A multi-modal tempo and beat tracking system based on audio-visual information from live guitar performances

AUTHOR(S):

Itohara, Tatsuhiko; Otsuka, Takuma; Muzumoto, Takeshi; Lim, Angelica; Ogata, Tetsuya; Okuno, Hiroshi G.

CITATION:

Itohara, Tatsuhiko ...[et al]. A multi-modal tempo and beat tracking system based on audio-visual information from live guitar performances. EURASIP Journal on Audio, Speech, and Music Processing 2012, 2012: 6.

ISSUE DATE:

2012-01-20

URL:

http://hdl.handle.net/2433/187381

Itohara *et al. EURASIP Journal on Audio, Speech, and Music Processing* 2012, **2012**:6
http://asmp.eurasipjournals.com/content/2012/1/6

EURASIP Journal on Audio, Speech, and Music Processing
a SpringerOpen Journal

RESEARCH Open Access

A multimodal tempo and beat-tracking system based on audiovisual information from live guitar performances

Tatsuhiko Itohara*, Takuma Otsuka, Takeshi Mizumoto, Angelica Lim, Tetsuya Ogata and Hiroshi G Okuno

Abstract

The aim of this paper is to improve beat-tracking for live guitar performances. Beat-tracking is a function to estimate musical measurements, for example musical tempo and phase. This method is critical to achieve a synchronized ensemble performance such as musical robot accompaniment. Beat-tracking of a live guitar performance has to deal with three challenges: tempo fluctuation, beat pattern complexity and environmental noise. To cope with these problems, we devise an audiovisual integration method for beat-tracking. The auditory beat features are estimated in terms of tactus (phase) and tempo (period) by Spectro-Temporal Pattern Matching (STPM), robust against stationary noise. The visual beat features are estimated by tracking the position of the hand relative to the guitar using optical flow, mean shift and the Hough transform. Both estimated features are integrated using a particle filter to aggregate the multimodal information based on a beat location model and a hand's trajectory model. Experimental results confirm that our beat-tracking improves the F-measure by 8.9 points on average over the Murata beat-tracking method, which uses STPM and rule-based beat detection. The results also show that the system is capable of real-time processing with a suppressed number of particles while preserving the estimation accuracy. We demonstrate an ensemble with the humanoid HRP-2 that plays the theremin with a human guitarist.

1 Introduction

Our goal is to improve beat-tracking for human guitar performances. Beat-tracking is one way to detect musical measurements such as beat timing, tempo, body movement, head nodding, and so on. In this paper, the proposed beat-tracking method estimates tempo, beats per minute (bpm), and *tactus*, often referred to as the foot tapping timing or the beat [1], of music pieces.

Toward the advancement of beat-tracking, we are motivated with an application to musical ensemble robots, which enable synchronized play with human performers, not only expressively but also interactively. Only a few attempts, however, have been made so far with interactive musical ensemble robots. For example, Weinberg et al. [2] reported a percussionist robot that imitates a co-player's playing to play according to the co-player's timing. Murata et al. [3] addressed a musical robot ensemble with robot noise suppression with the Spectro-Temporal Pattern Matching (STPM) method. Mizumoto et al. [4] report a thereminist robot that performs a trio with a human flutist and a human percussionist. This robot adapts to the changing tempo of the human's play, such as *accelerando* and *fermata*.

We focus on the beat-tracking of a guitar played by a human. The guitar is one of the most popular instruments used for casual musical ensembles consisting of a melody and a backing part. Therefore, the improvement of beat-tracking of guitar performances enables guitarist, from novices to experts, to enjoy applications such as a beat-tracking computer teacher or an ensemble with musical robots.

In this paper, we discuss three problems in beat-tracking of live human guitar performances: (1) tempo fluctuation, (2) complexity of beat patterns, and (3) environmental noise. The first is caused by the irregularity of humans. The second is illustrated in Figure 1; some patterns consist of upbeats, that is, syncopation. These patterns are often observed in guitar playing. Moreover, beat-tracking of one instrument, especially in

* Correspondence: itohara@kuis.kyoto-u.ac.jp
Graduate School of Informatics, Kyoto University, Sakyo, Kyoto, Japan

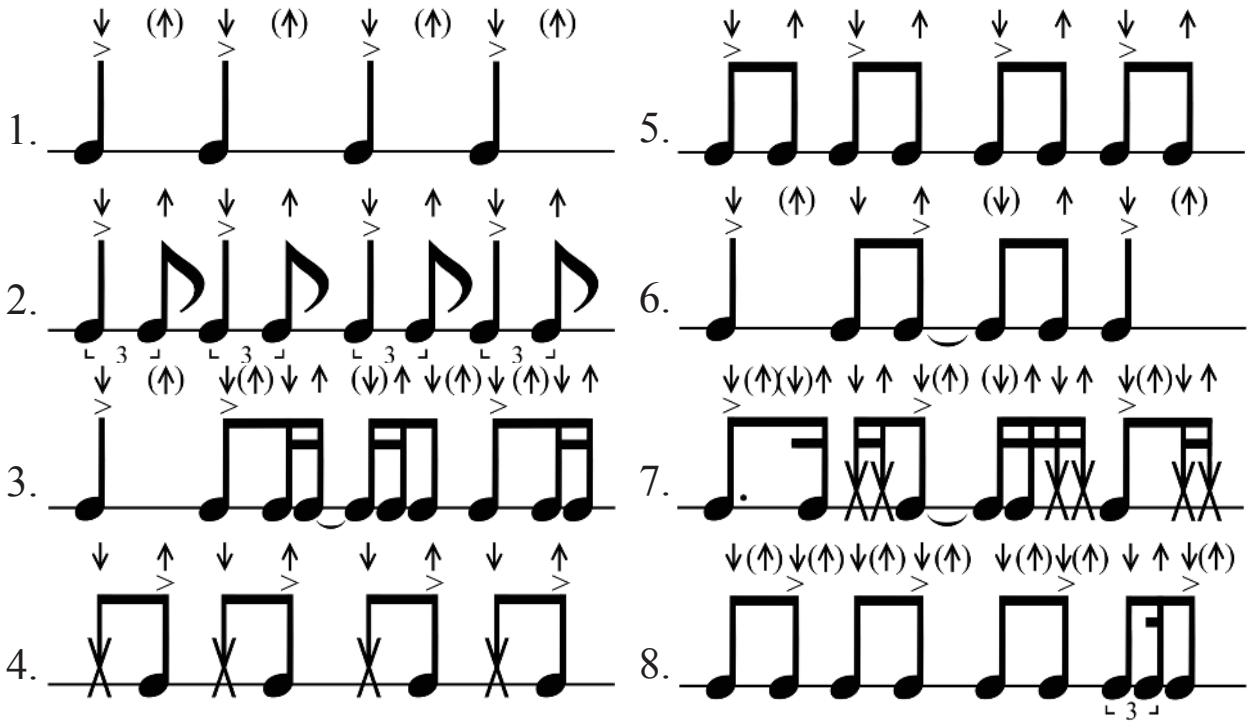

Figure 1 Typical guitar beat patterns. The symbol × represents guitar-cutting, a percussive sound made with quick muting sounds. The > denotes accented, ↑ and ↓ denote the directions of strokes, and (↑) and (↓) denote air strokes.

syncopated beat patterns, is challenging since beat-tracking of one instrument has less onset information than with many instruments. For the third, we focus on stationary noise, for example, small perturbations in the room, and robot fan noise. It degrades the signal-to-noise ratio of the input signal, so we cannot disregard such noise.

To solve these problems, this paper presents a particle-filter-based audiovisual beat-tracking method for guitar playing. Figure 2 shows the architecture of our method. The core of our method is a particle-filter-based integration of the audio and visual information based on a strong correlation between motions and beat timings of guitar playing. We modeled their relationship in the probabilistic distribution of our particle-filter method. Our method uses the following audio and visual beat features: the audio beat features are the normalized cross-correlation and increments obtained from the audio signal using Spectro-Temporal Pattern Matching (STPM), a method robust against stationary noise, and the visual beat features are the relative hand positions from the neck of the guitar.

We implement a human-robot ensemble system as an application of our beat-tracking method. The robot plays its instrument according to the guitar beat and tempo. The task is challenging because the robot fan and motor noise interfere with the guitar's sound. All of our experiments are conducted in the situation with the robot.

Section 2 discusses the problems with guitar beat-tracking, and Section 3 presents our audiovisual beat-tracking approach. Section 4 shows that the experimental results demonstrate the superiority of our beat-tracking to Murata's method in tempo changes, beat structures and real-time performance. Section 5 concludes this paper.

2 Assumptions and problems

2.1 Definition of the musical ensemble with guitar

Our targeted musical ensemble consists of a melody player and a guitarist and assumes quadruple rhythm for simplicity of the system. Our beat-tracking method can accept other rhythms by adjusting the hand's trajectory model explained in Section 3.2.3.

At the beginning of a musical ensemble, the guitarist gives some *counts* to synchronize with a co-player as he would in real ensembles. These counts are usually given by voice, gestures or hit sounds from the guitar. We determine the number of counts as four and consider that the tempo of the musical ensemble can be only altered moderately from the tempo implied by counts.

