2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Particle-filter Based Audio-visual Beat-tracking for Music Robot Ensemble with Human Guitarist

Tatsuhiko Itohara, Takuma Otsuka, Takeshi Mizumoto, Tetsuya Ogata, and Hiroshi G. Okuno

***Abstract*— This paper presents an audio-visual beat-tracking method for ensemble robots with a human guitarist. Beat-tracking, or estimation of tempo and beat times of music, is critical to the high quality of musical ensemble performance. Since a human plays the guitar in out-beat in back beat and syncopation, the main problems of beat-tracking of a human's guitar playing are twofold: tempo changes and varying note lengths. Most conventional methods have not addressed human's guitar playing. Therefore, they lack the adaptation of either of the problems. To solve the problems simultaneously, our method uses not only audio but visual features. We extract audio features with Spectro-Temporal Pattern Matching (STPM) and visual features with optical flow, mean shift and Hough transform. Our beat-tracking estimates tempo and beat time using a particle filter; both acoustic feature of guitar sounds and visual features of arm motions are represented as particles. The particle is determined based on prior distribution of audio and visual features, respectively Experimental results confirm that our integrated audio-visual approach is robust against tempo changes and varying note lengths. In addition, they also show that estimation convergence rate depends only a little on the number of particles. The real-time factor is 0.88 when the number of particles is 200, and this shows out method works in real-time.**

I. INTRODUCTION

Ensemble music robots are expected to improve the symbiosis between people and robots. Music can be shared with everybody because it has few barriers such as language. However, the music robots reported thus far, e.g., robots that play instruments or dance to music, only allow us to enjoy performances in a passive way. In contrast, musical co-player robots enable us to appreciate music more actively because we can enjoy singing, playing instruments, or dancing with robots.

Music co-player robots are required to estimate tempo and predict beat-timings the partner's play to achieve an ensemble. This function, called beat-tracking, is used to estimate the timing of quarter note beats and the tempo. For example, Weinberg *et al.* reported a percussionist robot that imitates co-player's playing and plays according to a co-player's timing [1]. Mizumoto *et al.* reported a thereminist robot that performs with a co-player's flutist and a human percussionist [2]. This robot adapts to the changing tempo of the human's play, such as *accelerando* and *fermata*.

We focus on beat-tracking for guitar playing. The guitar is one of the most popular instruments used for basic ensembles consisting of a melody and a backing part. Therefore, developing music robots that can play with a guitarist will increase the robot's opportunities to play with humans in an ensemble.

Tatsuhiko Itohara, Takuma Otsuka, Takeshi Mizumoto, Tetsuya Ogata, and Hiroshi G. Okuno are with Graduate School of Informatics, Kyoto University, Sakyo, Kyoto 606-8501, Japan {itohara, ohtsuka, mizumoto, tall, ogata, okuno} @kuis.kyoto-u.ac.jp

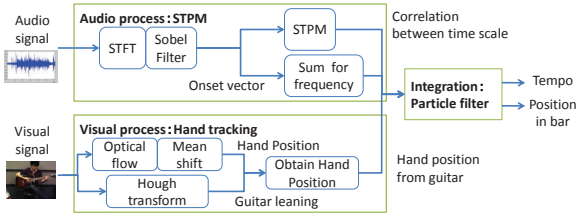

Fig. 1. Architecture underlyning our beat-tracking technique

Two issues should be solved for the guitar beat tracking: (1) fluctuating tempo and note length, and (2) the fewer number of beats. For the first issue, the fluctuation derives from human's natural expression in music. The second issue is due to a relatively low number of beats compared to popular music.

This paper presents a particle-filter-based audio-visual beat-tracking method for guitar. Fig. 1 shows the architecture of our method. The key idea is integration of the audio and visual information because other beat-trackings use audio information only and guitar motions have a strong correlation with beat timings. Our method employs a particle filter that incrementally estimates the tempo and beat time; particles have a tempo and beat time as their state, and evaluate the state value using audio and visual features. The audio features are the normalized cross correlation and increments of the audio signal using Spectro-Temporal Pattern Matching (STPM) [3], and the visual features are the relative hand motion using hand-motion tracking by optical flow and mean shift [4], and localization of the neck of the guitar by Hough transform [5].

Section II discuss the problems with guitar beat-tracking, and Section III introduces our approach to audio-visual beat-tracking. Section IV shows that the experimental results demonstrate the superiority of our beat-tracking to Murata's method, addressing the problems of robot ensemble, in tempo and note length changes and real-time performance. Section V shows our concludes of the methods.

