A ROBOT SINGER WITH MUSIC RECOGNITION BASED ON REAL-TIME BEAT TRACKING

Kazumasa Murata†, Kazuhiro Nakadai‡,†, Kazuyoshi Yoshii*, Ryu Takeda*, Toyotaka Torii‡, Hiroshi G. Okuno*, Yuji Hasegawa‡ and Hiroshi Tsujino‡

† Graduate School of Information Science and Engineering, Tokyo Institute of Technology
‡ Honda Research Institute Japan Co., Ltd., ∗ Graduate School of Informatics, Kyoto University
murata@cyb.mei.titech.ac.jp, {nakadai, tory, yuji.hasegawa, tsujino}@jp.honda-ri.com, {yoshii,rtakeda,okuno}@kuis.kyoto-u.ac.jp

ABSTRACT

A robot that can provide an active and enjoyable user interface is one of the most challenging applications for music information processing, because the robot should cope with high-power noises including self voices and motor noises. This paper proposes noise-robust musical beat tracking by using a robot-embedded microphone, and describes its application to a robot singer with music recognition. The proposed beat tracking introduces two key techniques, that is, spectro-temporal pattern matching and echo cancellation. The former realizes robust tempo estimation with a shorter window length, thus, it can quickly adapt to tempo changes. The latter is effective to cancel self periodic noises such as stepping, scatting, and singing. We constructed a robot singer based on the proposed beat tracking for Honda ASIMO. The robot detects a musical beat with its own microphone in a noisy environment. It tries to recognize music based on the detected musical beat. When it successfully recognizes music, it sings while stepping according to the beat. Otherwise, it performs scatting instead of singing because the lyrics are unavailable. Experimental results showed fast adaptation to tempo changes and high robustness in beat tracking even when stepping, scatting and singing.

1 INTRODUCTION

Music information processing draws attention of researchers and industrial people for recent years. Many techniques in music information processing such as music information retrieval are mainly applied to music user interfaces for cellular phones, PDAs and PCs, and various commercial services have been launched[12]. On the other hand, robots like humanoid robots are recently getting popular. They are expected to help us in a daily environment as intelligent physical agents in the future. This means that the robot should not only perform tasks but also make us more enjoyable than PDA or PC based interface. Thus, music is important media for such rich human-robot interaction because music is one of the popular hobbies for humans. This will contribute to MIR society in a sense that robot provides real-world MIR applications. Therefore, we started to apply music information processing to robots. As a first step, we focused on musical beat tracking because it is a basic function to recognize music. However, to be applied to a robot , three issues should be considered for beat tracking as follows:

1. real-time processing by using a robot-embedded microphone,
2. quick adaptation to tempo changes, and
3. high noise-robustness for environmental noises, a robot's own voices and motor noises.

The first issue is crucial to realize natural user interface. A lot of beat-tracking methods have been studied in the field of music information processing [6]. They focus on extraction of complicated beat structures with off-line processing, although there are some exceptions like [5, 8]. Nakadai *et al*. reported the importance of auditory processing by using robots' own ears. They proposed "robot audition" as a new research area[14]. Some robot audition systems which achieved highly noise-robust speech recognition have been reported [7, 18]. However, beat tracking for noisy signals such as robot-noise-contaminated music signals has not been studied so far. The second issue is essential for real-world applications like a robot. For example, in [19], Goto's algorithm was used. It was able to cope with real recording data such as CD music and to apply it to software robot dancer called *Cindy*[3], because it integrates 12 different agents to track musical beats. However, this approach to improve robustness results in insensitivity of tempo changes. This is because a self-correlation-based method requires a longer window to improve noise-robustness, while a short window is necessary to adapt to drastic tempo changes quickly. Thus, they reported that it took around ten seconds to adapt a stepping cycle to tempo changes. Indeed, some probabilistic methods were proposed to cope with tempo changes [10, 2], but these methods tend to require high computational costs and the large amount of memory. Thus, they have difficulty in embedded applications. The last issue is similar to the first one in terms of a noise problem. However, when we consider singing, scatting and stepping functions synchronizing to musical beats, a new problem arises. The noises caused by such functions are periodic because they are generated according to "periodic" beat signals. If the noises and the beats are synchronized, there will be no problem. How-

ever, because scatting/singing is based on estimated beats, entrainment can occur between real and estimated beats in tempo and phase. Thus, it takes a while for them to attain fully synchronization, that is, there is no error between these two beats. This means that the noises affect the performance of beat tracking badly. Scatting and singing cause a much bigger problem than stepping, because the loudspeaker embedded in a robot is usually closer to a robot-embedded microphone than motors and fans. These noises should be suppressed.