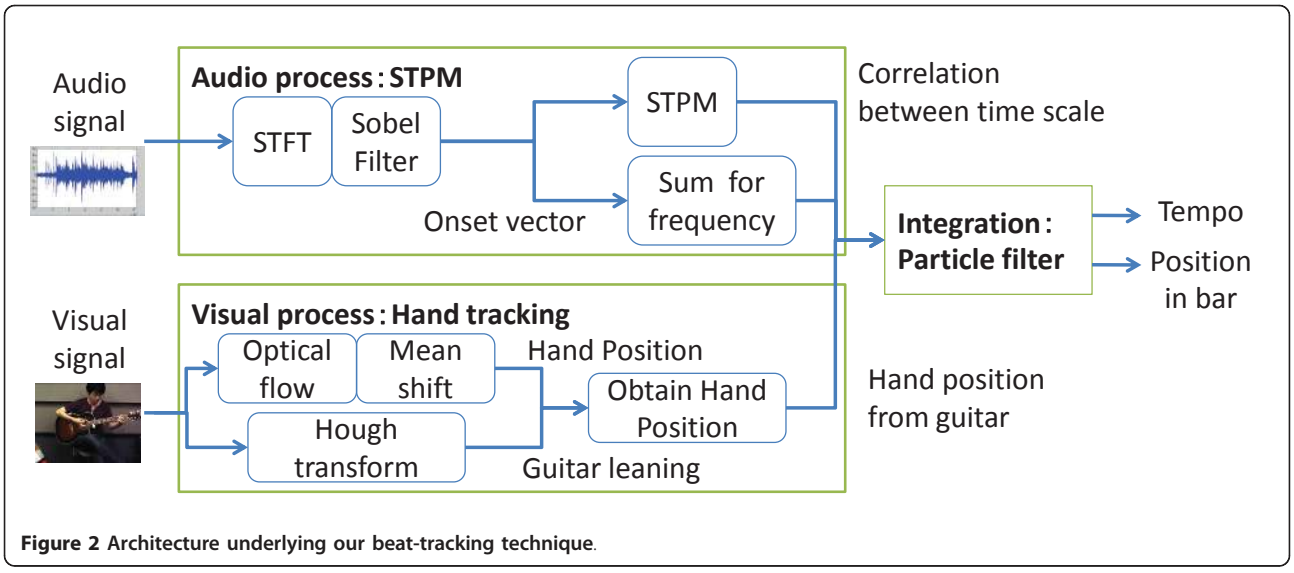

Figure 2 Architecture underlying our beat-tracking technique.

Our method estimates the beat timings without prior knowledge of the co-player's score. This is because (1) many guitar scores do not specify beat patterns but only melody and chord names, and (2) our main goal focuses on improvisational sessions.

Guitar playing is mainly categorized into two styles: stroke and arpeggio. Stroke style consists of hand waving motions. In arpeggio style, however, a guitarist pulls strings with their fingers mostly without moving their arms. Unlike most beat-trackers in the literature, our current system is designed for a much more limited case where the guitar is strummed, not in a finger picked situation. This limitation allows our system to perform well in a noisy environment, to follow sudden tempo changes more reliably and to address single instrument music pieces.

Stroke motion has two implicit rules, (1) beginning with a down stroke and (2) air strokes, that is, strokes with a soundless tactus, to keep the tempo stable. These can be found in the scores, especially pattern 4 for air strokes, in Figure 1. The arrows in the figure denote the stroke direction, common enough to appear on instruction books for guitarists. The scores say that 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 note (sixteen beats). We assume music with eight-beat measures and model the hand's trajectory and beat locations.

No prior knowledge on the color of hands is assured in our visual-tracking. This is because humans have various hand colors and such colors vary according to the lighting conditions. The motion of the guitarist's arm, on the other hand, is modeled with prior knowledge: the stroking hand makes the largest movement in the body of a playing guitarist. The conditions and assumptions for guitar ensemble are summarized below:

Conditions and assumptions for beat-tracking

Conditions:

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

Assumptions:

(1) Quadruple rhythm
(2) Not much variance from the tempo implied by counts
(3) Hand movement and beat locations according to eight beats
(4) Stroking hand makes the largest movement in the body of a guitarist

2.2 Beat-tracking conditions

Our beat-tracking method estimates the tempo and *bar-position*, the location in the bar at which the performer is playing at a given time from audio and visual beat features. We use a microphone and a camera embedded in the robot's head for the audio and visual input signal, respectively. We summarize the input and output specifications in the following box:

Input-output

Input:

- Guitar sounds captured with robot's microphone
- Images of guitarist captured with robot's camera

Output:

- Bar-position
- Tempo

2.3 Challenges for guitar beat-tracking

A human guitar beat-tracking must overcome three problems to cope with tempo fluctuation, beat pattern complexity, and environmental noise. The first problem is that, since we do not assume a professional guitarist, a player is allowed to play fluid tempos. Therefore, the beat-tracking method should be robust to such changes of tempo.

The second problem is caused by (1) beat patterns complicated by upbeats (syncopation) and (2) the sparseness of onsets. We give eight typical beat patterns in Figure 1. Patterns 1 and 2 often appear in popular music. Pattern 3 contains triplet notes. All of the accented notes in these three patterns are down beats. However, the other patterns contain accented upbeats. Moreover, all of the accented notes of patterns 7 and 8 are upbeats. Based on these observations, we have to take into account how to estimate the tempos and bar-positions of the beat patterns with accented upbeats.

The sparseness is defined as the number of onsets per time unit. We illustrate the sparseness of onsets in Figure 3. In this paper, guitar sounds consist of a simple strum, meaning low onset density, while popular music has many onsets as is shown in the Figures. The figure shows a 62-dimension mel-scaled spectrogram of music after the Sobel filter [5]. The Sobel filter is used for the enhancement of onsets. Here, the negative values are set to zero. The concentration of darkness corresponds to strength of onset. The left one, from popular music, has equal interval onsets including some notes between the onsets. On the other hand, the right one shows an absent note compared with the tactus. Such absences mislead a listener of the piece as per the blue marks in the figure. What is worse, it is difficult to detect the tactus in a musical ensemble with few instruments because there are few supporting notes to complement the syncopation; for example, the drum part may complement the notes in larger ensembles.

As for the third problem, the audio signal in beat-tracking of live performances includes two types of noise: stationary and non-stationary noise. In our robot application, the non-stationary noise is mainly caused by the robot joints' movement. This noise, however, does not affect beat-tracking, because it is small–6.68 dB in signal-to-noise ratio (SNR)–based on our experience so far. If the robot makes loud noise when moving, we may apply Ince's method [6] to suppress such ego noise. The stationary noise is mainly caused by fans on the computer in the robot and environmental sounds including air-conditioning. Such noise degrades the signal-to-noise

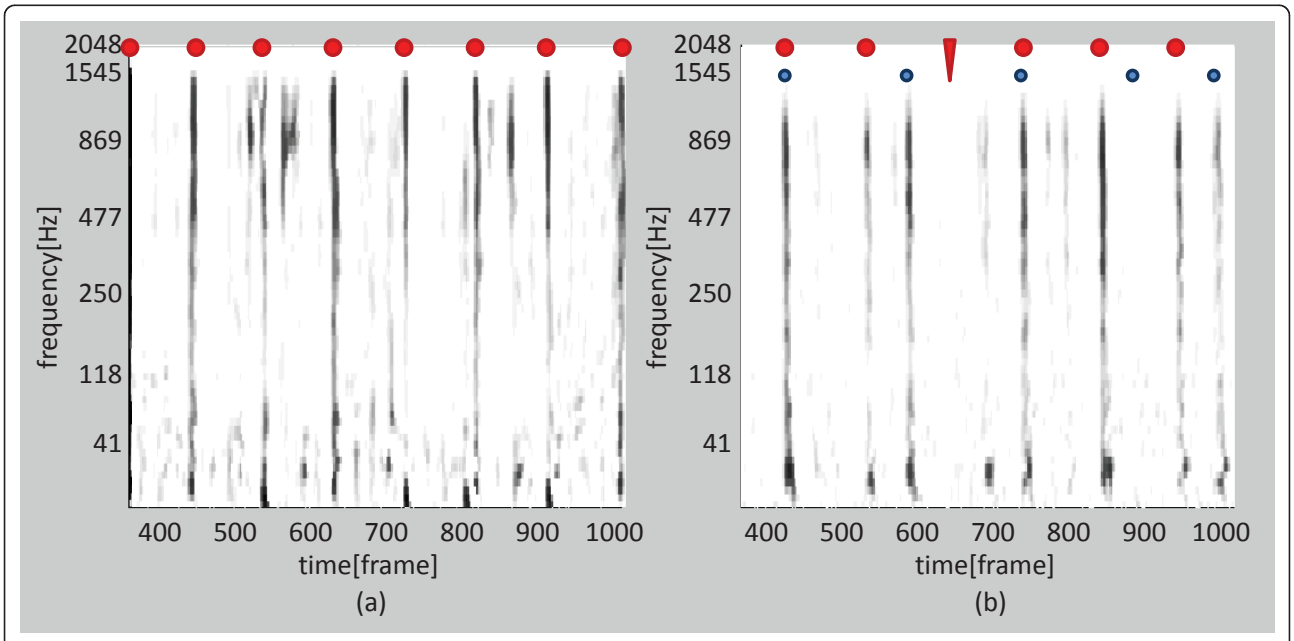

Figure 3 The strength of onsets in each frequency bin with the power spectrogram after Sobel filtering. **a** Popular music (120 BPM), **b** guitar backing performance (110 bpm). Red ballets, red triangles, blue ballet denote tactuses of the pieces, absent notes at tactuses, error candidates of tactuses. In this paper, a frame is equivalent to 0.0116 sec. Detailed parameter values about time frame are shown in Section 3.1.

ratio of the input signal, for example, 5.68dB in SNR, in our experiments with robots. Therefore, our method should include a stationary noise suppression method.

We have two challenges for visual hand tracking: false recognition of the moving hand and low time resolution compared with 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 thus captures other nearly skin-colored objects. While optical-flow-based methods are considered suitable for hand tracking, we have difficulty in employing this method because flow vectors include some noise from the movements of other parts of the body. Usually, audio and visual signals have different sampling rates from one another. According to our setting, 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.

problems

Audio signal:

(1) Complexity of beat patterns
(2) Sparseness of onsets
(3) Fluidity of human playing tempos
(4) Antinoise signal

Visual signal:

(1) Distinguishing hand from other parts of body
(2) Variations in hand color depend on individual humans and their surroundings
(3) Low visual resolution

2.4 Related research and solution of the problems

2.4.1 Beat-tracking

Beat-tracking has been extensively studied in music processing. Some beat-tracking methods use agents [7,8] that independently extract the inter-onset intervals of music and estimate tempos. They are robust against beat pattern complexity but vulnerable to tempo changes because their target music consists of complex beat patterns with a stable tempo. Other methods are based on statistical methods like a particle filter using a MIDI signal [9,10]. Hainsworth improves the particle-filter-based method to address raw audio data [11].

For the adaptation to robots, Murata achieved a beat-tracking method using the SPTM method [3], which suppresses robot stationary noise. While this STPM-based method is designed to adapt to sudden tempo changes, the method is likely to mistake upbeats 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 upbeats with its beat-detecting rule.

In order to robustly track the human's performance, Otsuka et al. [12] use a musical score. They have reported an audio-to-score alignment method based on a particle filter and revealed its effectiveness despite tempo changes.

2.4.2 Visual-tracking

We use two methods for visual-tracking, one based on optical flow and one based on color information. With the optical-flow method, we can detect the displacement of pixels between frames. For example, Pan et al. [13] use the method to extract a cue of exchanged initiatives for their musical ensemble.