II. ASSUMPTION AND PROBLEM

A. Definition of the ensemble with guitar

Our ensemble consists of a robot and a human guitarist. At the beginning of an ensemble, we take some *counts* to synchronize with a co-player robot as humans do. These counts are usually given by voices, gestures, and hit sounds from the guitar. Our method estimates the beat timings

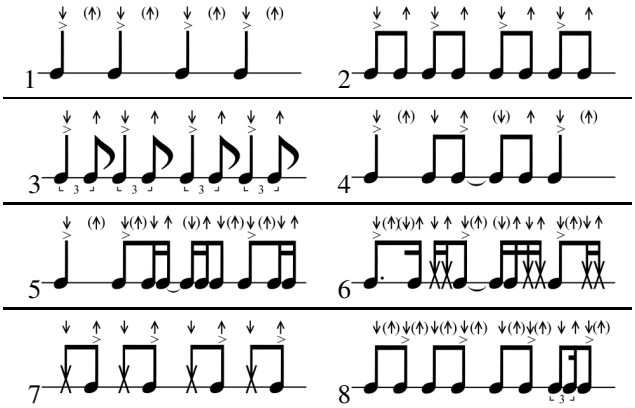

Fig. 2. Typical guitar beat patternsD Note × represents guitar-cutting, a percussive sound made with quick muting sounds. The > denotes accented, ↑ and ↓ denote the directions of strokes, and (↑) and (↓) denote air strokes.

without prior knowledge of the musical score a human plays from. This is because 1) there are many guitar scores without beat patterns, that is, only melody and chord names are written, and 2) our main goal is improvisational session.

Now, we will state some assumptions we make for our ensemble. Guitar playing is mainly categorized into two classes: stroke and arpeggio. We choose stroke style because the stroke motion has a strong correlation with the beat time and interval of the performance. Therefore, we can make the best use of the visual features of guitar playing. Let our ensemble have quadruple rhythm. Then, we determine the number of counts as four and considered the tempo of the ensemble can be only altered moderately from the tempo implied by counts. Note that our beat-tracking can accept other rhythms by adjusting the hand trajectory model we introduce in Section III-B.3.

Stroke motion has an implicit rule for air strokes, that is, a stroke at soundless beat-times, to keep tempo changes due to human error small. This can be confirmed from the score of Pattern 4 in Fig. 2. Strokes at the beginning of each bar go downward, and the cycle of a stroke usually lasts the length of a quarter note (eight beats) or of an eighth notes (sixteen beats). We assume eight-beat music and model hand trajectory to estimate the hand position.

We decide not to use previous knowledge on the color of hands in our visual-tracking; humans have various hand colors. The stroking hand, on the other hand, makes the largest movement in the body of a playing guitarist. We can confirm this by the way of stroking.

Conditions and assumptions for ensemble

Conditions:
(1) Stroke (guitar-playing style)
(2) Take counts at the beginning of the performance
(3) Unknown guitar-beat patterns
(4) With no previous knowledge of hand colors

Assumptions:
(1) Quadruple rhythmCeight to the bar
(2) Not much variance from the tempo implied by counts
(3) Stroking hand creates the largest movement in the body of a guitarist

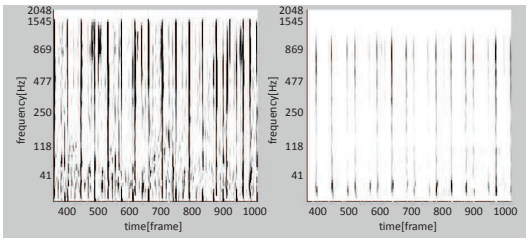

Fig. 3. These two graphs show the strength of onsets in each bin with the power spectrogram of music after Sobel filtering. At the left is j-pop music (BPM 120), and at the right is a guitar backing performance (BPM 110). The values represent by darkness is 64-dimension mel-scaled spectrograms of music whose negative values are set to zero. The horizontal axis, the vertical axis, and the concentration of darkness correspond to the time frame, frequency, and strength of onset. Here, a frame is equivalent to 0.0116 sec.

B. Beat-tracking Conditions

Our method estimates the beat-time and tempo from audio and visual features. We use a microphone embedded in the robot's head for the audio input signal. Our beat-tracking method estimates the tempo and "bar-position", where the performer is playing bars each time. Finally we obtain beat-times from the bar-positions. We give the conditions for our beat-tracking in the box below.

Input-output

Input:
(1) Guitar sounds captured with robot's microphone
(2) Images of guitarist captured with robot's camera

Output:
(1) Bar-position
(2) Tempo

C. Issues of Our Beat-tracking

Guitar beat-tracking has two issues with varying note length and fewer onsets. The first issue is caused by the complexity of beat patterns. We give eight typical beat patterns in Fig. 2. Patterns 1 and 2 are often used in popular music. Pattern 3 contains triplet notes. Whereas all of the accented notes in these three patterns are down beats, the other patterns contain accented up beats. Moreover, all of the accented notes of Patterns 7 and 8 are up beats. For all these reasons, we have to consider how to estimate the tempos and bar-positions of the beat patterns with accented up beats.

We can confirm the sparseness of onsets in guitar performance from Fig. 3. This sparseness is mainly caused by the fewer instruments used in the music. In contrast, as popular music has some musical parts whose accented beats are mainly down beats as in drum parts, such music has advantages in beat-tracking. In addition, the audio signals in our beat-tracking have two issues with fluid tempo caused by humans playing instruments and mixed noise such as that from the fan in robots.

