The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Incremental Polyphonic Audio to Score Alignment using Beat Tracking for Singer Robots

Takuma Otsuka, Kazumasa Murata, Kazuhiro Nakadai, Toru Takahashi,
Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno

Abstract— **We aim at developing a singer robot capable of listening to music with its own "ears" and interacting with a human's musical performance. Such a singer robot requires at least three functions: listening to the music, understanding what position in the music is being performed, and generating a singing voice. In this paper, we focus on the second function, that is, the capability to align an audio signal to its musical score represented symbolically. Issues underlying the score alignment problem are: (1) diversity in the sounds of various musical instruments, (2) difference between the audio signal and the musical score, (3) fluctuation in tempo of the musical performance. Our solutions to these issues are as follows: (1) the design of features based on a chroma vector in the 12-tone model and onset of the sound, (2) defining the *rareness* for each tone based on the idea that scarcely used tone is salient in the audio signal, and (3) the use of a switching Kalman filter for robust tempo estimation. The experimental result shows that our score alignment method improves the average of cumulative absolute errors in score alignment by 29% using 100 popular music tunes compared to the beat tracking without score alignment.**

I. Introduction

Robots are expected to become more involved in human society thanks to remarkable developments in their physical functions. For example, housework or nursing robots are being developed and tested to help people or caretakers. For symbiosis between humans and robots in everyday situations, robots need not only advanced physical functions but also the ability to interact naturally with humans.

Among many possible kinds of interactions, we focus especially on interactions between humans and robots through music. This is because music plays an important role in human cultures. Even people who do not share a language can share a friendly and joyful time through music although natural communications by other means are difficult. Therefore, *music robots* that can interact with humans through music are essential for robots to live in harmony with humans.

The objective of our research is to develop a singer robot that can sing with accompaniments or with human singers. Singing is the most basic means of musical expressions for humans. This is also an effective type of musical expression for robots. Robots are able to make expressive sounds in a variety of ways such as changing the volume or the timbre of their singing voices. Furthermore, robots can make expressive physical body motions, like squaring their torsos while singing loudly. The type of musical interactions we are concerned with is explained in the following examples. A man plays a musical instrument while the robot sings along. When the man plays faster, the robot's singing gets faster accordingly. When the music reaches the exciting chorus part, the robot sings louder and moves its body more actively and the man responds to the robot's singing and enlivens his performance. Our ultimate goal is a singing robot capable of these rich interactions.

The singing robot we are envisioning consists of three main functions. The first one is the capability to listen to music. The robot should actually listen to the music with its own "ears", that is, microphones. "Hearing" music with humans the same way they do is necessary for musical interaction. Generally, the sound that the robot hears is mixed with music and self-generated sounds such as motor sounds or the robot's own singing voice. The robot therefore has to extract the sound of music from the mixed sound. Second, the robot has to understand what position of the music is being played to sing the correct melody and words. Finally, the robot should sing in such a way that its voice matches the music. The robot synthesizes a singing voice and may move its body for vivid musical expression.

This paper addresses the second function. We believe the robot should be able to sing along with various musical performances. Musical performances can vary in the types of musical instruments, the tempo, the musical key, or even an arrangement of the melody. To understand what position in the music is performed robustly, a solution to refer to a prior-recorded audio signal is not acceptable. This is because the signal can be different for each performance. Therefore, we give the musical score in symbolic representation to the robot so that the robot can estimate the position of the music allowing for the variation of musical instruments and the tempo.

II. System architecture and issues of score alignment for singer robots

This section explains the singer robot's architecture and the issues to realize the singer robot.

T. Otsuka, T. Takahashi, K. Komatani, T. Ogata, H. G. Okuno are with Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan. {ohtsuka, tall, komatani, ogata, okuno}@kuis.kyoto-u.ac.jp

K. Murata is with Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Tokyo, 152-8552, Japan. murata@cyb.mei.titech.ac.jp

K. Nakadai is with Honda Research Institute Japan, Co., Ltd., Wako, Saitama, 351-0114, Japan, and also with Graduate School of Information Science and Engineering, Tokyo Institute of Technology. nakadai@jp.honda-ri.com

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

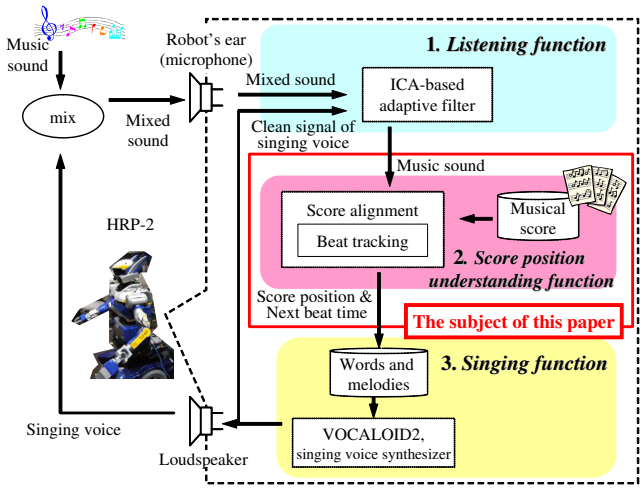

Fig. 1. Singer robot architecture

A. The Architecture of The Singer Robot

The architecture of the singer robot is depicted in Figure 1. This is an extension of the beat tracking based singer robot developed by Murata *et al.* [1]. This architecture has three essential functions: 1) *listening*, 2) *score position understanding* and 3) *singing function*. We explain each function along with Figure 1.