With color information, we can compute the prior probabilistic distribution for tracked objects, for example, with a method based on particle filters [14]. There have been many other methods for extracting the positions of instruments. Lim et al. [15] use a Hough transform to extract the angle of a flute. Pan et al. [13] use a mean shift [16,17] to estimate the position of the mallet's endpoint. These detected features are used as the cue for the robot movement. In Section 3.2.2, we give a detailed explanation of Hough transform and mean shift.

2.4.3 Multimodal integration

Integrating the results of elemental methods is a filtering problem, where observations are input features extracted with some preprocessing methods and latent states are the results of integration. The Kalman filter [18] produces estimates of latent state variables with linear relationships between observation and the state variables based on a Gaussian distribution. The Extended Kalman Filter [19] adjusts the state relationships of non-linear representations but only for differentiable functions. These methods are, however, unsuitable for the beat-tracking we face because of the highly non-linear model of the hand's 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, for example, a non-Gaussian distribution. At frame t, z_t and x_t denote the variables of the observation and latent states, respectively. The probability density function (PDF) of latent state variables $p(x_t|z_{1:t-1})$ is approximated as follows:

$$p(x_t|z_{1:t}) \approx \sum_{i=1}^{I} \omega_t^{(i)} \delta\left(x_t - x_t^{(i)}\right), \tag{1}$$

where the sum of weights $w_t^{(i)}$ is 1. I is the number of particles and $w_t^{(i)}$ and $x_t^{(i)}$ correspond to the weight and state variables of the ith particle, respectively. The $\delta(x_t - x_t^{(i)})$ is the Dirac delta function. Particle filters

are commonly used for beat-tracking [9-12] and visual-tracking [14] as is shown in Section 2.4.1 and 2.4.2. Moreover, Nickel et al. [20] applied a particle filter as a method of audiovisual integration for the 3D identification of a talker. We will present the solution for these problems in the next section.

3 Audio and visual beat features extraction

3.1 Audio beat feature extraction with STPM

We apply the STPM [3] for calculating the audio beat 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 44.1kHz. A Hamming window of 4,096 points with the shift size of 512 points is used as a window function. The 2,049 linear frequency bins are reduced to 64 mel-scaled frequency bins by a mel-scaled filter bank. Then, the Sobel filter [5] is applied to the spectra to enhance its edges and to suppress the stationary noise. Here, the negative values of its result are set to zero. The resulting vector, $d(t,f)$, is called an onset vector. Its element at the tth time frame and f-th mel-frequency bank is defined as follow:

$$d(t,f) = \begin{cases} p_{\text{sobel}}(t,f) & \text{if } p_{\text{sobel}}(t,f) > 0, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\begin{aligned} p_{\text{sobel}}(t,f) = &-p_{\text{mel}}(t-1,f+1) + p_{\text{mel}}(t+1,f+1) \\ &- p_{\text{mel}}(t-1,f-1) + p_{\text{mel}}(t+1,f-1) \\ &- 2p_{\text{mel}}(t-1,f) + 2p_{\text{mel}}(t+1,f), \end{aligned} \tag{3}$$

where p_{sobel} is the spectra to which the Sobel filter is applied to. $R_t(k)$, the inter-frame correlation with the frame k frames behind, is calculated by the normalized cross-correlation (NCC) of onset vectors defined in Eq. (4). This is the result for STPM. In addition, we define F_t as the sum of the values of the onset vector at the tth time frame in Eq. (5). F_t refers to the peak time of onsets. $R_t(k)$ relates to the musical tempo (period) and F_t to the tactus (phase).

$$R_t(k) = \frac{\sum_{j=1}^{N_F}\sum_{i=0}^{N_{P-1}} d(t-i,j)d(t-k-i,j)}{\sqrt{\sum_{j=1}^{N_F}\sum_{i=0}^{N_{P-1}} d(t-i,j)^2 \sum_{j=1}^{N_F}\sum_{i=0}^{N_{P-1}} d(t-k-i,j)^2}} \tag{4}$$

$$F_t = \log\left(\sum_{f=1}^{N_F} d(t,f)\right) \quad peak, \tag{5}$$

where $peak$ is a variable for normalization and is updated under the local peak of onsets. The N_F denotes the number of dimensions of onset vectors used in NCC and N_P 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].

3.2 Visual beat feature extraction with hand tracking

We extract the visual beat features, that is, the temporal sequences of hand positions with these three methods: (1) hand candidate area estimation by optical flow, (2) hand position estimation by mean shift, and (3) hand position tracking.

3.2.1 Hand candidate area estimation by optical flow

We use Lucas-Kanade (LK) method [21] for fast optical-flow calculation. Figure 4 shows an example of the result of optical-flow calculation. We define the center of hand candidate area as a coordinate of the flow vector, which has the length and angle nearest from the middle values of flow vectors. This is because the hand motion should have the largest flow vector according to the assumption (3) in Section 2.1, and this allows us to remove noise vectors with calculating the middle values.

3.2.2 Hand position estimation by mean shift

We estimate a precise hand position using mean shift [16,17], a local maximum detection method. Mean shift has two advantages: low computational costs and robustness against outliers. We used the hue histogram as a kernel function in the color space which is robust against shadows and specular reflections [22] defined by:

$$\begin{pmatrix} I_x \\ I_y \\ I_z \end{pmatrix} = \begin{pmatrix} 2 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/3 & 1/3 & 1/3 \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix} \tag{6}$$

$$hue = \tan^{-1}(I_y/I_x). \tag{7}$$

3.2.3 Hand position tracking

Let $(h_{x,t}, h_{y,t})$ be the hand coordination calculated by the mean shift. Since a guitarist usually moves their hand near the neck of their guitar, we define r_t, a hand position at t time frame, as the relative distance between the hand and the neck as follows:

$$r_t = \rho_t - (h_{x,t}\cos\theta_t + h_{y,t}\sin\theta_t), \tag{8}$$

where ρ_t and θ_t are the parameters of the line of the neck computed with Hough transform [23] (see Figure 5a for an example). In Hough transform, we compute 100 candidate lines, remove outliers with RANSAC [24], and get the average of Hough parameters. Positive values indicate that a hand is above the guitar; negative values indicate below. Figure 5b shows an example of the sequential hand positions.

Figure 4 Optical flow. **a** is the previous frame, **b** is the current frame, and **c** indicates flow vecto The horizontal axis and the vertical axis correspond to the time frame and hand positio respectively.

Now, let ω_t and θ_t be a beat interval and bar-position at the tth 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 2.1, we presume that down strokes are at $\theta_t = n\pi/2$ and up strokes are at $\theta_t = n\pi/2 + \pi/4 (n = 0,1,2,3)$. In other words, zero crossover points of hand position are at these θ. In addition, since a hand stroking is in a smooth motion to keep the tempo stable, we assume that the sequential hand position can be represented with a continuous function. Thus, hand position r_t is defined by

$$r_t = -a \sin(4\theta_t), \tag{9}$$

where a is a constant value of hand amplitude and is set to 20 in this paper.

4 Particle-filter-based audiovisual integration

4.1 Overview of the particle-filter model

The graphical representation of the particle-filter model is outlined in Figure 6. The state variables, ω_t and θ_t, denote the beat interval and bar-position, respectively. The observation variables, $R_t(k)$, F_t, and r_t denote inter-frame correlation with k frames back, normalized onset summation, and hand position, respectively. The $w_t^{(i)}$ and $\theta_t^{(i)}$ are parameters of the ith particle. Now, we will explain the estimation process with the particle filter.

4.2 State transition with sampling

The state variables at the tth time frame $\left[\omega_t^{(i)} \theta_t^{(i)}\right]$ are sampled from Eqs. (10) and (11) with the observations at the $(t - 1)$th time frame. We use the following proposal distributions:

$$\begin{aligned} \omega_t^{(i)} &\sim q\left(\omega_t | \omega_{t-1}^{(i)}, R_t(\omega_t), \omega_{\text{init}}\right) \\ &\propto R_t(\omega_t) \times \text{Gauss}\left(\omega_t | \omega_{t-1}^{(i)}, \sigma_{w_q}\right) \\ &\quad \times \text{Gauss}\left(\omega_t | \omega_{\text{init}}, \sigma_{w_{\text{init}}}\right) \end{aligned} \tag{10}$$

$$\begin{aligned} \theta_t^{(i)} &\sim q\left(\theta_t | r_t, F_t, \omega_{t-1}^{(i)}, \theta_{t-1}^{(i)}\right) \\ &= \text{Mises}\left(\theta_t | \hat{\Theta}_t^{(i)}, \beta_{\theta_q}, 1\right) \times \text{penalty}(\theta_t^{(i)} | r_t, F_t), \end{aligned} \tag{11}$$

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,

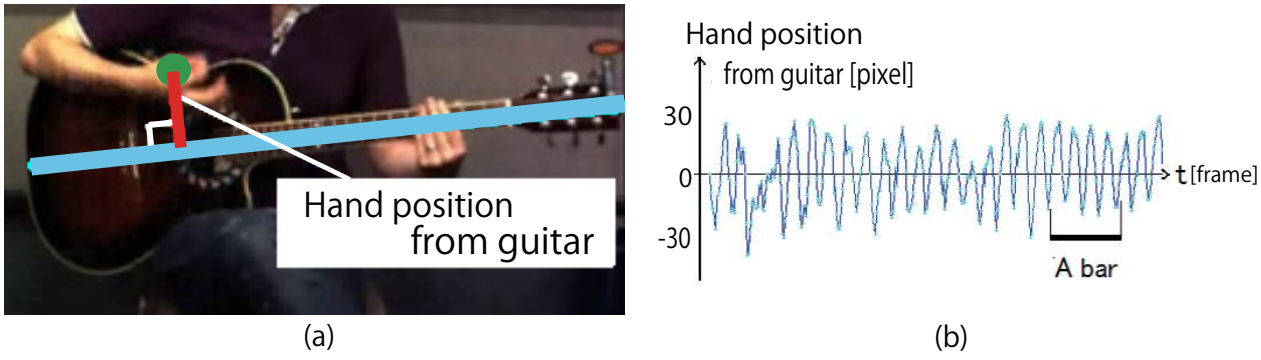

Figure 5 Hand position from guitar. **a** Definition image. **b** Example of sequential data.