We have two issues in the visual hand tracking; the false recognition of the moving hand and the compensation of the low time resolution compared to the audio signal. A naive application of color histogram-based hand trackers is vulnerable to false detections caused by the varying luminance of

the skin color and capture other nearly skin colored objects. While optical flow-based methods are considered suitable to the hand-tracking, we have difficulty in employing this approach because flow vectors include some noises from the movements of other parts of the body. The temporal resolution of a visual signal is about one-quarter compared to an audio signal. Therefore, we have to synchronize these two signals to integrate them.

Issues

Audio signal:
(1) Complexity of beat patterns
(2) Sparseness of onsets
(3) Fluidity of human playing tempos
(4) Anti-noise signal

Visual signal:
(1) Distinguishing hand from other parts of body
(2) Low visual resolution

D. Related Research and Solution of the Problems

1) Beat-tracking: Beat-tracking has been extensively studied in music processing. Goto uses agents that independently extract the inter-onset intervals of music and estimate tempos [6]. Whiteley's method is based on statistical methods like a particle filter [7]. They are robust against varying note lengths but vulnerable to tempo changes because their target music consists of complex beat patterns and stable tempo. Moreover, they are not intended to adopt robots and therefore they are not robust against robot noise.

In contrast, some beat-tracking methods address the noise problem. Sethares makes "rhythm tracks", representing the rhythmic structure by preprocessing the audio and by modeling the audio including noise [8]. Murata's method, using STPM, is also robust against robot noise. While this STPM-based method is designed to adapt to sudden tempo changes, the method is likely to mistake up-beats for down-beats. This is partly because the method fails to estimate the correct note lengths and partly because no distinctions can be made between the down and up beats with its beat detecting rule. In order to robustly track the human's performance, Otsuka et al. use a musical score: They have reported an audio-to-score alignment based on a particle filter and revealed its effectiveness to the tempo changes [9].

2) Visual-tracking: We introduce two methods of visual-tracking, those based on optical flow and those based on color information. With optical flow methods, we can detect the displacement of pixels between frames. There have been some studies applying it to an ensemble with a music robot. For example, Pan *et al.* use it to extract a queue of exchanged initiatives for their ensemble [10].

With color information, we can compute the prior probabilistic distribution for tracking objects. This enables a robot to track the instrument of its partner, e.g., with a method based on a particle filter [11]. There have been many other methods of extracting the positions of instruments. Lim *et al.* [12] use a Hough transform to extract the angle of a flute. Pan *et al.* [10] use a mean shift to estimate the position of the mallet's endpoint. These detected features are used to queue their performances. In Section III-B.2, we give a detailed explanation of the Hough transform and mean shift.

TABLE I
COMPARISON OF BEAT-TRACKING METHODS.

	Following Tempo Change	Following Note Length	Anti noise	Remarks
Hierarchical Beat-tracking [6]	Medium	Superior	Don't care	Detecting chord change
Murata [3]	Superior	Poor[1]	Superior	STPM + rule-based
Whiteley [7]	Superior	Superior	Don't care	Prior probability of rhythm pattern + MIDI feature
Otsuka [9]	Superior	Superior	Superior	Score alignment
Our method	Superior	Superior	Superior	Visual information + nonuse of score

3) Integration: Integrating the results of elemental methods is a filtering problem, where observations are features extracted with methods and latent states are the results of estimates with integration. The Kalman filter produces estimates of latent state variables with linear relationships between them and with a Gaussian distribution. The Extended Kalman Filter adjusts the state relationships of representations of nonlinear but differentiable functions. These methods are however unsuitable for the beat-tracking we propose because of the highly nonlinear model of the hand trajectory of guitarists.

Particle filters, on the other hand, which are also known as Sequential Monte Carlo methods, estimate the state space of latent variables with highly nonlinear relationships and a non-Gaussian distribution. At frame t, z_t and x_t denote the variables of the observation and latent states. The probability density function (PDF) of latent state variables $p(x_t|z_{1:t-1})$ is approximated as:

$$p(x_t|z_{1:t-1}) \approx \sum_{i=1}^{I} w_n^{(i)} \delta(x_t - x_t^{(i)}), \quad (1)$$

where the sum of weights $w_n^{(i)}$ is 1. Here, I is the number of particles and $w_t^{(i)}$ and $x_t^{(i)}$ correspond to the weight and state variables of the i-th particle. The $\delta(x_t - x_t^{(i)})$ is the Dirac delta function. This method is used for beat-tracking [7], [9] and visual-tracking [11]. Moreover, Nickel *et al.* [13] apply it as a method of audio-visual integration for the 3D identification of a talker.

4) Solution of the Problems: Now, we will discuss the approaches to solving these problems. A guitarist usually strokes downward at down beats and upward at up beats. We make the best of these characteristics along with the STPM beat-tracking: Our beat-tracking recognizes complex beat patterns under the condition of sparse onsets. Our visual beat-tracking consists of (1) an optical flow method to roughly distinguish between the most salient hand motions and other small movements of other parts of the body, and (2) a mean shift with the color space robust against varying luminance to detect a precise position. Here, we show the comparison of beat-tracking methods in Table I.