1) The *listening function* achieves the separation of music sound. When the robot is singing, the music the robot "hears" is a mixture of the target music and its own singing voice. The robot extracts the music sound because singing voice can impede the following music recognition such as beat tracking as reported in [1], [2]. The echo cancellation techniques are available for the separation of the music sound and singing voice. We use the independent-component-analysis (ICA)-based adaptive filter [3] since this method realizes a robust incremental separation.

2) After the separation, the *score position understanding function* estimates the current score position of the music sound and predicts the next beat time. Here, "beat" means the position of the quarter notes in the musical score. The estimation of the score position is critical for the singer robot because it cannot sing the appropriate word and the melody without the knowledge of the current score position. The beat tracking method is only used for this function in Murata's singer robot. This results in an inharmonious performance since the false detection of the beat causes a time-lag between the robot's singing and the music. The prediction of the next beat time is necessary because singing voice generation takes a while.

3) The *singing function* selects the appropriate melody and word according to the output of the *score position understanding function*. The singing voice is synthesized with VOCALOID2 [4] developed by YAMAHA.

B. Problem Statement

The situation where a singer robot and humans interact through music is described as follows: they enjoy the music in a casual way. They are in an ordinary room rather than a concert hall. The type of music is pop or folk music that is familiar to most people.

The type of musical instruments are unknown to the robot. Humans may play the piano, guitar or other instruments. The music may be an ensemble using multiple instruments. The only thing that is available to the robot is the musical score of the performance.

The musical score is incomplete in the following sense. First, the tempo is undetermined in the score because the tempo tends to vary when humans perform a musical piece. Second, percussions or drums are not necessarily played as written in the score. For example, the drummer may play the drums in a complex pattern or people may clap their hands simply on each beat.

The score alignment problem becomes even harder if it is unknown which part the music starts from. Since the robot sings to humans' music performance, they can determine beforehand where the music starts. This paper discuss the score alignment under the condition that the music starts at the beginning of the score this time.

Here let us describe the problem.

Score alignment problem for singer robots

Input: music audio signal and the corresponding musical score in standard MIDI file,
Output: the score position currently performed,
Assumptions:
1) no prior knowledge as to what instruments are used,
2) the tempo of input music is known,
3) the score for percussion is unknown, and
4) the music starts at the beginning of the score.

The score alignment for singer robots need to satisfy the following requirements in terms of the implementation.
Requirement 1) incremental processing,
Requirement 2) low computational cost.

C. Three Issues and Our Solutions

We break down the score alignment in question into the following three issues and present our solutions.

a) Diversity in the sound of various instruments: The timbre of sound differs in both (1) frequency domain and (2) time domain as shown in Figure 2 and 3, respectively. These figures show a single note on the piano at left and on the flute at right.
(1) Figure 2 indicates that although these two are the same note A4 with fundamental frequency 440 [Hz], the shape of the spectrum for each sound is different.
(2) Figure 3 shows the power envelope in time domain of each note. The power envelope of instrument sounds generally consists of *attack*, *decay*, *sustain* and *release*. Some instruments such as piano or guitar have declining sustain, while instruments such as flute, violin or saxophone have persisting sustain.

When multiple notes are performed by various instruments at the same time, in other words, dealing with a polyphonic

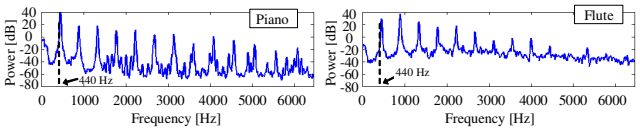

Fig. 2. The spectrum of a single note in frequency domain

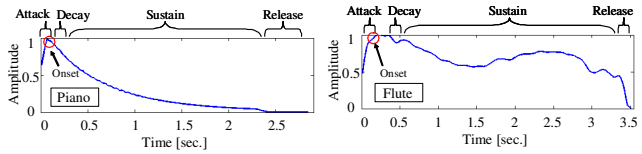

Fig. 3. The power envelope of a single note in time domain

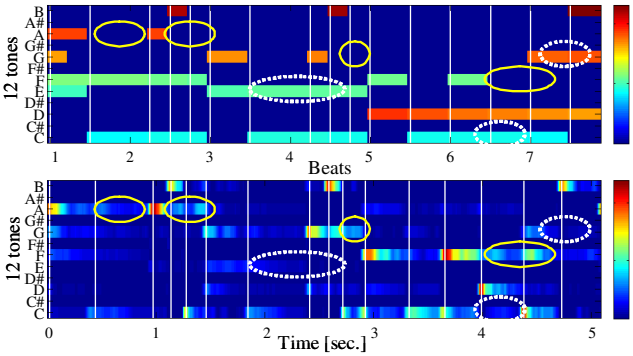

Fig. 4. Difference in chroma vector between the audio signal and the musical score (Top: the musical score, Bottom: the audio signal). The color represents *rareness* on the top, and the power of the signal on the bottom.

audio, detecting the fundamental frequency of each note or recognizing the sustaining sounds becomes even more difficult.