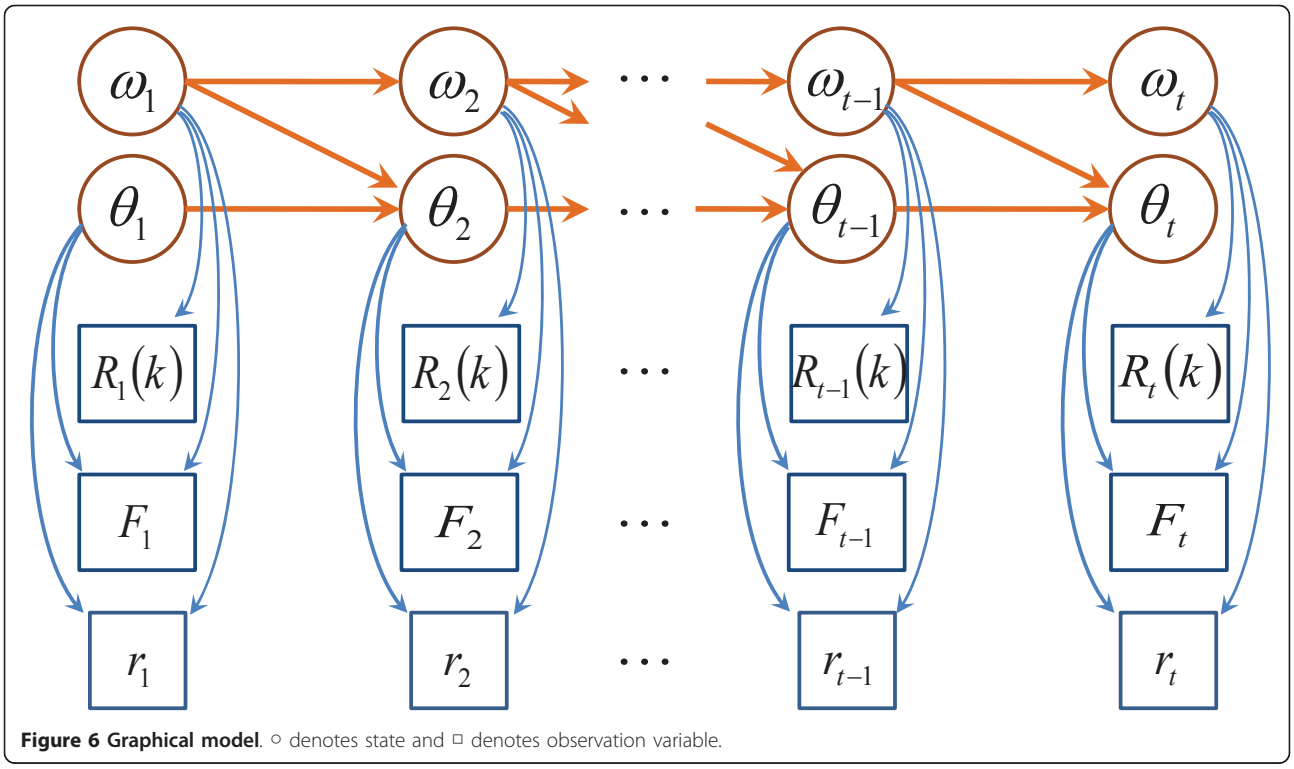

Figure 6 Graphical model. ○ denotes state and □ denotes observation variable.

respectively. The σ_{ω_*} denotes the standard deviation for the sampling of the beat interval. The ω_{init} denotes the beat interval estimated and fixed with the counts. Mises $(\theta|\mu, \beta, \tau)$ represents the PDF of a von Mises distribution [25], also known as the circular normal distribution, which is modified to have τ peaks. This PDF is defined by

$$\text{Mises}(\theta|\mu,\beta,\tau) = \frac{\exp(\beta\cos(\tau(\theta-\mu)))}{2\pi \text{I}_0(\beta)}, \tag{12}$$

where $\text{I}_0(\beta)$ is a modified Bessel function of the first kind of order 0. The μ denotes the location of the peak. The β denotes the concentration; that is, $1/\beta$ is analogous to σ^2 of a normal distribution. Note that the distribution approaches a normal distribution as β increases. Let $\hat{\Theta}_t^{(i)}$ be a prediction of $\theta_t^{(i)}$ defined by:

$$\hat{\Theta}_t^{(i)} = \theta_{t-1}^{(i)} + b/\omega_{t-1}^{(i)}, \tag{13}$$

where b denotes a constant for transforming from beat interval into an angle rate of the bar-position.

We will now discuss Eqs. (10) and (11). In Eq. (10), 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. (11), penalty$(\theta|r, F)$ is the result of five multiplied multipeaked window functions. Each function has a condition. If it is satisfied, the function is defined by the von Mises distribution; otherwise, it shows 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. Figure 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 definition of the distribution.

$$r_{t-1} > 0 \cap r_t < 0 \Rightarrow \text{Mises}(0, 2.0, 4) \tag{14}$$

$$r_{t-1} < 0 \cap r_t > 0 \Rightarrow \text{Mises}(\frac{\pi}{4}, 1.9, 4) \tag{15}$$

$$r_{t-1} > r_t \Rightarrow \text{Mises}(0, 3.0, 4) \tag{16}$$

$$r_{t-1} < r_t \Rightarrow \text{Mises}(\frac{\pi}{4}, 1.5, 4) \tag{17}$$

$$F_t > \textit{thresh.} \Rightarrow \text{Mises}(0, 20.0, 8). \tag{18}$$

All β parameters are set experimentally through a trial and error process. *thresh.* is a threshold that determines whether F_t is constant noise or not. Eqs. (14) and (15) are determined with the assumption of zero crossover points of stroking. Eqs. (16) and (17) are determined with the stroking directions. These four equations are based on the model of the hand's trajectory presented in

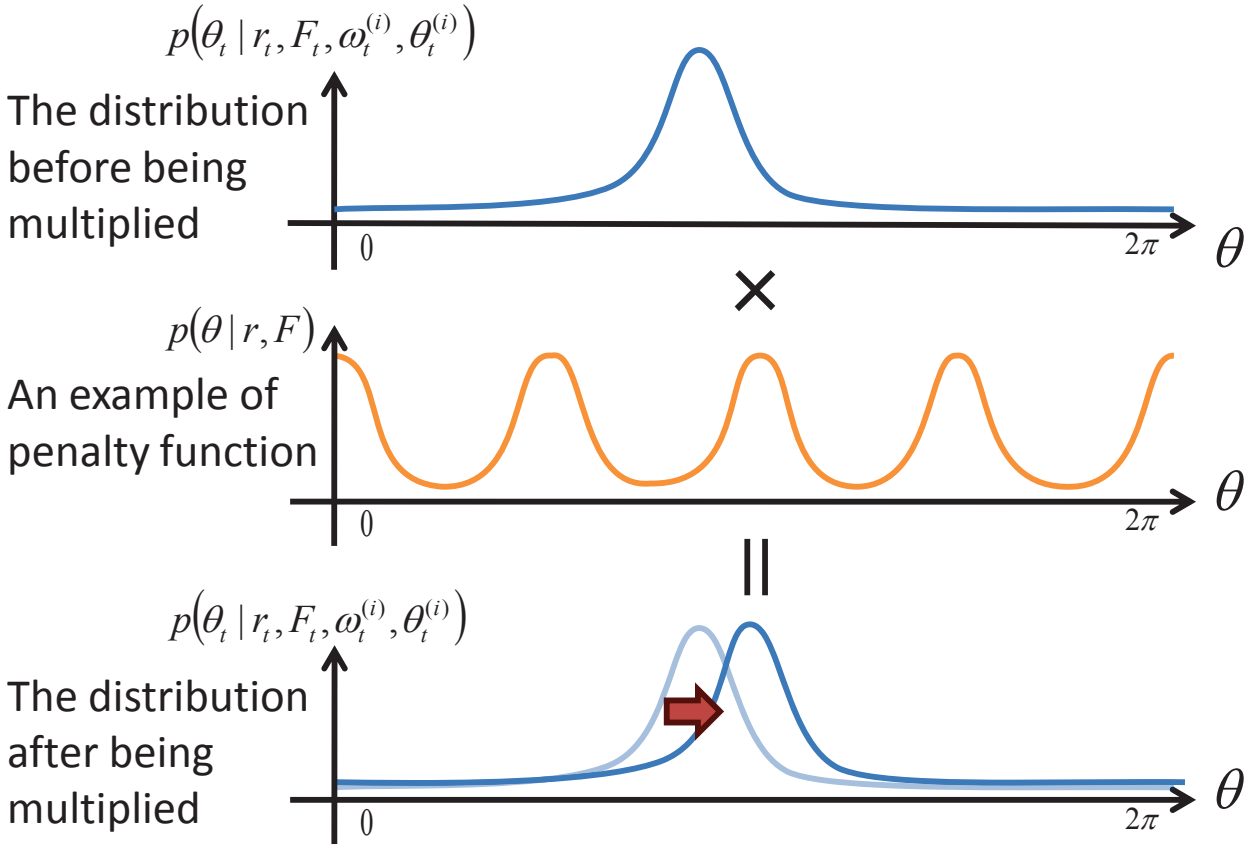

Figure 7 Example of changes in θ distribution while multiplying *penalty* function. Beginning the top, we show the distribution before being multiplied, an example of the *penalty* functio and the distribution after being multiplied. This *penalty* function is expressed by the von Mis distribution of the cycle of $\pi/2$.

Eq. (9). Equation (18) is based on eight beats; that is, notes should be on the tops of the modified von Mises function which has eight peaks.

4.3 Weight calculation

Let the weight of the ith particle at tth time frame be $w_t^{(i)}$. The weights are calculated using observations and state variables:

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p\left(\omega_t^{(i)}, \theta_t^{(i)} | \omega_{t-1}^{(i)}, \theta_{t-1}^{(i)}\right) p\left(R_t(\omega_t^{(i)}), F_t, r_t | \omega_t^{(i)}, \theta_t^{(i)}\right)}{q\left(\omega_t | \omega_{t-1}^{(i)}, R_t(\omega_t^{(i)}), \omega_{\text{init}}\right) q\left(\theta_t | r_t, F_t, \omega_{t-1}^{(i)}, \theta_{t-1}^{(i)}\right)}. \quad (19)$$

The terms of the numerator in Eq. (19) 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 its weight has with the high probabilities of these functions. 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.