[1]On mono-instrument music. On poly-instrument music (such as pop), this method should obtain better results.

III. AUDIO-VISUAL BEAT-TRACKING

We regard the problem with our beat-tracking as the sequential estimation of PDF of the state variables from observations, where the states are tempos and bar-positions, and the observations are the temporal sequences of audio and visual features. We solve this problem with a particle filter.

A. Audio Feature Extraction with STPM

We introduce the STPM for calculating the audio features, that is, inter-frame correlation $R_t(k)$ and the normalized summation of onsets F_t, where t is the frame index. Spectra are consecutively obtained by applying a short time Fourier transform (STFT) to an input signal sampled at 44.1 kHz. A Hamming window of 4096 points is used as a window function and its shift size is 512 points. We send the spectra to a mel-scaled filter bank to reduce the number of frequency bins from 2,049 linear frequency bins to 64 mel-scaled frequency bins. We apply a Sobel filter to the spectra to enhance its edges and set the negative values of its result to zero. Let $d(t,f)$ be the *onset vector*, the result of the above, at the t-th time frame and f-th mel-filter bank bin. Inter-frame correlation with k frames back $R_t(k)$ is calculated by the normalized cross-correlation (NCC) of onset vectors defined in Eq. (2). This is the result for STPM. In addition, we define F_t as the sum of the values of the onset vector at the t-th time frame in Eq. (3) and then we detect the peak time of onsets.

$$R(t,k) = \frac{\sum_{j=1}^{F_\omega}\sum_{i=0}^{P_\omega-1} d(t-i,j)d(t-k-i,j)}{\sqrt{\sum_{j=1}^{F_\omega}\sum_{i=0}^{P_\omega-1} d(t-i,j)^2 \sum_{j=1}^{F_\omega}\sum_{i=0}^{P_\omega-1} d(t-k-i,j)^2}}, \quad (2)$$

$$F(t) = \log\left(\sum_{f=1}^{F_\omega} d(t,f)\right) / peak, \quad (3)$$

where *peak* is a variable for normalization and is updated under the local peak of onsets. The F_ω denotes the number of dimensions of onset vectors used in NCC and P_ω denotes the frame size of pattern matching. We set these parameters to 62 dimensions and 87 frames (equivalent to 1 sec.) according to Murata *et al.* [3].

B. Visual Feature Extraction with Hand-Tracking

We extract the visual features, that is, the temporal sequences of hand positions with these three methods.

Hand-tracking
1) Estimation of area where hand is with optical flow
2) Estimation of hand position with mean shift
3) Complementary sequential hand position with hand-tracking

1) Estimation of area where hand is with optical flow: The assumption described in Section II-A assures us that the hand trajectory can be extracted with the optical flow. We use the Lucas-Kanade (LK) method [14] to avoid the large cost of calculating flow vectors with all pixels of a frame. Although flow vectors include some noises, e.g., motions irrelevant to the hand, the medians of length and angle of the vectors accord with those of the vector of the hand. Then we define the center of the hand candidate area as the coordinate of a flow vector whose length and angle are nearest from the middle values.

Fig. 4. Position of hand.

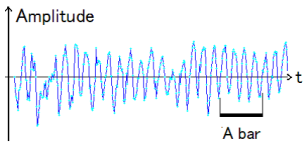

Fig. 5. Example of sequential hand positions.

2) Estimation of hand position with mean shift: We apply mean shift to visual signals to extract the precise hand position. The box below has the algorithm for mean shift. Mean shift detects the maxima of a density function given discrete data sampled from that function Let $(h_{x,t}, h_{y,t})$ be *hand coordination*, that is, the coordination of the hand in a frame, or the result of mean shift. Mean shift has good features in that it involves low computational cost and it is robust against outliers far from the centroid. The default window is the result in Section III-B.1. The kernel is the hue histogram calculated with the color space of Miyazaki *et al.* [15]. The main feature of this color space is the highly accurate detection of hand color caused by its robustness against shadows and specular reflections. In addition, as we only use *hue*, the calculation of weight is independent of *intensity*.

3) Complementary sequential hand position with hand-tracking: We model the hand trajectory with hand coordination $(h_{x,t}, h_{y,t})$ to complement visual features at low resolution. A guitarist usually moves his hand near the neck of his guitar. We define a hand position at t time frame r_t as the relative distance between the hand and the neck as

$$r_t = \rho_t - h_{x,t}\cos\theta_t + h_{y,t}\sin\theta_t, \quad (4)$$

where ρ_t and θ_t are the parameters of the line of the neck computed with the Hough transform. In the Hough transform, we compute 100 candidate lines, detect and remove outliers with RANSAC [16], and get the average of Hough parameters. Positive values denote a hand position that is above from the guitar. Similarly, negative values denote it is below. Fig. 4 has a photograph of the hand position. Fig. 5 shows an example of the sequential hand positions. The horizontal axis and the vertical axis correspond to the time frame and hand position.