We use a 12-dimensional chroma vector as frequency domain feature, and detect onset times as time domain features. The merit of the chroma vector is the robustness to the variety in the spectral shape of various instruments and the availability in a polyphonic audio signal. The chroma vector extracts each power of 12 pitch names, that is, C, C♯, ..., B, instead of the fundamental frequency. The onset of each note in this paper is defined as the peak near from the steep increase in the power shown as Figure 3. We extract the onsets for two reasons: (a) we need to obtain the time when each note starts for the score alignment, and (b) onsets are more easily extracted as the increase of the power in time domain than sustains or releases especially in a polyphonic audio signal.

b) Difference between the audio signal and the musical score: Figure 4 shows the musical score at top and the audio signal at bottom in chroma vector sequence. The vertical white lines indicate onsets of musical notes. Onsets in the musical score is defined as the starting frame of each note. The color in the musical score indicates the rareness explained below and the color in the audio signal indicates the power of each tone.

The audio signal of actual performance differs from the musical score in chroma vector representation in these ways:
(1) The power of previous sound persists in solid line circles although no note appears in the musical score,
(2) the musical score has notes while the power of signals is hardly observed in dotted circles, and
(3) the volume of each note is not specified by the score.

We alleviate the difference between the audio signal and the musical score based on the idea that musical notes in scarcely used pitch name are often outstanding in the audio signal. We can preprocess the musical score because the musical score is available before we acquire the audio signal. Therefore, we define our original feature, rareness, for each pitch name in the musical score. The definition of the rareness is analogous to information entropy. In Figure 4, the rareness for B is high because the number of B notes is fewer than that of the other pitch notes. By contrast, the rareness for C or E is low since these notes are frequently used in the score. Each tone is weighted by its rareness. Thus, infrequent notes can be extracted more easily from a polyphonic audio signal than frequently used notes.

c) Fluctuation in tempo: Stable tempo estimation is essential not only for correct score alignment but also for the robot to produce a smooth and mellifluous singing voice. Fluctuations in tempo can be observed in (1) actual performance by human and (2) estimated tempo by beat tracking [1].
(1) The speed, or tempo, of human musical performance changes as shown in Figure 5. The plots in Figure 5 are calculated from MIDI data strictly aligned with human performance. Each tempo is obtained by dividing the length of the note in the musical score by the length in time.
(2) Figure 6 shows the fluctuation in the beat tracking. The sequence of tempo includes not a few outliers. Outliers are typically caused by change in the drumming pattern.

We adopt a switching Kalman filter (SKF) for stable tempo estimation. SKF enables incremental tempo estimation from the tempo sequence including errors.

D. Another Issue in Application to Singer Robots

When music is played in a room surrounded by walls, the music sound inevitably includes direct reflections and reverberation. These can affect both onset detection and the shape of chroma vectors. Sirect reflections can cause false onset detections even if the music is played exactly the same way as written in the score. In general, we can alleviate the effect of direct reflections by analyzing the music signal with low time resolution. For example, we can set a large shift interval for short-time Fourier transform. However, this solution results in an inaccurate score alignment in terms of time accuracy.

Reverberation causes difference in the audio signal from the musical score since it prolongs the shape of chroma vector sequence. This reverberation is not written in the musical score because the reverberation depends on the environment where the robot and music players exist.

We set 11 [msec] long shift interval for short-time Fourier transform to ensure the time accuracy. We evaluate the effect of reverberation on the score alignment in Section IV.

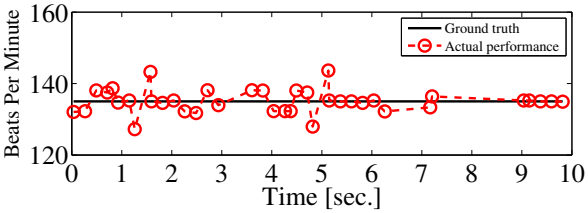

Fig. 5. Fluctuation in the tempo of actual performance

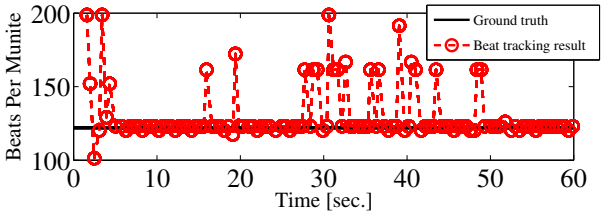

Fig. 6. Fluctuation in tempo estimation by the beat tracking

E. Related Studies

This section describes two kinds of related studies. One is related to music robots and the other is related to score alignment methods.

1) Existing Music Robots: Robots that moves their body to music, *music robots*, have interested many researchers. Several music robots have been reported so far such as dancing robots [5], [6]. These robots, however, focus on generating appropriate motion and ignore the music sounds which are essential for natural interactions through music. We can expect an advanced robot dance by giving them the ability to listen to music.