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

$$\omega_t = \omega_{t-1} + n_\omega \quad (20)$$

$$\theta_t = \hat{\Theta}_t + n_\theta, \quad (21)$$

where n_ω denotes the noise of the beat interval distributed with a normal distribution and n_θ denotes the one of the bar-position distributed with a von Mises distribution. Therefore, the state transition model function is expressed as the product of the PDF of these distributions.

$$p\left(\omega_t^{(i)}, \theta_t^{(i)} | \omega_{t-1}^{(i)}, \theta_{t-1}^{(i)}\right) = \text{Mises}(\hat{\Theta}_t, \beta_{n_\theta}, t)\text{Gauss}(\omega_{t-1}, \sigma_{n_\omega}) \quad (22)$$

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 are $w_t^{(i)}$ and -*a*sin

, respectively. The F_t is empirically approximated with the values of the observation as:

$$\begin{aligned} F_t &\approx f(\theta_{\text{beat}_t}, \sigma_f) \\ &\equiv \text{Gauss}(\theta_t^{(i)}; \theta_{\text{beat,t}}, \sigma_f) * rate + bias, \end{aligned} \tag{23}$$

where $\theta_{\text{beat,t}}$ is the bar-position of the nearest beat in the model of eight beats from $\hat{\Theta}_t^{(i)}$. *rate* is a constant value for the maximum of approximated F_t to be 1, and is set to 4. *bias* is uniformly distributed from 0.35 to 0.5. Thus, the observation model function is expressed as the product of these three functions (Eq. (27)).

$$p\left(R_t(\omega_t)|\omega_t^{(i)}\right) = \text{Gauss}\left(\omega_t; \omega_t^{(i)}, \sigma_\omega\right) \tag{24}$$

$$p\left(F_t|\omega_t^{(i)}, \theta_t^{(i)}\right) = \text{Gauss}\left(F_t; f(\theta_{\text{beat,t}}, \sigma_f), \sigma_f\right) \tag{25}$$

$$p\left(r_t|\omega_t^{(i)}, \theta_t^{(i)}\right) = \text{Gauss}\left(r_t; -a\sin(4\hat{\Theta}_t^{(i)}), \sigma_r\right) \tag{26}$$

$$\begin{aligned} &p\left(R_t\left(\omega_t^{(i)}\right), F_t, r_t|\omega_t^{(i)}, \theta_t^{(i)}\right) \\ &= p\left(R_t(\omega_t)|\omega_t^{(i)}\right) p\left(F_t|\omega_t^{(i)}, \theta_t^{(i)}\right) p\left(r_t|\omega_t^{(i)}, \theta_t^{(i)}\right) \end{aligned} \tag{27}$$

We finally estimate the state variables at the tth time frame from the average with the weights of particles.

$$\bar{\omega}_t = \sum_{i=1}^{I} w_t^{(i)} \omega_t^{(i)} \tag{28}$$

$$\bar{\theta}_t = \arctan\left(\sum_{i=1}^{I} w_t^{(i)} \sin\theta_t^{(i)} / \sum_{i=1}^{I} w_t^{(i)} \cos\theta_t^{(i)}\right) \tag{29}$$

Finally we resample the particles to avoid degeneracy; that is, almost all weights become zero except for a few when the weight values satisfy the following equation:

$$\frac{1}{\sum_{i=1}^{I}\left(w_t^{(i)}\right)^2} < N_{th}, \tag{30}$$

where N_{th} is a threshold for resampling and is set to 1.

5 Experiments and results

In this section, we evaluate our beat-tracking system in the following four points:

1. Effect of audiovisual integration based on the particle filter,
2. Effect of the number of particles in the particle filter,
3. Difference between subjects, and
4. Demonstration.

Section 5.1 describes the experimental materials and the parameters used in our method for the experiments. In Section 5.2, we compare the estimation accuracies of our method and Murata's method [3], to evaluate the statistical approach. Since both methods share STPM, the main difference is caused by either the heuristic rule-based approach or statistical one. In addition, we evaluate the effect of adding the visual beat features by comparing with a particle filter using only audio beat features. In Section 5.3, we discuss the relationship between the number of particles versus computational costs and the accuracy of the estimates. In Section 5.4, we present the difference among subjects. In Section 5.5, we give an example of musical robot ensemble with a human guitarist.

5.1 Experimental setup

We asked four guitarists to perform one of each eight kinds of the beat patterns given in Figure 1, at three different tempos (70, 90, and 110), for total of 96 samples. The beat patterns are enumerated in order of beat pattern complexity; a smaller index number indicates that the pattern includes more accented down beats which is easily tracked, while a larger index number indicates that the pattern includes more accented upbeats that confuse the beat-tracker. A performance consists of four counts, seven repetitions of the beat pattern, one whole note and one short note, shown in Figure 8. The average length of each sample was 30.8[sec] for 70 bpm, 24.5 [sec] for 90 bpm and 20.7[sec] for 110. The camera recorded frames at about 19 [fps]. The distance between the robot and a guitarist was about 3 [m] so that the entirety of the guitar could be placed inside the camera frame. We use a one-channel microphone and the sampling parameters shown in Section 3.1 Our method uses 200 particles unless otherwise stated. It was implemented in C++ on a Linux system with an Intel Core2 processor. Table 1 shows the parameters of this experiment. The unit of the parameter relevant to θ is [deg] that ranges from 0 to 360. They all are defined experimentally through a trial and error process.

In order to evaluate the accuracy of beat-tracking methods, we use the following thresholds to define successful beat detection and tempo estimations from ground truth: 150 msec for detected beats and 10 bpm for estimated tempos, respectively.

Two evaluative standard are used, F-measure and AMLc. F-measure is a harmonic mean *of precision* (r_{prec}) and *recall* (r_{recall}) of each pattern. They are calculated by

$$F - \text{measure} = 2/\left(1/r_{\text{prec}} + 1/r_{\text{recall}}\right), \tag{31}$$

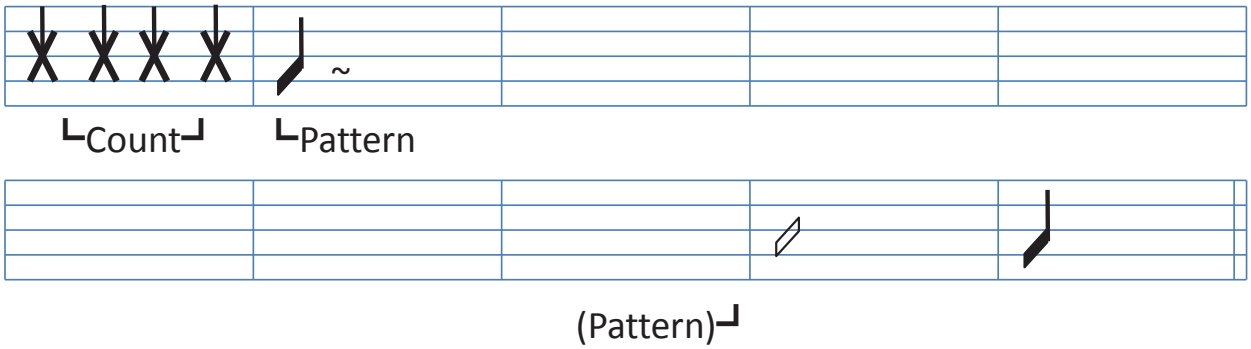

Figure 8 The score used in our experiments. X denotes the counts given by the hit soun from the guitar. White box denotes a whole note. Black box in the last of the score denot a short note.

$$r_{\text{prec}} = N_e/N_d, \tag{32}$$

$$r_{\text{recall}} = N_e/N_c, \tag{33}$$

where N_e, N_d, and N_c correspond to the number of correct estimates, whole estimates and correct beats, respectively. AMLc is the ratio of the longest continuous correctly tracked section to the length of the music, with beats at allowed metrical levels. For example, one inaccuracy in the middle of a piece leads to 50% performance. This represents that the continuity is in correct beat detections and is critical factor in the evaluation of musical ensembles.

The beat detection errors are divided into three classes: substitution, insertion and deletion errors. Substitution error means that a beat is poorly estimated in terms of the tempo or bar-position. Insertion errors and deletion errors are false-positive and false-negative estimations. We assume that a player does not know the other's score, thus one estimates score position by number of beats from the beginning of the performance. Beat insertions or deletions undermine the musical ensemble because the cumulative number of beats should be correct or the performers will lose synchronization. Algorithm 1 shows how to detect inserted and deleted beats. Suppose that a beat-tracker correctly detects two beats with a certain false estimation between them. When the method just incorrectly estimates a beat there, we regard it as a substitution error. In the case of no beat or two beats there, they are counted as a deleted or inserted beats, respectively.

Table 1 Parameter settings: abbreviations are SD for standard deviation, and dist. for distribution

Denotation		Value
Concentration of dist. of sampling θ_t	β_{θ_q}	36,500
Concentration of dist. of θ_t transition	β_{n_θ}	3,650
SD of dist. of ω_{init}	$\sigma_{\omega_{\text{init}}}$	15
SD of dist. of sampling ω_t	σ_{ω_q}	11
SD of dist. of ω_t transition	σ_{n_ω}	1
SD of the approximation of F_t	σ_f	0.2
SD of the observation model of R_t	σ_ω	1
SD of the observation model of r_t	σ_r	2
F_t threshold of beat or noise	*thresh.*	0.7

5.2 Comparison of audiovisual particle filter, audio only particle filter, and Murata's method

Table 2 and Figure 9 summarize the precision, recall and F-measure of each pattern with our audiovisual integrated beat-tracking (**Integrated**), audio only particle filter (**Audio only**) and Murata's method (**Murata**). **Murata** does not show any variance in its result, that is, no error bars in result figures because its estimation is a deterministic algorithm, while the first two plots show

Table 2 Results of the accuracy of beat-tracking estimations

(a) Precision (%)									
Beat Pattern	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Ave.**
Integrated	69.9	75.7	71.1	**65.1**	**48.3**	**46.8**	**74.0**	40.1	**61.4**
Audio only	43.6	46.6	45.6	28.7	24.7	18.1	43.6	**41.5**	36.5
Murata	**86.3**	**82.4**	**83.2**	44.1	39.9	22.4	25.5	22.3	50.8
(b) Recall (%)									
Beat Pattern	1	2	3	4	5	6	7	8	Ave.
Integrated	70.3	75.8	71.9	**66.0**	**47.7**	**45.6**	**74.5**	**39.7**	**61.4**
Audio only	40.8	43.9	42.5	28.7	23.4	17.9	41.6	38.8	34.7
Murata	**89.6**	**87.1**	**87.0**	48.8	43.7	26.7	27.2	24.4	54.3
(c) F-measure (%)									
Beat Pattern	1	2	3	4	5	6	7	8	Ave.
Integrated	70.1	75.7	71.5	**65.5**	**48.0**	**46.1**	**74.2**	39.9	**61.4**
Audio only	42.2	45.2	44.0	28.7	24.0	18.0	42.6	**40.1**	35.6
Murata	**87.9**	**84.7**	**85.1**	46.3	41.7	24.3	26.3	23.3	52.5

Bold numbers represent the largest results for each beat pattern.