Now, let ω_t and θ_t be a beat-interval and bar-position at the t-th time frame, where a bar is modeled as a circle, $0 \le \theta_t < 2\pi$ and ω_t is inversely proportional to the angle rate, that is, tempo. With assumption 3 in Section II-A, we presume that down strokes are at $\theta_t = \frac{n\pi}{2}$ and up strokes are at $\theta_t = \frac{n\pi}{2} + \frac{\pi}{4} (n = 0,1,2,3)$. In other words, zero crossover points of hand position are at these θ. In addition, as stroking is a smooth motion to keep the tempo stable, we assume the sequential hand position can be represented with a continuous function. Then, hand position r_t is defined by

$$r_t = -a\sin(4\theta_t), \quad (5)$$

where a is a constant value of hand amplitude.

TABLE II

CATALOG OF PARAMETERS FOR OUR PARTICAL FILTER.

Indices	i	:	Particle index
	t	:	Filter step
Observation variables	$R_t(k)$	:	Inter-frame correlation with k frames back
	F_t	:	Normalized onset summation
	r_t	:	Hand position
State variables	$\theta_t^{(i)}$	:	Bar-position
	$\omega_t^{(i)}$	:	Beat-interval

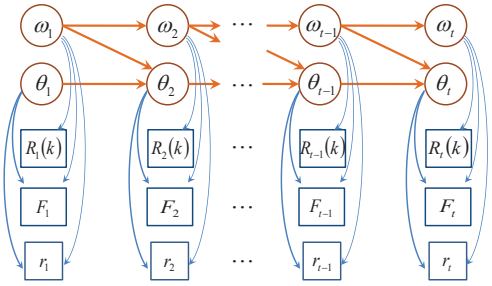

Fig. 6. Graphical ModelD◯ denotes state and □ denotes observation variables.

C. Particle Filter-based Integration

1) Overview of particle filter: Table II lists a catalog of the parameters in our particle filter method. The θ_t and ω_t denote the bar-position and beat-interval as state variables. The $R_t(k)CF_t$, and r_t denote inter-frame correlation with k frames back, normalized onset summation, and hand position as observation variables. The $\theta_t^{(i)}$ and $\omega_t^{(i)}$ are parameters of the i-th particle. We will explain the estimation process with the particle filter here.

2) State transition with sampling: The state variables at the t-th time frame $[\theta_t^{(i)}\omega_t^{(i)}]$ are sampled with Eqs. (6) and (7) with the observations at the $t-1$-th time frame. The graphical model is outlined in Fig. 6.

$$\begin{aligned}\omega_t^{(i)} &\sim q(\omega_t|\omega_{t-1}^{(i)}, R_t(\omega_t), \omega_{init}) \\ &\propto R_t(\omega_t) \times \mathrm{Gauss}(\omega_t|\omega_{t-1}^{(i)}, \sigma_{\omega_q}) \times \mathrm{Gauss}(\omega_t|\omega_{init}, \sigma_{\omega_q})\end{aligned} \tag{6}$$

$$\begin{aligned}\theta_t^{(i)} &\sim q(\theta_t|r_t, F_t, \omega_{t-1}^{(i)}, \theta_{t-1}^{(i)}) \\ &= \mathrm{Gauss}\left(\theta_t|\hat{\Theta}_t^{(i)}, \sigma_{\theta_q}\right) \times \mathrm{penalty}(\theta_t^{(i)}|r_t, F_t)\end{aligned} \tag{7}$$

Here, $\mathrm{Gauss}(x|\mu,\sigma)$ represents the PDF of a Gaussian distribution where x is a variable and parameters μ and σ correspond to the mean and standard deviation. The σ_{ω_q} denotes the standard deviation for the sampling of the beat-interval and σ_{θ_q} denotes that for the bar-position. The ω_{init} denotes the beat-interval estimated and fixed with the counts, first four beats. The $\hat{\Theta}_t^{(i)}$ is a predicted value of $\theta_t^{(i)}$ and is defined by

$$\hat{\Theta}_t^{(i)} = \theta_{t-1}^{(i)} + \frac{b}{\omega_{t-1}^{(i)}}, \tag{8}$$

where b is a proportional constant for the transform of a beat-interval into an angle rate of the bar-position.

We will now discuss Eqs. (6) and (7). In Eq. (6), the first term $R_t(k)$ is multiplied with two window functions of different means. The first is calculated from the previous frame and the second is from the counts. In Eq. (7), $\mathrm{penalty}(\theta|r,F)$ is the result of five multiplied window functions. These functions are the summation of normal distributions of variable θ on satisfying of the conditions each function has, otherwise they are 1 in any θ. This *penalty* function pulls the peak of the θ distribution into its own peak and modifies the distribution to match it with the assumptions and the models. Fig. 7 shows the change in the θ distribution by multiplying the *penalty* function. In the following, we present the conditions for each window function and the values of θ for the peaks of the function.

