 (bpm; beats per minute) to 150 (bpm). (2) the score position of each particle is set $\Delta T / \hat{b}_i$.

4. Experimental Evaluation

Our system is implemented on MacOSX with Intel Core2 Duo. The error of musical note prediction and the real time factor with various numbers of particles are evaluated. The error of prediction is defined as the difference of reported prediction time and the ground truth. The error values are averaged for each song. 10 jazz songs from RWC music database [6] are used for the experiment. Parameters are empirically set as follows: $L = 3$ (sec), $\Delta T = 1$ (sec), $G = 10$, $\sigma^2_u = 0.05$ (sec^2), $\sigma^2_v = 0.1$ (sec^2). The sampling rate is 44.1 (kHz) and Fourier transform is executed with 2048 (pt) window length and 441(pt) window shift.

Table 1: Particle number N vs. mean prediction error (sec)

N particles	50	100	500	1000
Average error	4.8	3.3	2.1	1.8
Real time factor	0.33	0.58	2.88	5.64

4.1 Results

Table 1 shows the results. These results confirm that the error decreases when we use more particles. Some error still exists mainly because our model is ignorant of note onset information. When we add this information to the observation model, the error is expected to reduce.

To run our method in real-time, the observation, resampling, and state transition of all particles must be executed within ΔT. However, the computational cost increases in proportion to the number of particles, and exceeds ΔT when $N \approx 200$. This problem will be alleiviated when this algorithm is implemented on a processor that has more than 2 cores. This approach will be effective because the observation and state transition is calculated independently with respect to each particle.

5. Conclusion

Our goal is to achieve a human-robot interaction through music. This paper presented score following algorithm using particle filter that enables robots to synchronize with the music. Our future works include: (1) real time implementation, (2) feasibility test of our algorithm on an actual robot, (3) model refinement to reduce the prediction error. In applying our method to an actual robot, the supprresion of self-generated sound from the robot will be a major issue, such as motor noise.

References

[1] G. Weinberg *et al.* : “Toward Robotic Musicianship”, *Computer Music Journal* 30(4):28–45, 2006.

[2] K. Murata *et al.*: “A Robot Uses Its Own Microphone to Synchronize Its Steps to Musical Beats While Scatting and Singing”, *Proc. of IROS*, 2008, pp.2459–2464.

[3] M. Goto, “A Chorus Section Detection Method for Musical Audio Signals and Its Application to a Music Listening Station’, *IEEE Trans. on Audio, Speech and Language Processing* 14(5):1783–1794, 2006.

[4] H. Kenmochi *et al.*, “Vocaloid – commercial singing synthesizer based on sample concatenation”, *Proc. of INTERSPEECH*, 2007 pp4010–4011.

[5] R. Dannenberg *et al.*, “Music Score Alignment and Computer Accompaniment”, *Communications of the ACM* 49(8):38–43, 2006.

[6] M. Goto *et al.*, “RWC Music Database: Popular Music Database and Royalty-Free Music Database”, *IPSJ Sig Notes* 2001(103):38–43, 2001.