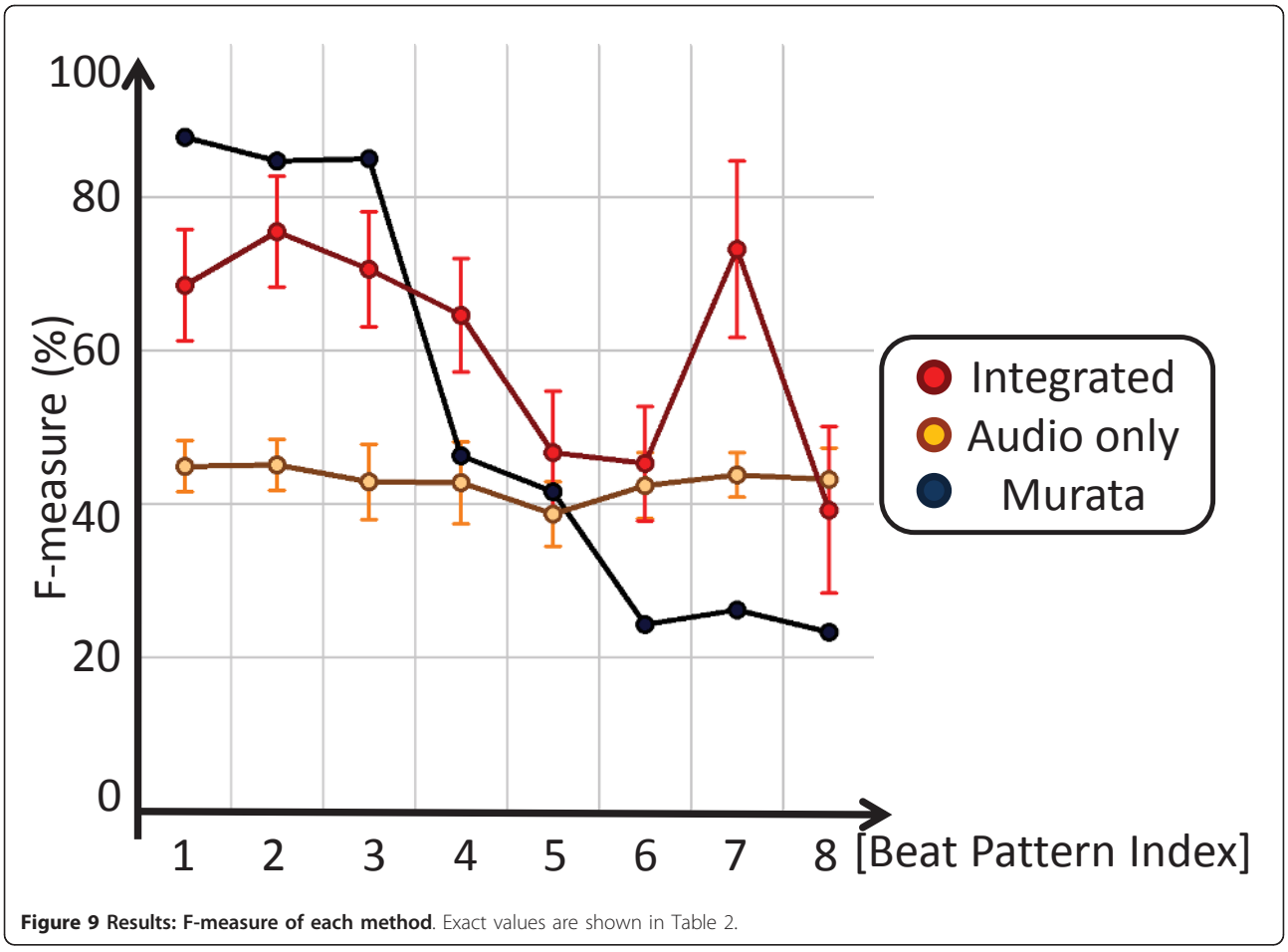

Figure 9 Results: F-measure of each method. Exact values are shown in Table 2.

variance due to the stochastic nature of particle filters. Our method **Integrated** stably produces moderate results and outperforms **Murata** for patterns 4-8. These patterns are rather complex with syncopations and downbeat absences. This demonstrates that **Integrated** is more robust against beat patterns than **Murata**. The comparison between **Integrated** and **Audio only** confirms that the visual beat features improve the beat-tracking performance; **Integrated** improves precision, recall, and F-measure by 24.9, 26.7, and 25.8 points in average from **Audio only**, respectively.

The F-measure scores of the patterns 5, 6, and 8 decrease for **Integrated**. The following mismatch causes this degradation; though these patterns contain sixteenth beats that make the hand move at double speed, our method assumes that the hand always moves downward only at quarter note positions as Eq. (9) indicates. To cope with this problem, we should allow for downward arm motions at eighth notes, that is, sixteen beats. However, a naive extension of the method would result in degraded performances with other patterns.

The average of F-measure for **Integrated** shows about 61%. The score is deteriorated due to these two reasons: (1) the hand's trajectory model does not match the sixteen-beat patterns, and (2) the low resolution and the error in estimating visual beat feature extraction do not make the *penalty* function effective in modifying the θ distribution.

Table 3 and Figure 10 present the AMLc comparison among the three method. As well as the F-measure result, **Integrated** is superior to **Murata** for patterns 4-8. The AMLc results of patterns 1 and 3 are not so high despite the high F-measure score. Here, we define *result rate* as the ratio of the AMLc score to the F-measure one. In patterns 1 and 3, the result rates are not so high, 72.7 and 70.8. Likewise the F-measure results, the result rates of patterns 4 and 5 remark lower scores, 48.9 and 55.8. On the other hand, the result rates of patterns 2 and 7 show

Table 3 Results of AMLc

Beat Pattern	1	2	3	4	5	6	7	8	Ave.
Integrated	49.9	64.2	50.0	**43.6**	**22.8**	**25.3**	**54.8**	**26.5**	**42.1**
Audio only	18.6	18.0	17.6	16.8	14.7	18.5	18.3	16.6	17.4
Murata	**84.2**	**68.9**	**78.6**	24.1	11.0	8.4	19.4	16.9	38.9

Bold numbers represent the largest results for each beat pattern.

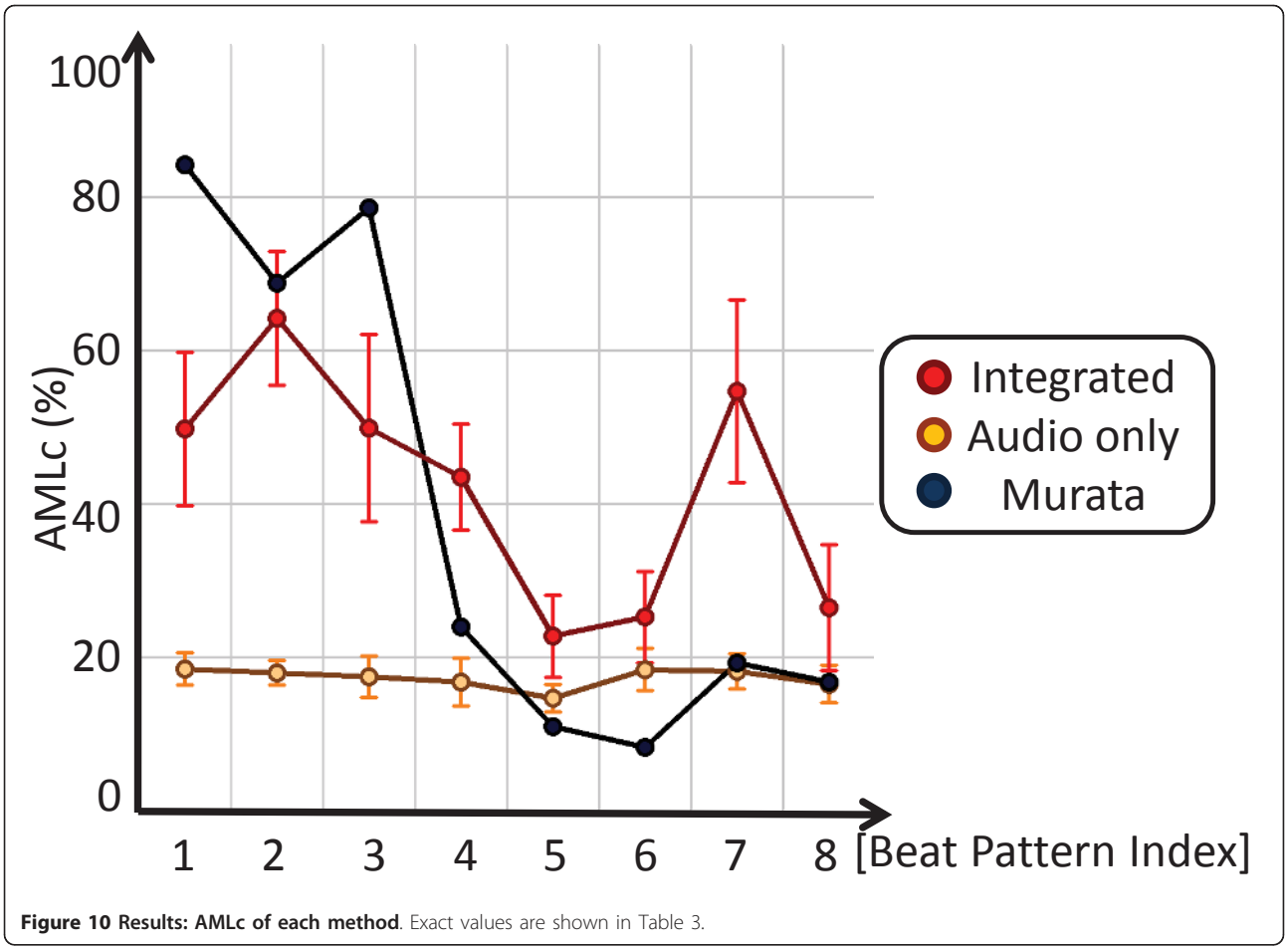

Figure 10 Results: AMLc of each method. Exact values are shown in Table 3.

still high percentage as 85.0 and 74.7. The hand's trajectory of patterns 2 and 7 is approximately the same with our model, a sign curve. In pattern 3, however, the triplet notes affect the trajectory to be late in the upward movement. In pattern 1, no upbeats, that is, no constraints in the upward movement allow the hand to move loosely upward in comparison with the trajectories in other patterns. To conclude, the result rate has a relationship with the similarity of a hand's trajectory of each pattern with our model. The model should be refined to raise scores in our future work.