In this paper, we proposed a new real-time beat-tracking algorithm by using two techniques to solve the above three issues. One is spectro-temporal pattern matching to realize faster adaptation to tempo changes. The other is noise cancellation based on semi-blind Independent Component Analysis (semi-blind ICA)[16]. We then developed a robot singer with a music recognition function based on proposed real-time beat-tracking for Honda ASIMO. When music is played, the developed robot first detects its beat, secondly recognizes the music based on musical beat information to retrieve the lyrics information from a lyrics database, and finally sings with stepping synchronizing to its musical beat. We evaluated the performance of the proposed beat tracking method in terms of adaptation speed, and noise-robustness through the developed robot system.

2 RELATED WORK IN ROBOTICS

In robotics, music is a hot research topic[1]. Sony exhibited a singing and dancing robot called QRIO. Kosuge *et al*. showed that a robot dancer, MS DanceR, performed social dances with a human partner [17]. Nakazawa *et al*. reported that HRP-2 imitated the spatial trajectories of complex motions of a Japanese traditional folk dance by using a motion capture system [15]. Although these robots performed dances and/or singing, they were programmed in advance without any listening function. Some robots have music listening functions. Kotosaka and Schaal [11] developed a robot that plays drum sessions with a human drummer. Michalowski *et al*. developed a small robot called Keepon which can move its body quickly according to musical beats [13]. Yoshii *et al*. developed a beat tracking robot using Honda ASIMO [19]. This robot was able to detect musical beats by using a real-time beat tracking algorithm [3], and the robot that times its steps to the detected musical beats was demonstrated. These robots worked well only when a music signal is given. However, it is difficult for them to cope with noises such as environmental noises, self voices, and so on. Thus, they have difficulties in singing and scatting that make high power noises.

3 REAL-TIME BEAT TRACKING ALGORITHM

Figure 1 shows an overview of our newly-developed real-time beat tracking algorithm. This algorithm has two input

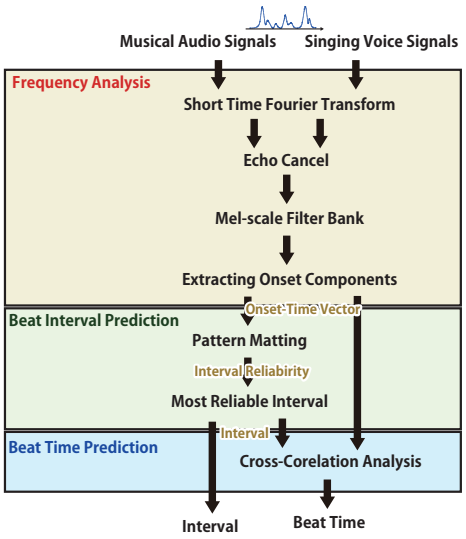

Figure 1. Overview of our real-time beat-tracking

signals. One is a music signal which is usually contaminated by noise sources such as self-noises. The other is a self-noise signal such as a scatting or a singing voice. Because the self-noise is known in advance for the system, pure self-noise can be directly obtained from line-in without using a microphone. The outputs are predicted beat time, and tempo value. It consists of three stages – frequency analysis, beat interval prediction and beat time prediction.

3.1 Frequency Analysis

Spectra are consecutively obtained by applying the short time Fourier transform (STFT) to two input signals sampled at 44.1 kHz. The Hanning window of 4,096 points is used as a window function, and its shift length is 512 points. Echo canceling is, then, applied. It is essential to eliminate self-noises such as singing and scatting voices to improve beat tracking. We introduced semi-blind ICA for echo cancellation[16] which was proposed by our group for self-voice cancellation. We also extended this method to support multi-channel input signals. We used a two-channel version of semi-blind ICA. One channel takes the spectra contaminated by self-noises as an input, and the other channel takes a pure self-noise as an input. The noise-suppressed spectra are sent to Mel-scale Filter Bank. It reduces the number of frequency bins from 2,049 linear frequency bins to 64 mel-scale frequency bins to reduce computational costs in later processes. A frequency bin where a spectral power rapidly increases is detected as an onset candidate at the mel-scale frequency domain. We used the Sobel filter, which is used for visual edge detection, to detect frequency bins only with rapid power increase. Let $d_s(t,f)$ be the spectral power at the t-th time frame and the f-th mel-filter bank bin after the Sobel filtering. An onset belief $d(t,f)$ is estimated by

$$d(t,f) = \begin{cases} d_s(t,f) & \text{if } \mathrm{d_s(t,f)} > 0, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