Although robots that actually listen to music have been reported, these robots pay attention to only beat structures in music audio signal. Therefore, these robots are able to simply perform repetitive motions, such as playing the percussion, stepping, or scatting. Weinberg *et al.* developed a percussionist robot named Haile [7], and a robot named Simon [8] that plays the marimba with their own arms. These robots improvises their performance by genetic algorithm. Although they extract a beat structure from an audio signal of human performance such as the rhythmic stability, they obtain pitch information through the MIDI communication in symbolic representation, not in an acoustic manner. Yoshii *et al.* applied the real-time beat tracking method invented by Goto [9] to Honda's ASIMO. This robot extracts the beat structure from the music audio signal and steps to the music. When the music tempo changes, the robot also changes the stepping speed accordingly. Mizumoto *et al.* applied Yoshii's real-time beat tracking and an adaptive filter based on independent-component-analysis developed by Takeda *et al.* [3] to Robovie-R2 so that it can count musical beats as "one, two, three, four" [2]. They enabled the robot to count beats successfully by suppressing its own voice with the adaptive filter. Murata *et al.* developed another beat tracking method based on spectro-temporal pattern matching. They applied this beat tracking method and Takeda's adaptive filter to Honda's ASIMO [1]. This robot steps and makes a scatting voice to the music and sings according to the musical beats.

However, the beat tracking has an unsolved problem when it is applied to a singer robot system. The problem with Murata's system in singing is that this robot counts the number of beats from the beginning of the music to decide what to sing. With the beat tracking based estimation of the music position, it is difficult to recover from a wrong estimation if it fails to extract beat time or the correct tempo.

Our work can contribute to music robot's capability other than singing. Giving robots a musical score and enabling them to align the music audio signal with the musical score is an effective way to enable the robots to perform along with the music. For example, in addition to singing, robots may dance to music whose dancing motion is choreographed in advance or they may play a certain melody on a musical instrument with human players.

2) Existing Score Alignment Methods: In the music information processing field, there have been several studies related to score alignment [10]–[12]. However, the existing studies have such constraints as batch processing [10] or assumptions that the music is played on a single instrument [11] or that the instrument used in the music is given [12]. These assumptions are undesirable for singer robots in our situation.

III. Score Alignment Method

This section describes our score alignment method. Figure 7 shows the overview of the method. This score alignment consists of four main parts:

1) Feature extraction from the audio signal and the musical score,
2) beat interval (tempo) calculation by beat tracking,
3) incremental matching between audio signal feature and score feature in chroma vectors,
4) tempo estimation with Switching Kalman Filters using beat tracking result and matching results.

The input is the audio signal and the musical score. The output is the score position currently performed and the predicted next beat time. The score position is determined by the matching described in section III-D and the next time in the audio signal is the output of Kalman Filter described in section III-E.

A. Feature Extraction from the Audio Signal

Two features are extracted from the audio signal. One is a chroma vector and the other is onset times. Onset is the starting point of each musical note. Table I shows the indices used in the following equations.

1) Chroma vector generation: The system first obtains the spectrogram of a music audio signal by applying the short-time Fourier transform (STFT). STFT is calculated with a Hanning window of 4096 [points], a shifting interval of 512 [points] and sampling rate of 44.1 [kHz]. Let $p(t,\omega)$ be the power at time frame t and frequency bin ω. Chroma vector $\mathbf{c}(t) = [c(1,t), c(2,t), ..., c(12,t)]^T$ (T means the transpose of the vector) is generated for each time frame t. Each

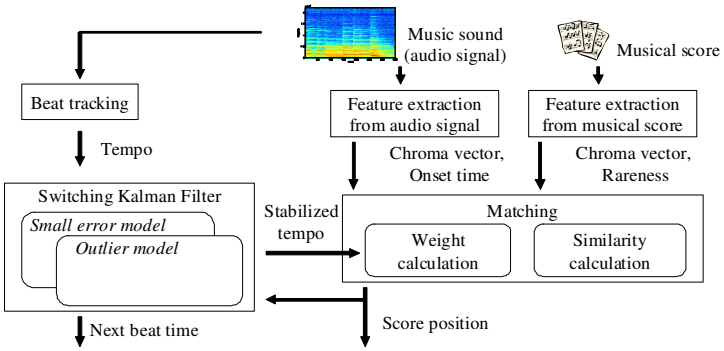

Fig. 7. Overview of our score alignment

component, which corresponds to one of 12 pitch names, is represented as Eq. (1) with band-pass filters for each pitch name.

$$c(i,t) = \sum_{h=\mathrm{Oct_L}}^{\mathrm{Oct_H}} \int_{-\infty}^{\infty} BPF_{i,h}(\omega) p(t,\omega) d\omega, \tag{1}$$

where $BPF_{i,h}$ is the band-pass filter for note name i at h-th octave. Oct_L and Oct_H are lower and higher bound octave to consider respectively. The peak of the band is the fundamental frequency of the note. The edges of the band are the frequencies of neighboring notes. For example, the BPF for note "A4" ("A" note at 4th octave) whose fundamental frequency is 440 [Hz] has the peak of its band at 440 [Hz]. The edges of its band are at 415 [Hz], "G♯4" and 466 [Hz], "A♯4." In this paper, we set $\mathrm{Oct_L} = 3$ and $\mathrm{Oct_H} = 7$. In other words, the lowest note was "C3", 131 [Hz], and the highest note was "B7", 3951 [Hz].