In Figure 11, **Integrated** demonstrates less errors than **Murata** with regard to the total errors of insertions and deletions. A detailed analysis shows that **Integrated** has less deletion errors than **Murata** in some patterns. On the other hand, **Integrated** has more insertion errors than **Murata**, especially in sixteen beats. However, the adaption to sixteen beats would produce fewer insertions in **Integrated**.

5.3 The influence of the number of particles

As a criterion of the computational cost, we use a real-time factor to evaluate our system in terms of a real-time system. The real-time factor is defined as computation time divided by data length; for example, when the system takes 0.5s to process 2 s data, the real-time factor is 0.5/2 = 0.25. The realtime factor must be less than 1 to run the system in real-time. Table 4 shows the real-time factors with various numbers of particles. The real-time factor increases in proportion to the number of particles. The real-time factor is kept under 1 with 300 particles or less. We therefore conclude that our method works well as a real-time system with fewer than 300 particles.

Table 4 also shows that the F-measures differ by only about 1.3% between 400 particles showing the maximum result and 200 particles where the system works in real-time. This suggests that our system is capable of real-time processing with almost saturated performance.

5.4 Results with various subjects

Figure 12 indicates that we can observe only little difference among the subjects except Subject 3. In the case of Subject 3, the similarity of the skin color to the guitar caused frequent loss of the hand's trajectory. To improve the estimation accuracy, we should tune the

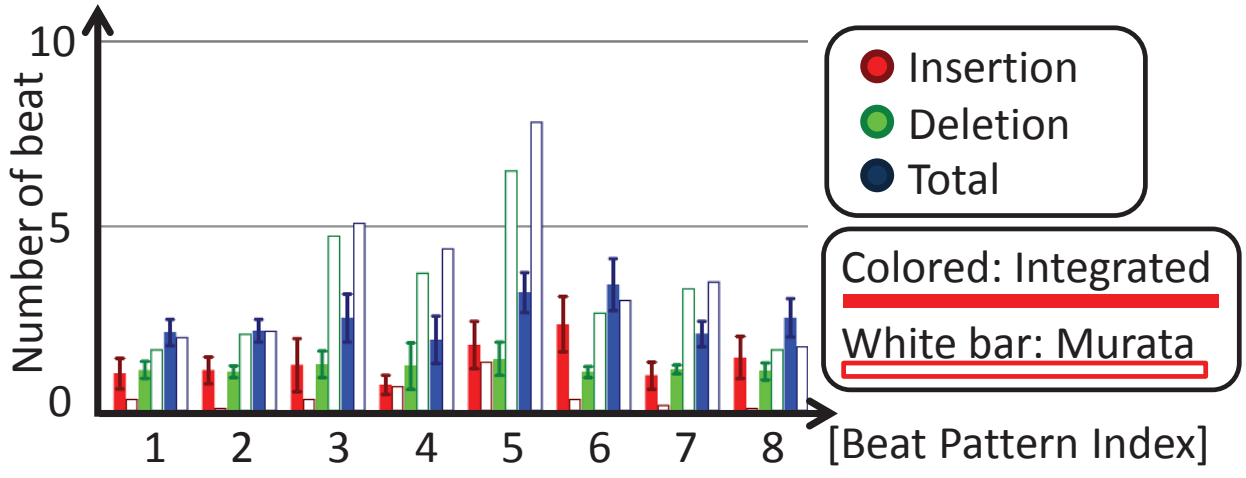

Figure 11 Results: Number of inserted and deleted beats.

algorithm or parameters to be more robust against such confusion.

5.5 Evaluation using a robot

Our system was implemented on a humanoid robot HRP-2 that plays an electronic instrument called the theremin as in Figure 13. The video is available on Youtube [26]. The humanoid robot HRP-2 plays the theremin with a feed-forward motion control developed by Mizumoto et al. [27]. HRP-2 captures a mixture of sound consisting of its own theremin performance and human partner's guitar performance with its microphones. HRP-2 first suppresses its own theremin sounds by using the semi-blind ICA [28] to obtain the audio signal played by the human guitarist. Then, our beat-tracker estimates the tempo of the human performance and predicts the tactus. According to the predicted tactus, HRP-2 plays the theremin. Needless to say, this prediction is coordinated to absorb the delay of the actual movement of the arm.

6 Conclusions and future works

We presented an audiovisual integration method for beat-tracking of live guitar performances using a particle filter. Beat-tracking of guitar performances has three following problems: tempo fluctuation, beat pattern complexity and environmental noise. The auditory beat features are the autocorrelation of the onsets and the onset summation extracted with a noise-robust beat estimation method, called STPM. The visual beat feature is the distance of the hand position from the guitar neck, extracted with the optical flow and mean shift and by Hough line detection, respectively. We modeled the stroke and the beat location based on an eight-beat assumption to address the single instrument situation. Experimental results show the robustness of our method against such problems. The F-measure of beat-tracking estimation improves by 8.9 points on average compared with an existing beat-tracking method. Furthermore, we confirmed that our method is capable of real-time processing by suppressing the number of particles while preserving beat-tracking accuracy. In addition, we demonstrate a musical robot ensemble with a human guitarist.

Table 4 Influence of the number of particles on the estimation accuracy and computational speed

Number of particles	50	100	200	300	400
Real-time factor	0.18	0.33	0.64	0.94	1.25
Precision (%)	57.7	59.7	61.4	62.2	62.5
Recall (%)	57.0	59.5	61.4	62.4	62.9
F-measure (%)	57.3	59.6	61.4	62.3	62.7

We still have two main problems to improve the quality of synchronized musical ensembles: beat-tracking with higher accuracy and robustness against estimation errors. For the first problem, we have to get rid of the assumption of quadruple rhythm and eight beats. The hand-tracking method should be also refined. One possible way for improved hand tracking is the use of infrared sensors that are recently gathering many researchers' interest. In fact, our preliminary experiments suggest that the use of an infrared sensor instead of an RGB camera would enable more robust hand tracking. Thus, we can also expect an improvement of the beat-tracking itself by using this sensor.

We suggest two extensions as future works to increase robustness to estimation errors: audio-to-score alignment with reduced score information, and the beat-tracking with prior knowledge of rhythm patterns.

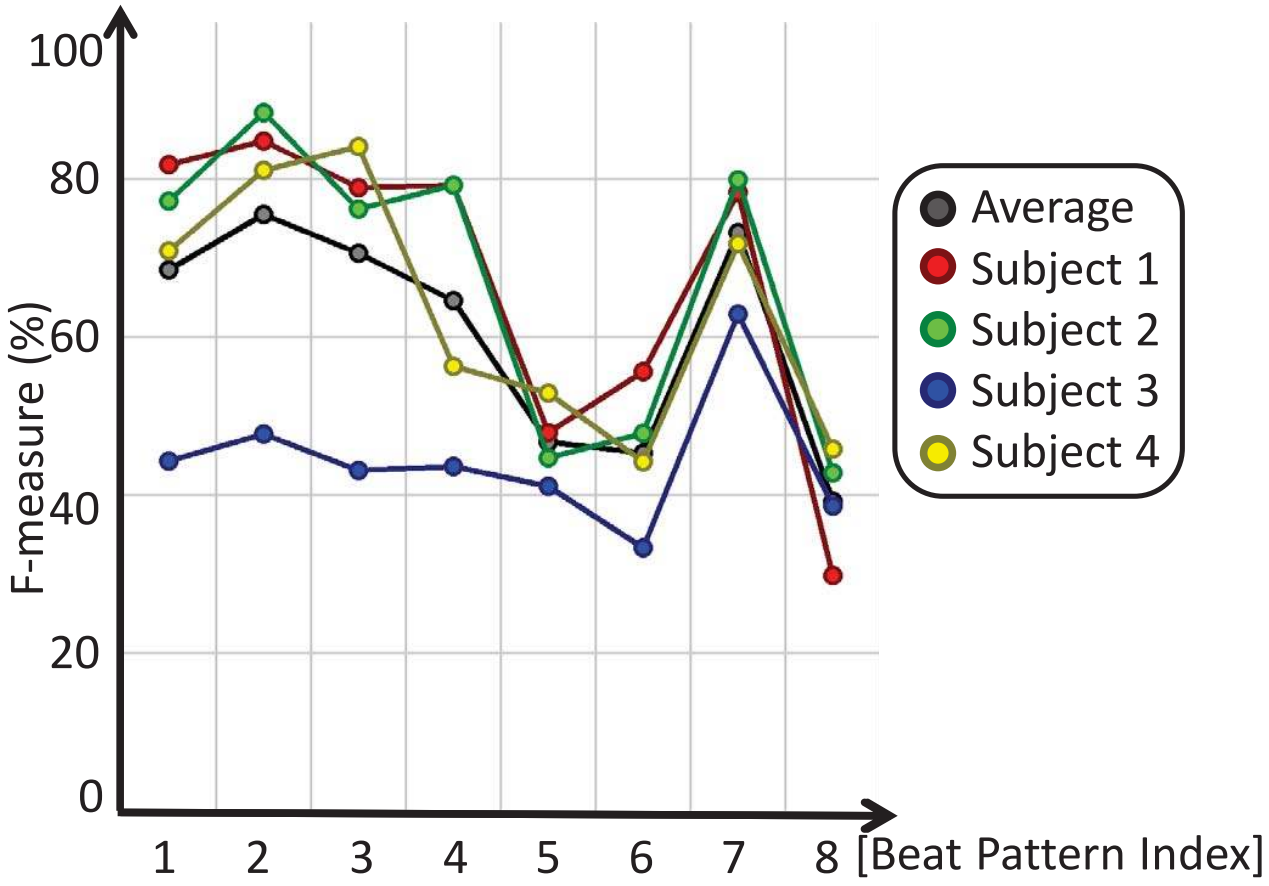

Figure 12 Comparison among the subjects.