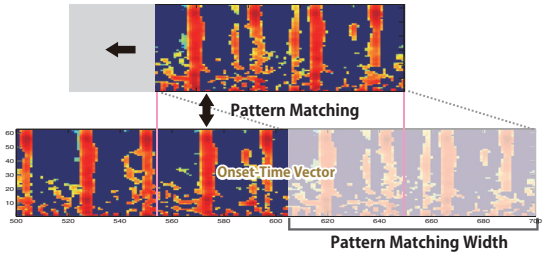

Figure 2. Spectro-Temporal Pattern Matching

where $f = 1, 2, ..., 62$. Thus, a 62-dimensional onset time vector is extracted for each time frame.

3.2 Beat Interval Prediction

To estimate a beat interval defined as the temporal difference between two neighboring beats, spectro-temporal pattern matching is performed by using the onset time vector. As a pattern matching function, we used Normalized Cross-Correlation (NCC) defined by

$$R(t,i) = \frac{\sum_{j=1}^{62}\sum_{k=0}^{P_{width}-1} d(t-k,j)d(t-i-k,j)}{\sqrt{\sum_{j=1}^{62}\sum_{k=0}^{P_{width}-1} d(t-k,j)^2 \cdot \sum_{j=1}^{62}\sum_{k=0}^{P_{width}-1} d(t-i-k,j)^2}} \quad (2)$$

where P_{width} is window length for pattern matching, and i is the shift parameter (Fig. 2).

Frequency-line-based self-correlation is often used for interval estimation. It requires a longer window length for the self-correlation function to improve robustness. This leads to insensitivity to tempo changes. The proposed method uses NCC defined in Eq.(), which corresponds to a kind of *whitening* in signal processing. This improves noise-robustness, even when a window length is as short as 1 sec[1] . Therefore, faster adaptation to tempo changes is achieved. A set of local peaks is, then, extracted by

$$R_p(t,i) = \begin{cases} R(t,i) & \text{if R(t,i}-1) < \text{R(t,i)} < \text{R(t,i}+1), \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

When two peaks have comparable reliabilities, mis-detection of beat interval occurs. To avoid this mis-detection, beat interval is limited from 61 to 120 M.M.[2] When beat intervals for the first and the second biggest local peaks in $R_p(t,i)$ are I_1 and I_2, beat interval at time t is estimated by

$$I(t) = \begin{cases} 2|I_1 - I_2| & (|I_{n2} - I_1| < \delta \text{ or } |\text{I}_{n2} - \text{I}_2| < \delta) \\ 3|I_1 - I_2| & (|I_{n3} - I_1| < \delta \text{ or } |\text{I}_{n3} - \text{I}_2| < \delta) \\ I_1 & otherwise, \end{cases} \quad (4)$$

$$I_{n2} = 2|I_1 - I_2|, \quad I_{n3} = 3|I_1 - I_2|,$$

[1] This is minimum window length because the lower tempo limit is 60BPM due to a hardware specification of our robot.

[2] Mälzel's Metronome: the number of quarter notes per minute. For example, if the tempo is 60 M.M., the quarter-note length is 1,000 [ms].

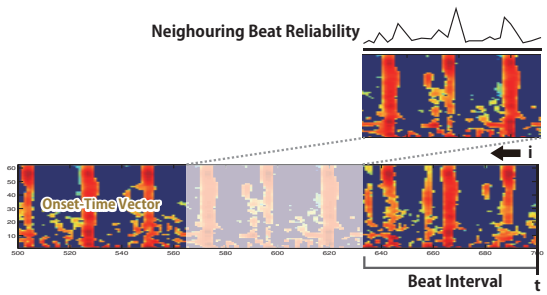

Figure 3. Neighboring Beat Reliability.

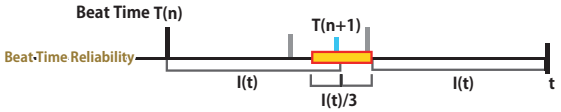

Figure 4. Beat Time Detection

where δ means an error margin parameter. This formulation are defined empirically to avoid mis-estimation such as double and triple tempos.