To emphasize the pitch name currently played, we apply the convolution in Eq. (2).

$$\begin{aligned} c'(i,t) \;=\; & -c(i+1,t-1) - 2c(i+1,t) - c(i+1,t+1) \\ & -c(i,t-1) \quad + 6c(i,t) \quad + 3c(i,t+1) \\ & -c(i-1,t-1) - 2c(i-1,t) - c(i-1,t+1) \end{aligned} \tag{2}$$

This convolution is processed cyclically for index i. For example, when $i = 1$ (pitch name is "*C*"), $c(i-1,t)$ is in fact substituted by $c(12,t)$ (pitch name is "*B*"). By subtracting the neighboring pitch name power, a component with more power than others can be emphasized, analogous to edge extraction in image processing. By subtracting the power of the previous time frame, the increase in power is stressed. Finally, we obtain the chroma vector for the audio signal, $c_{sig}(i,t)$ with Eq. (3).

$$c_{sig}(i,t) = \begin{cases} c'(i,t) & (c'(i,t) > 0), \\ 0 & otherwise. \end{cases} \tag{3}$$

2) Onset detection: We use the onset extraction method proposed by Rodet *et al.* [13]. This method exploits the increase in power at the onset time which lies particularly in the high frequency region. Sound onsets of pitched instruments have the centroid in the higher frequency region than those of percussive instruments such as drums. Thus, this method is particularly effective in detecting the onsets of pitched instruments. First, the power called high frequency content is obtained as:

$$h(t) = \sum_{\omega} \omega p(t,\omega). \tag{4}$$

High frequency content is a weighted power where the weight increases linearly with the frequency bin. Onset time t_n is determined by picking the peaks of $h(t)$ using a median filter.

TABLE I
DENOTATIONS OF INDICES

Symbols	Definitions
i	index for 12 pitch names (C, C♯, ..., B)
t	time frame of audio signal
n	index for onsets in audio signal
t_n	n-th onset time in audio signal
f	frame index of musical score
m	index for onsets in musical score
f_m	m-th onset frame in musical score

B. Feature Extraction from the Musical Score

1) Chroma vector generation: A musical score is divided into frames such that the length of one frame is equal to one-48th of a bar. This frame resolution can deal with sixteenth notes and triplets. The chroma vector for the musical score is defined as Eq. (5):

$$c_{sco}(i,m) = \begin{cases} 1 & \text{pitch name } i \text{ starts at frame } f_m, \\ 0 & otherwise, \end{cases} \tag{5}$$

where the index f_m means the m-th onset frame in the musical score.

2) The definition of rareness: The rareness $r(i,m)$ for each pitch name index i at frame f_m is defined as Eq. (7).

$$n(i,m) \;=\; \frac{\sum_{p \in M} c_{sco}(i,p)}{\sum_{i=1}^{12} \sum_{p \in M} c_{sco}(i,p)}, \tag{6}$$

$$r(i,m) \;=\; \begin{cases} -\log_2 n(i,m) & (n(i,m) > 0), \\ \max\limits_{i}(-\log_2 n(i,m)) & (n(i,m) = 0), \end{cases} \tag{7}$$

where M denotes a frame range whose length is two bars with its center at frame f_m. Therefore, $n(i,m)$ is the distribution of each pitch name around frame f_m.

C. Beat Tracking

We use the beat tracking method developed by Murata *et al.* [13]. First, a spectrogram $p(t,\omega)$ whose frequency bin is in linear scale is transformed into $p_{mel}(t,\varphi)$ whose frequency bin is in 64 dimensional Mel-scale. The onset vector $d(t,\varphi)$ is defined as in Eq. (8).

$$d(t,\varphi) = \begin{cases} p_{mel}^{sobel}(t,\varphi) & (p_{mel}^{sobel}(t,\varphi) > 0), \\ 0 & otherwise, \end{cases} \tag{8}$$

$$\begin{aligned} p_{mel}^{sobel}(t,\varphi) \;=\; & -p_{mel}(t-1,\varphi+1) + p_{mel}(t+1,\varphi+1) \\ & -2p_{mel}(t-1,\varphi) + 2p_{mel}(t+1,\varphi) \\ & -p_{mel}(t-1,\varphi-1) + p_{mel}(t+1,\varphi-1). \end{aligned} \tag{9}$$

Equation (9) means the onset emphasis with a Sobel filter.

Second, is beat interval (tempo) estimation. Beat interval reliability $R(t,k)$ is defined as Eq. (10) using normalized cross-correlation.

$$R(t,k) = \frac{\sum_j \sum_{l=0}^{P_w-1} d(t-l,j)d(t-k-l,j)}{\sqrt{\sum_j \sum_{k=l}^{P_w-1} d(t-l,j)^2 \sum_j \sum_{l=0}^{P_w-1} d(t-k-l,j)^2}}, \quad (10)$$

where P_w is the window length for reliability calculation and k is the time shift parameter. The beat interval $I(t)$ is determined based on the time shift value k where $R(t,k)$ takes the local peak.