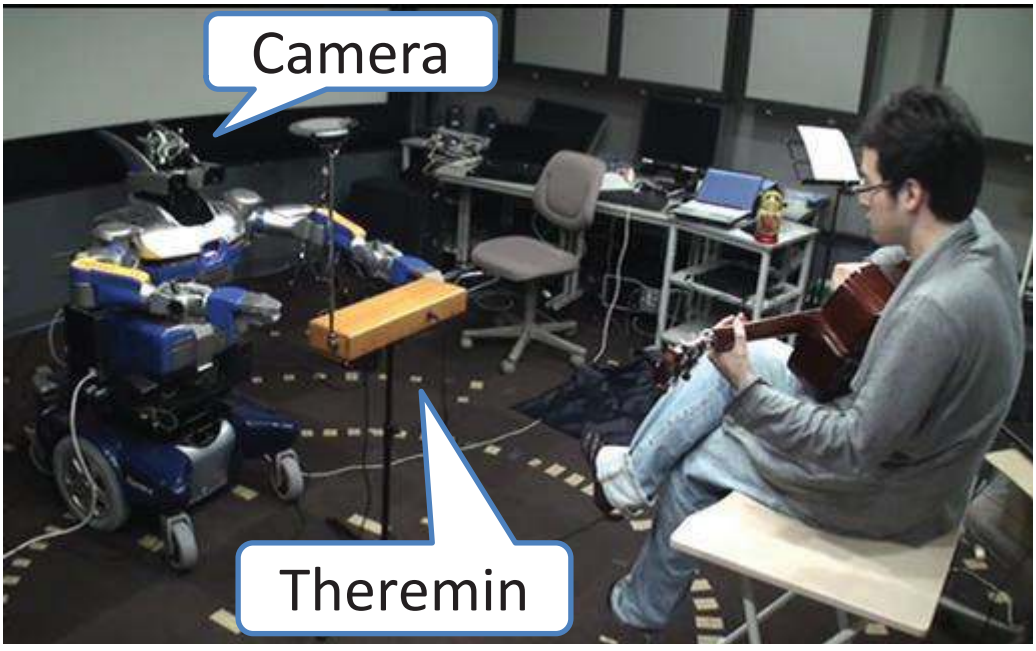

Figure 13 An example image of musical robot ensemble with a human guitarist.

While standard audio-to-score alignment methods [12] require a full set of musical notes to be played, for example, an eighth note of F in the 4th octave and a quarter note of C in the 4th octave, guitarists use scores with only the melody and chord names, with some ambiguity with regard to the octave or note lengths. Compared to beat-tracking, this melody information would allow us to be aware of the score position at the bar level and to follow the music more robustly against insertion or deletion errors. The prior distribution of rhythm patterns can also alleviate the insertion or deletion problem by forming a distribution of possible beat positions in advance. This kind of distribution is expected to result in more precise sampling or state transition in particle-filter methods. Finally, we have to remark that we need the subjective evaluation as to how much our beat-tracking improves the quality of the human-robot musical ensemble.

Algorithm 1 Detection of inserted and deleted beats

deleted ← 0 {*deleted* denotes the number of deleted beats}
inserted ← 0 {*inserted* denotes the number of inserted beats}
prev_index ← 0
for all *detected_beat* **do**
  if |tempo(*detected_beat*)-tempo(*ground_truth_beat*)| < 10
  and | beat_time(*detected_beat*)-*beat_time*(*ground_truth_beat*)| < 150 **then**
    {*detected_beat* is correct estimation}
    new_index ← index(*ground_truth_beat*)
    N ← (*new_index* - *prev_index* - 1) - *error_count*
    deleted ← *deleted* + MAX(0, *N*)
    inserted ← *inserted* + MAX(0, -*N*)
    prev_index ← new_index
    error_count ← 0
  else
    error_count ← *error_count* + 1
  end if
end for

Acknowledgements
This research was supported in part of by a JSPS Grant-in-Aid for Scientific Research (S) and in part by Kyoto University's Global COE.

Competing interests
The authors declare that they have no competing interests.

Received: 16 April 2011 Accepted: 20 January 2012
Published: 20 January 2012

References

1. A Klapuri, A Eronen, J Astola, Analysis of the meter of acoustic musical signals. IEEE Trans Audio Speech Lang Process. **14**, 342–355 (2006)
2. G Weinberg, B Blosser, T Mallikarjuna, A Raman, The creation of a multi-human, multi-robot interactive jam session, in *Proc of Int'l Conf on New Interfaces of Musical Expression*, pp. 70–73 (2009)
3. K Murata, K Nakadai, R Takeda, HG Okuno, T Torii, Y Hasegawa, H Tsujino, A beat-tracking robot for human-robot interaction and its evaluation, in *Proc of IEEE/RAS Int'l Conf on Humanoids (IEEE)*, pp. 79–84 (2008)
4. T Mizumoto, A Lim, T Otsuka, K Nakadai, T Takahashi, T Ogata, HG Okuno, Integration of flutist gesture recognition and beat-tracking for human-robot ensemble, in *Proc of IEEE/RSJ-2010 Workshop on Robots and Musical Expression*, pp. 159–171 (2010)
5. A Rosenfeld, A Kak, in *Digital Picture Processing*, vol. 1 & 2. (Academic Press, New York, 1982)
6. G Ince, K Nakadai, T Rodemann, Y Hasegawa, H Tsu-jino, J Imura, A hybrid framework for ego noise cancellation of a robot, in *Proc of IEEE Int'l Conf on Robotics and Automation (IEEE)*, pp. 3623–3628 (2011)
7. S Dixon, E Cambouropoulos, Beat-tracking with musical knowledge, in *Proc of European Conf on Artificial Intelligence*, pp. 626–630 (2000)
8. M Goto, An audio-based real-time beat-tracking system for music with or without drum-sounds. J New Music Res. **30**(2), 159–171 (2001). doi:10.1076/jnmr.30.2.159.7114
9. AT Cemgil, B Kappen, Integrating tempo tracking and quantization using particle filtering, in *Proc of Int'l Computer Music Conf*, pp. 419 (2002)
10. N Whiteley, AT Cemgil, S Godsill, Bayesian modelling of temporal structure in musical audio, in *Proc of Int'l Conf on Music Information Retrieval*, pp. 29–34 (2006)
11. S Hainsworth, M Macleod, Beat-tracking with particle filtering algorithms, in *Proc of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE)*, pp. 91–94 (2003)
12. T Otsuka, K Nakadai, T Takahashi, K Komatani, T Ogata, HG Okuno, Design and Implementation of Two-level Synchronization for Interactive Music Robot, in *Proc of AAAI Conference on Artificial Intelligence*, pp. 1238–1244 (2010)
13. Y Pan, MG Kim, K Suzuki, A robot musician interacting with a human partner through initiative exchange, in *Proc of Conf on New Interfaces for Musical Expression*, pp. 166–169 (2010)
14. K Petersen, J Solis, A Takanishi, Development of a realtime 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*, pp. 313–318 (2008)
15. A Lim, T Mizumoto, L Cahier, T Otsuka, T Takahashi, K Komatani, T Ogata, HG Okuno, Robot musical accompaniment: integrating audio and visual cues for realtime synchronization with a human flutist, in *Proc of IEEE/RSJ Int'l Conf on Intelligent Robots and Systems*, pp. 1964–1969 (2010)
16. D Comaniciu, P Meer, Mean shift: A robust approach toward feature space analysis, in *Proc of IEEE Transactions on pattern analysis and machine intelligence*, IEEE Computer Society, pp. 603–619 (2002)
17. K Fukunaga, *Introduction to Statistical Pattern Recognition*, (Academic Press, New York, 1990)
18. R Kalman, A new approach to linear filtering and prediction problems. J Basic Eng. **82**, 35–45 (1960). doi:10.1115/1.3662552
19. EH Sorenson, Kalman Filtering: Theory and Application, (IEEE Press, New York, 1985)
20. K Nickel, T Gehrig, R Stiefelhagen, J McDonough, A joint particle filter for audio-visual speaker tracking, in *Proc of Int'l Conf on multimodal interfaces*, pp. 61–68 (2005)
21. BD Lucas, T Kanade, An iterative image registration technique with an application to stereo vision, in *Proc of Int'l Joint Conf on Artificial Intelligence*, pp. 674–679 (1981)
22. D Miyazaki, RT Tan, K Hara, K Ikeuchi, Polarization-based inverse rendering from a single view, in *Proc of IEEE Int'l Conf on Computer Vision*, pp. 982–987 (2003)
23. DH Ballard, Generalizing the Hough transform to detect arbitrary shapes. Pattern recognition. **13**(2), 111–122 (1981). doi:10.1016/0031-3203(81)90009-1
24. M Fischler, R Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM. **24**(6), 381–395 (1981). doi:10.1145/358669.358692
25. R von Mises, Uber dir "Ganzzahligkeit" der Atom-gewichte und verwandte Fragen. Phys Z. **19**, 490–500 (1918)
26. T Itohara, HRP-2 follows the guitar. http://www.youtube.com/watch?v=-fuOdhMeF3Y

27. T Mizumoto, T Otsuka, K Nakadai, T Takahashi, K Komatani, T Ogata, HG Okuno, Human-robot ensemble between robot thereminist and human percussionist using coupled oscillator model, in *Proc of IEEE/RSJ Int'l Conf on Intelligent Robots and Systems (IEEE)*, pp. 1957–1963 (2010)
28. R Takeda, K Nakadai, K Komatani, T Ogata, HG Okuno, Exploiting known sound source signals to improve ICA-based robot audition in speech separation and recognition, in *Proc of IEEE/RSJ Int'l Conf on Intelligent Robots and Systems*, pp. 1757–1762 (2007)

doi:10.1186/1687-4722-2012-6
Cite this article as: Itohara *et al.*: A multimodal tempo and beat-tracking system based on audiovisual information from live guitar performances. *EURASIP Journal on Audio, Speech, and Music Processing* 2012 **2012**:6.