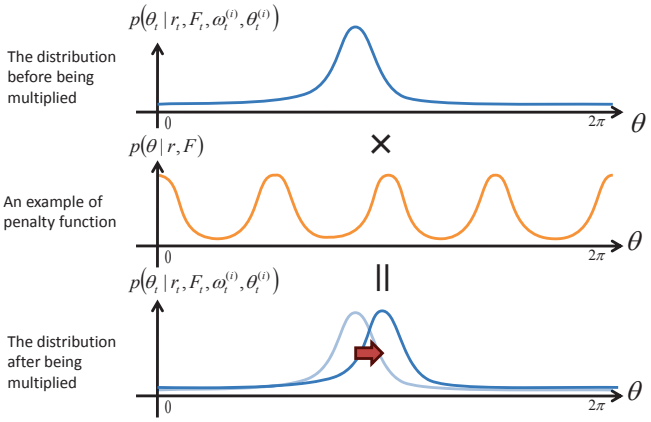

Fig. 7. Example of change in θ distribution while multiplying *penalty* function.

$$r_{t-1} > 0 \cap r_t < 0 \Rightarrow \frac{n\pi}{2}, \tag{9}$$

$$r_{t-1} < 0 \cap r_t > 0 \Rightarrow \frac{n\pi}{2} + \frac{\pi}{4}, \tag{10}$$

$$r_{t-1} > r_t \Rightarrow \frac{n\pi}{2}, \tag{11}$$

$$r_{t-1} < r_t \Rightarrow \frac{n\pi}{2} + \frac{\pi}{4}, \tag{12}$$

$$F_t > thresh. \Rightarrow \frac{m\pi}{4}. \tag{13}$$

Here, $n = 0,1,2,3$, $m = 0,1,\ldots,8$ and $thresh.$ is a threshold that determines whether F_t is constant noise or not. Eqs. (9) and (10) are based on the assumption of stroking directions. Eqs. (11) and (12) are based on the model of the hand trajectory. Eq. (13) is based on assumption (1) in Section II-A.

3) Weight calculation: The weights of particles at the t-th time frame are used to estimate the points of beat-intervals and bar-positions and are sequentially calculated with observations and state variables by

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(\omega_t^{(i)}, \theta_t^{(i)}|\omega_{t-1}^{(i)}, \theta_{t-1}^{(i)}) p(R_t(\omega_t^{(i)}), F_t, r_t|\omega_t^{(i)}, \theta_t^{(i)})}{q(\omega_t|\omega_{t-1}^{(i)}, R_t(\omega_t^{(i)}), \omega_{init}) q(\theta_t|r_t, F_t, \omega_{t-1}^{(i)}, \theta_{t-1}^{(i)})}. \tag{14}$$

The terms of the numerator in Eq. 14 are called a state transition model function and a observation model function. The more the values of a particle match each model, the larger value these function return. The denominator is called a proposal distribution. When a particle of low probability is sampled, its weight increases with the low value of the denominator.

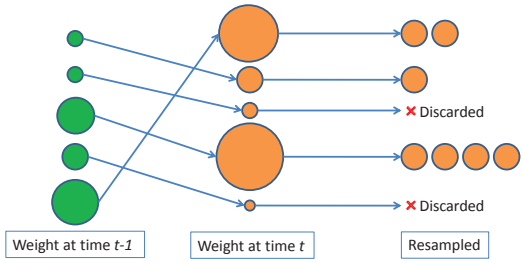

Fig. 8. Flow for resampling.

The two equations below give the derivation of the state transition model function.

$$\theta_t = \hat{\Theta}_t + n_\theta \tag{15}$$

$$\omega_t = \omega_{t-1} + n_\omega, \tag{16}$$

where n_ω and n_θ correspond to the noise of the beat-interval and bar-position distributed with a normal distribution. The state transition model function is expressed as the product of the PDF of these distributions.

Here, we give the deviation of the observation model function. The $R_t(\omega)$ and r_t are distributed according to the normal distributions where the means of the distributions are $\omega_t^{(i)} and -a\sin(4\hat{\Theta}_t^{(i)})$, respectively. The F_t is empirically approximated with the values of the observation as

$$\begin{aligned} F_t &\approx f(\theta_{beat_t}, \sigma_f) \\ &\equiv \mathrm{Gauss}(\theta_t^{(i)}; \theta_{beat,t}, \sigma_f) * rate. + bias.), \end{aligned} \tag{17}$$

where $\theta_{beat,t}$ is the bar-position of the nearest up or down beat from $\hat{\Theta}_t^{(i)}$, $rate.$ is a constant value for the maximum of approximated F_t being 1, and $bias.$ is uniformly distributed from 0.35 to 0.5. Then, the observation model function is expressed as the product of these three functions.

$$p(R_t(\omega_t)|\omega_t^{(i)}) = \mathrm{Gauss}(\omega_t; \omega_t^{(i)}, \sigma_\omega) \tag{18}$$

$$p(F_t|\omega_t^{(i)}, \theta_t^{(i)}) = \mathrm{Gauss}(F_t; f(\theta_{beat,t}, \sigma_f), \sigma_f) \tag{19}$$

$$p(r_t|\omega_t^{(i)}, \theta_t^{(i)}) = \mathrm{Gauss}(r_t; -a\sin(4\hat{\Theta}_t^{(i)}), \sigma_r) \tag{20}$$