D. Matching between Audio Signal And Musical Score

The matching process consists of two steps:

1) weighting subsequent score positions by using the tempo output by switching Kalman filters as described in section III-E,
2) calculation of similarities between audio signal chroma vectors at the onset time and the musical score chroma vector along with the *rareness*.

1) Weighting Score Positions Based on Tempo: Let (t_n, f_m) be the last matching pair, where t_n is time in the audio signal and f_m is the frame index of the musical score. Given the new onset in the audio signal detected at t_{n+1} and the tempo at that time, the number of frames, F, to go forward in the musical score is estimated as

$$F = A(t_{n+1} - t_n), \quad (11)$$

where factor A corresponds to the tempo. The faster the music is, the larger A becomes. The weight for score frame f_{m+k} is defined as,

$$W(k) = \exp\left(-\frac{(f_{m+k} - f_m - F)^2}{2\sigma^2}\right), \quad (12)$$

where k is the number of onsets in musical score to go forward, and σ is the variance for the weight. We set $\sigma = 24$ in our implementation, which corresponds to the length of half note. Note that k can be a negative number. Negative k allows us to consider the matching such as (t_{n+1}, f_{m-1}), where the matching moves backward in the musical score.

2) Similarities between Audio Signal and Musical Score: The similarity between the pair (t_n, f_m) is defined by the equation

$$S(n,m) = \sum_{i=1}^{12} \sum_{\tau=t_n}^{t_{n+1}} r(i,m) c_{sco}(i,m) c_{sig}(i,\tau), \quad (13)$$

where i is pitch name index, $r(i,m)$ is *rareness*, c_{sco} and c_{sig} are chroma vectors generated from musical score and audio signal respectively.

When the last matching is (t_n, f_m), the new matching will be (t_{n+1}, f_{m+k}), where

$$k = \operatorname*{argmax}_l W(l) S(n+1, m+l). \quad (14)$$

In our implementation, the search range of k for each matching step is limited within 2 bars to reduce computational cost.

E. Tempo Estimation using Switching Kalman Filters

We use switching Kalman filters (SKF) [14] to deal with two types of errors in the matching results and the tempo estimated by the beat tracking method. These are
(1) small errors caused by slight changes of the performance speed, and
(2) outliers in tempo estimation by beat tracking.

(1) Overview of SKF: SKF is an extension of Kalman filter (KF). KF is a linear prediction filter with state a transition model and an observation model. The KF estimates the state, which is unobservable, from observed values including errors in a discrete time series.

SKF has multiple state transition models and observation models. The model is automatically switched based on the likelihood of each model every time SKF obtains an observation value. In this paper, the SKF have two models: (1) small observation error model and (2) large observation error model for outliers. Other modeling elements such as state transitions are common to the two models.

(2) Modeling of SKF: We used the SKF model proposed by Cemgil *et al.* [15] for estimating the beat time and beat interval. Suppose the k-th beat time is b_k and the beat interval at that time is Δ_k and that the tempo is stable. The next beat time is represented as $b_{k+1} = b_k + \Delta_k$ and the next beat interval is $\Delta_{k+1} = \Delta_k$. Let state vector be $\mathbf{x}_k = [b_k \ \Delta_k]^{\mathrm{T}}$ and the state transition is represented as

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{v}_k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k, \quad (15)$$

where $\mathbf{F}_k$ is a state transition matrix and $\mathbf{v}_k$ is transition error vector derived from a normal distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{Q}$. Given the most recent state is $\mathbf{x}_k$, the next beat time b_{k+1} can be predicted as the first component of $\mathbf{x}_{k+1}$ shown as follows:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k \quad (16)$$

Let the observation vector be $\mathbf{z}_k = [b'_k \ \Delta'_k]^{\mathrm{T}}$, where b'_k is the beat time calculated from the matching result and Δ'_k is the beat interval reported by beat tracking. The observation is modeled as

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{w}_k, \quad (17)$$

where $\mathbf{H}_k$ is the observation matrix and $\mathbf{w}_k$ is the observation error vector derived from a normal distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{R}$.

Specifically, SKF switches observation error covariance matrices $\mathbf{R}^i$ $(i = 1,2)$, where i is model number. Through preliminary experiments, we set $\mathbf{R}^i$ as follows: $\mathbf{R}^1 = \mathrm{diag}(0.02, 0.005)$ for small error model and $\mathbf{R}^2 = \mathrm{diag}(1, 0.125)$ for outlier model, where $\mathrm{diag}(a_1, ..., a_n)$ denotes $n \times n$ diagonal matrix whose elements are $a_1, ..., a_n$ from top-left to bottom-right.

1) Observation of Beat Times: In our implementation, beats lie at every 12 frames in the score as the score is divided into frames whose length corresponds to 48th note. Observed beat time b'_k is interpolated by matching results when no note exists at the k-th beat frame.

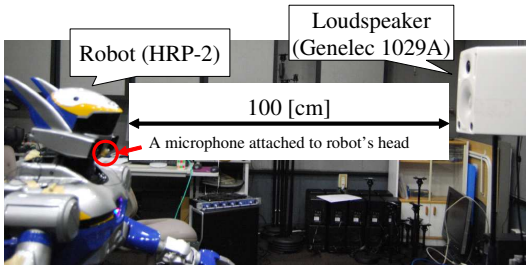

Fig. 8. Set up for impulse response measurement

IV. Experiments

This section reports on our experiments that evaluated the error in score alignment results of popular music pieces. We compare the error using four methods: (1) full of our method, (2) our method without SKF, (3) our method without rareness, and (4) the beat tracking only. Ground truth data for score alignment are generated from a MIDI file of each song. These MIDI files are strictly aligned with the actual performance. The error is defined as the absolute value of the difference between the beat time reported by each method and the ground truth data in seconds. The errors are averaged for each song.

A. Conditions

We use 100 pieces of popular music from the RWC music database (RWC-MDB-P-2001) developed by Goto *et al.* [16]. Needless to say, we use full length versions of these songs that include vocals and instruments for the experiments.

The specification of four methods are as follows:
(1) our method: the SKF and the rareness are in use,
(2) without SKF: no modification for tempo estimation,
(3) without rareness: all notes have equal rareness,
(4) beat tracking: this method determines the score position by counting the beats from the beginning of the music.

We experiment with using two kinds of music signals to evaluate what effect the reverberation in the room environment would have.

1) Clean music signal: music signal without reverberation,
2) Reverberated music signal: music signal with reverberation. The reverberation is simulated by impulse response convolution. Figure 8 shows the set up for impulse response measurement. This impulse response is measured in an experimental room. The reverberation time (RT_{20}) in the room is 156 [msec]. An auditorium or a music hall would have a longer reverberation time.

B. Results and Discussion

Table II shows the results with two types of music signals and four methods. The values are the means and standard deviations of 100 songs. The error in our method (1) is less than beat tracking method (4) with both clean and reverberated signals. Our method improves the error by 29% with clean signals and by 14% with reverberated signals. The SKF decreases the errors because method (1) has less errors than (2). Similarly, the rareness decreases the errors comparing the results of method (1) and (3). The results also indicate that the SKF is more effective than the rareness since method (2) has less errors than (3). This is because the rareness sometimes induces high similarity between a certain frame in the musical score and the wrong onset such as drum sounds. If the drum sound happens to have large power in a component of chroma vector with high rareness, this results in the wrong matching. To avoid this problem, we can consider the rareness for the combination of pitch names, not for a single pitch name.

Figure 9 and 10 show the number of music tunes distributed by the average error value for each method. The larger number of tunes with less average error indicates the better performance. With clean signals, Our method has 31 tunes with less than 2-second error while the beat tracking has 9 tunes. With reverberated signals, similarly, our method has 36 tunes with less than 2-second error while the beat tracking has 12 tunes. Thus, our method is superior to the beat tracking method in terms of estimating the score position with less errors. This is essential for generating a natural singing voice along with the music sound. The distribution of our method makes little difference between clean signals and reverberated signals, although our method has more error

TABLE II
Results: Average and Standard Deviation of Cumulative Absolute Errors

	Clean signal		Reverberated signal	
	Ave.	Std. dev.	Ave.	Std. dev.
(1) Our method	8.9	11.0	9.9	12.6
(2) w/o SKF	11.6	12.8	10.5	11.2
(3) w/o rareness	9.7	13.5	10.3	13.3
(4) Beat tracking	12.5	9.7	11.5	9.1

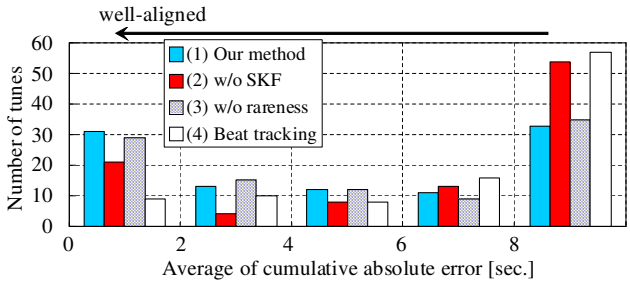

Fig. 9. The number of tunes by average error with clean signal

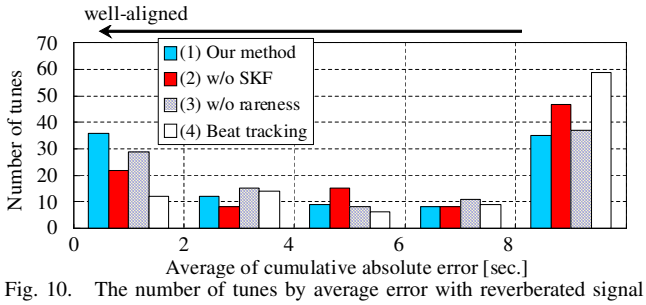

Fig. 10. The number of tunes by average error with reverberated signal

with reverberated signals shown as Table II. Therefore, the reverberation in our experimental room mainly affects the songs with much error. The reverberation has little effect on the songs with little error. A longer reverberation such as in a music hall can deteriorate the score alignment accuracy.