4) State estimation and resampling: We finally estimate the state variables at the t-th time frame from the average with the weights of particles.

$$\overline{\omega_t} = \sum_{i=1}^{I} w_t^{(i)} \omega_t^{(i)} \tag{21}$$

$$\overline{\theta_t} = \arctan\left(\sum_{i=1}^{I} w_t^{(i)} \sin\theta_t^{(i)} \Big/ \sum_{i=1}^{I} w_t^{(i)} \cos\theta_t^{(i)} \right) \tag{22}$$

Resampling is used to avoid the situation that all but one of the weights are close to zero. In resampling, particles with large weights are selected many times, whereas those with small weights are discarded because they are unreliable. Resampled particles have equal weights and the value of a large weight is to be expressed with a large number of particles. Fig. 8 outlines the flow of resampling.

IV. EXPERIMENTS AND RESULTS

This section explains two experiments on our beat-tracking method. First, we compared our method with Murata *et al.*'s method [3] that uses STPM as much as ours, but rule based beat-detecting. In addition, we confirmed the effect of adding visual features by comparing with particle-filter based audio only beat-tracking. Second, we discusses relationship between the number of particles, the speed of computation, and the accuracy of the estimates.

TABLE III
TABLE OF RESULTS

(a) Precision (%)

Pattern	1	2	3	4	5	6	7	8	Ave.
Integrated	60.7	74.1	61.1	**67.9**	52.4	**54.0**	**65.4**	**52.4**	**61.0**
Audio only	48.6	51.3	48.4	50.6	50.6	44.2	49.0	46.8	47.5
Murata	**91.5**	**88.2**	**88.3**	56.9	**57.7**	35.6	29.0	23.1	59.5

(b) Recall (%)

Pattern	1	2	3	4	5	6	7	8	Ave.
Integrated	78.4	84.0	69.8	**74.7**	**60.1**	**48.2**	**79.3**	32.3	**65.9**
Audio only	37.6	42.8	40.2	40.9	38.0	41.9	43.7	**45.2**	41.3
Murata	**90.9**	**88.2**	**82.2**	56.9	53.0	29.2	29.0	22.5	56.5

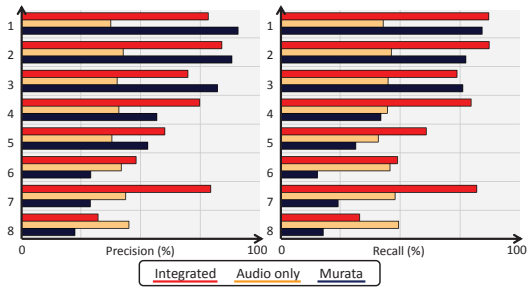

Fig. 9. Precision and recall of results.

A. Condition

We used music data of three tempo (70, 90, and 110) and eight beat patterns from two performers. The beat patterns are given in Fig. 2. We designed the beat patterns to order in the number of accented down beats. In addition, the beat patterns have more accented up beats as the index number of patterns increased. The tempo standard deviation of inter-onset interval ratio in our data is 0.0575 whereas the one of pop songs is theoretically zero. Our method uses 100 particles unless otherwise stated. The camera recorded frames at about 19 [fps]. The distance between the robot and guitarist was about 3 [m] and the entire guitar appeared in the camera frame. We regard the range between the estimates and correct values to be within $\pm 150[msec]$ in beat-time and ± 10[BPM] to be successful estimation. Then, we calculate the precision ($r_{prec} = N_e/N_d$) and recall($r_{recall} = N_e/N_c$) of each pattern where N_e, N_d, and N_c correspond to the number of correct estimates, whole estimates and correct beats. Our system was implemented in C++ on a Ubuntu10.04 operating system (32 bits) with an Intel Core 2 Quad processor and it was speeded up by a divided thread process in particle computation. In addition, we used the -O3 option in compilation. Here we will discuss the speed of computation only in the particle filter process.

B. Comparison of performance

Table III and Fig. 9 show the summarization of the precision and recall of each pattern with our audio-visual integrated beat-tracking ("Integrated"), our audio only one ("Audio only") and Murata *et al.*'s method ("Murata"). The

TABLE IV

RELATIONSHIP BETWEEN THE NUMNER OF PARTICLES, ESTIMATION ACCURACY, AND COMPUTATIONAL SPEED.

Number of particles	100	200	400
Real-time factors	0.48	0.88	1.69
Precision (%)	65.9	65.6	66.6
Recall (%)	69.1	68.4	68.8

results for "Integrated" are constantly high; however, the results for "Murata" decrease as the number of beat patterns increases. This demonstrates the robustness of our method against beat patterns. The comparison between "Integrated" and "Audio only" convinced us of the validity of adding visual features with higher results for "Integrated" by 13.5 points and 24.6 points to the averages of precision for the former and recall for the latter.