We confirm that the accuracy of our method depends on whether drums are played in a song by comparing the errors of musical tunes with drum sounds and of those without drum sounds. The number of musical tunes with and without drum sounds is 89 and 11, respectively. The average cumulative error of songs with drums is 7.3 [sec.] with the standard deviation 9.4 [sec.]; whereas that of songs without drums is 22.1 [sec.] with the standard deviation 14.5 [sec.]. The tempo estimation by beat bracking is apt to have huge fluctuations when drum sounds are absent. This causes incorrect matchings, leading to a high cumulative error.

There are other two reasons for errors in our method:

1) Matching error caused by false onset detections: False onset detections due to percussion sounds or signal power changes by vibrato or other shaky musical expressions cause matching at a previous score position. This mismatching leads to tempo acceleration and increased matching error.
2) Sound effects: Some songs contain sound effects that are not written in the musical score as musical note information. Sound effects can be interpreted as noise for our matching method based on chroma vector similarity.

V. Conclusion

Our goal is to develop an interactive singer robot. We first discussed three capabilities desired for the robot: (1) listening, (2) capability to understand score position and (3) singing. In this paper, we described a way to realize the second function, namely, a method to align a musical audio signal with the score. To create a score alignment method robust to the diversity in timbre of musical instruments, we designed the features based on chroma vector and onset of the sound. We also defined rareness for each pitch name to alleviate the difference between the audio signal and the musical score. For robust tempo estimation, we used a switching Kalman filter with small error model and outlier model. The experimental results indicated that our method had less error than the existing beat tracking method without score alignment by 29% with respect to cumulative absolute errors. However, false onset detections deteriorated the performance of our method. To improve this, we may predict the next pitched sound by reading ahead in the score in advance.

In our work in the near future, we plan to develop a singing system as shown in Figure 1 for a humanoid robot called HRP-2. In further future work, we intend to expand the singing expression such as active body motion associated with the quality of vocal sound. The motion data can be acquired, for example, using motion capture data of a human singer.

References

[1] K. Murata, K. Nakadai, K. Yoshii, R. Takeda, T. Torii, H. G. Okuno, Y. Hasegawa, and H. Tsujino. A robot uses its own microphone to synchronize its steps to musical beats while scatting and singing. In *IROS*, pages 2459–2464, 2008.

[2] T. Mizumoto, R. Takeda, K. Yoshii, K. Komatani, T. Ogata, and H. G. Okuno. A robot listens to music and counts its beats aloud by separating music from counting voice. In *IROS*, pages 1538–1543, 2008.

[3] R. Takeda, K. Nakadai, K. Komatani, T. Ogata, and H. G. Okuno. Barge-in-able robot audition based on ica and missing feature theory under semi-blind situation. In *IROS*, pages 1718–1723, 2008.

[4] H. Kenmochi and H. Ohshita. Vocaloid – commercial singing synthesizer based on sample concatenation. In *INTERSPEECH*, pages 4010–4011, 2007.

[5] A. Nakazawa, S. Nakaoka, and K. Ikeuchi. Imitating human dance motions through motion structure analysis. In *IROS*, pages 2539–2544, 2002.

[6] M. P. Michalowski, H. Kozima, and S. Sabanovic. A dancing robot for rhythmic social interaction. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction*, pages 89–96, 2007.

[7] G. Weinberg and S. Driscoll. Toward robotic musicianship. *Computer Music Journal*, 30(4):28–45, 2006.

[8] G. Weinberg and S. Driscoll. The design of a perceptual and improvisational robotic marimba player. In *Proc. of IEEE International Workshop on Robot and Human Interactive Communication*, pages 132–137, 2007.

[9] K. Yoshii, K. Nakadai, T. Torii, Y. Hasegawa, H. Tsujino, K. Komatani, T. Ogata, and H. G. Okuno. A biped robot that keeps steps in time with musical beats while listening to music with its own ears. In *IROS*, pages 1743–1750, 2007.

[10] Dannenberg, Roger B., and Ning Hu. Polyphonic audio matching for score following and intelligent audio editors. In *Proc. of the International Computer Music Conference*, pages 27–33, 2003.

[11] N. Orio, S. Lemouton, and D. Schwartz. Score following: state of the art and new developments. In *Proc. of the conference on New interfaces for musical expression*, pages 36–41, 2003.

[12] A. Cont. Realtime audio to score alignment for polyphonic music instruments and using sparse non-negative constraints and hierarchical hmms. In *IEEE Int'l Conference in Acoustics and Speech Signal Processing*, volume V, pages 245–248, 2006.

[13] X. Rodet and F. Jaillet. Detection and modeling of fast attack transients. In *International Computer Music Conference*, pages 30–33, 2001.

[14] K. P. Murphy. Switching kalman filters. Technical report, 1998.

[15] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram representation and kalman filtering. *Journal of New Music Research*, 28:4:259–273, 2001.

[16] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Popular music database and royalty-free music database. volume 2001, pages 35–42.