Here, we will discuss Patterns 5, 6 and 8 in which the "Integrated" results have lower scores than those in other patterns. The stroke cycle on these patterns lasts tha length of an eight note. As this cycle does not match the hand trajectory model (in Eq. (5)) under assumption (1) in Section II-A, this worsens the results in their patterns.

The average result for "Integrated" shows only about 70%. This is due to two main reasons: 1) the mismatch between the hand trajectory model and the actual model described above and, 2) the low resolution and the error in estimating visual features has an insignificant effect of the penalty function to modify the θ distribution.

C. Alternation of results by number of particles

Comparisons of the accuracy of estimation and the speed on each number of particles are summarized in Table IV. First, we will discuss the computational speed, represented by a real-time factor. Here, the real-time factor is a criterion of the quantitative evaluation for a real-time system and is calculated as the (execution time)/(performance time). This enables us to compare the speed of estimation trials without regarding the differences in performance times. Table IV shows the real-time factor is proportional to the number of particles and may be about 1 at just over 200 particles. We therefore concluded our method worked well as a real-time system with fewer than 200 particles.

The accuracy of estimation is indicated by result values in the table. The alternation in the number of particles barely made any difference to estimates. The reasons for this are to be resolved in future work.

V. CONCLUSIONS AND FUTURE WORKS

We introduced particle-filter based audio-visual beat-tracking of guitar performances for ensemble music robots. Our beat-tracking is robust against tempo changes in human performance and varying note lengths for guitar beat patterns. This enables the music robot ensemble to be more expressive. The experiments revealed that our method estimates the tempo and bar-position at a certain level without the dependence on beat patterns. Moreover, we confirmed its real-time performance in computational speed.

We have to solve two issues for a well harmonized ensemble: refining our beat-tracking to achieve better accuracy and action against errors. To solve the first issue, we can discuss the state and observation models. We should take into account of the hand-trajectory model of an eight-note cycle. A choice of cycles may be expressed with a new variable in a particle filter. In addition, hand-tracking needs to be refined. Some researchers have reported advances in visual-tracking with a recently developed infrared sensor. Adapting these sensors to robots may improve the accuracy of estimating hand-tacking and thus beat-tracking. To solve the second issue, we present two resolutions as future work: the use of scores that consist of melody and chord names, which many guitar scores use, and the prior distribution of rhythm patterns as previous knowledge on the particle filter.

VI. ACKNOWLEDGMENTS

This research was supported in part by Kakenhi #19100003 and #22118502 and in part by Kyoto University's Global COE.

REFERENCES

[1] G. Weinberg *et al.* The Creation of a Multi-Human, Multi-Robot Interactive Jam Session. In *Proc. of Int'l Conf. on New Interfaces of Musical Expression*, pages 70–73, 2009.

[2] T. Mizumoto *et al.* Integration of flutist gesture recognition and beat tracking for human-robot ensemble. In *Proc. of IEEE/RSJ-2010 Workshop on Robots and Musical Expression*, pages 159–171, 2010.

[3] K. Murata *et al.* A beat-tracking robot for human-robot interaction and its evaluation. In *Proc. of 8th IEEE-RAS Int'l Conf. on Humanoids*, pages 79–84. IEEE, 2008.

[4] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *Proc. of the 7th IEEE Int'l Conf. on Computer Vision*, volume 2, pages 1197–1203, 2002.

[5] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[6] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *J. of New Music Research*, pages 159–171, 2001.

[7] N. Whiteley *et al.* Bayesian modelling of temporal structure in musical audio. In *Proc. of the 7th Int'l Conf. on Music Information Retrieval*, pages 29–34, 2006.

[8] W.A. Sethares *et al.* Beat tracking of musical performances using low-level audio features. *Speech and Audio Processing, IEEE Transactions on*, 13(2):275–285, 2005.

[9] T. Otsuka *et al.* Design and Implementation of Two-level Synchronization for Interactive Music Robot. In *Proc. of Association for the Advancement of Artificial Intelligence*, pages 1238–1244, 2010.

[10] Y. Pan *et al.* A robot musician interacting with a human partner through initiative exchange. In *Proc. of the 2010 Conf on New Interfaces for Musical Expression*, pages 166–169, 2010.

[11] K. Petersen *et al.* Development of a real-time instrument tracking system for enabling the musical interaction with the Waseda Flutist Robot. In *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 313–318, 2008.

[12] A. Lim *et al.* Robot musical accompaniment: integrating audio and visual cues for real-time synchronization with a human flutist. In *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 1964–1969, 2010.

[13] K. Nickel *et al.* A joint particle filter for audio-visual speaker tracking. In *Proc. of the 7th Int'l Conf. on multimodal interfaces*, pages 61–68, 2005.

[14] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of Int'l joint Conf. on artificial intelligence*, volume 3, pages 674–679, 1981.

[15] D. Miyazaki *et al.* Polarization-based inverse rendering from a single view. In *Proc. of 9th IEEE Int'l Conf. on Computer Vision*, pages 982–987, 2003.

[16] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.