3.3 Beat Time Prediction

Beat reliability is estimated from two types of reliabilities – neighboring beat reliability and continuous beat reliability. Beat time is predicted according to beat reliability.

Neighboring beat reliability is a reliability on beat existence, and is calculated at the current time and at the previous beat time by using the beat interval shown in Fig. 3. A neighboring beat reliability $S_c(t,i)$ for time $t-i$ at time t is denoted by

$$S_c(t,i) = \begin{cases} \sum_{f=1}^{62} d(t-i,f) + \sum_{f=1}^{62} d(t-i-I(t),f) & (i \le I(t)) \\ 0 & (i > I(t)). \end{cases} \quad (5)$$

Continuous beat reliability is a reliability of a temporal beat sequence. It is calculated from neighboring beat reliabilities.

$$S_r(t,i) = \sum_{m}^{N_{S_r}} S_c(T_p(t,m),i) \quad (6)$$

$$T_p(t,m) = \begin{cases} t - I(t) & (m=0) \\ T_p(t,m-1) - I(T_p(t,m)) & (m \ge 1) \end{cases}$$

where $S_r(t,i)$ denotes continuous beat reliability for time $t-i$ at time t. $T_p(t,m)$ means the m-th previous beat time for time t, and N_{S_r} is the number of beats to calculate continuous beat reliability. This reliability is effective to decide the best beat sequence such as strong beats when multiple beat sequences are detected.

The neighboring beat reliability and the continuous beat reliability are integrated into a beat reliability defined by

$$S(t) = \sum_i (S_c(t-i,i)S_r(t-i,i)). \quad (7)$$

Beat time is then detected. Let the n-th beat time be $T(n)$. When $T(n) \geq t - \frac{3}{4}I(t)$, three-best peaks in $S(t)$ are extracted from $T(n)+\frac{1}{2}I(t)$ to $T(n)+\frac{3}{2}I(t)$. The peak which is closest to $T(n)+I(t)$ is estimated as the next beat time $T(n+1)$ shown in Fig. 4. In case that no peak is found from $T(n)+\frac{2}{3}I(t)$ to $T(n)+\frac{4}{3}I(t)$, $T(n)+I(t)$ is regarded as $T(n+1)$. This beat time detection process was defined empirically. The detected beat time $T(n+1)$ is a past beat, that is, $t > T(n+1)$. To apply beat tracking to scatting or singing, a future beat time T' should be predicted. By using the following extrapolation, a future beat time is predicted.

$$T' = \begin{cases} T_{\text{tmp}} & \text{if } \mathrm{T_{tmp}} \geq \frac{3}{2}\mathrm{I_m(t)} + \mathrm{t} \\ T_{\text{tmp}} + I_m(t) & \text{otherwise.} \end{cases} \quad (8)$$

$$T_{\text{tmp}} = T(m) + I_m(t) + (t - T(m)) - \{(t - T(m)) \mod I_m(t)\}$$

where $I_m(t)$ is a median value of a set of $I(t)$, and $T(m)$ is the latest beat time detected in beat time detection.

4 IMPLEMENTATION OF ROBOT SINGER

Fig. 5 shows the architecture of our robot singer based on the proposed beat-tracking. The system mainly consists of four components – Real-time Beat Tracker, Music Recognizer, Robot Controller, and Humanoid Robot. The Real-time Beat Tracker estimates predicted beat time and a beat interval from a noise-contaminated music signal captured by a robot's microphone as described in Sec. 3. The other three components are described in the following sections. In terms of implementation, Real-time Beat Tracker and Music Recognizer were implemented by C++ on Linux. These components work in real time on a remote PC with Pentium 4. In Robot Controller, scatting and stepping are running the same PC as the above two components, which only singing function is running on Windows PC.

4.1 Specifications of Humanoid Robot

We used Honda ASIMO with a microphone embedded in the head for a singer robot. It has two legs like humans and can stamp its feet on the floor, i.e., perform steps in a stationary location. The step interval is limited to between 1,000 and 2,000 [ms]. If the tempos of musical pieces are between 61 and 120 M.M., The robot records these signals with its own single microphone embedded in the front of the head. It has a loudspeaker for singing at the position of its chest.

4.2 Music Recognizer

Music recognizer consists of two parts – music activity detection and music retrieval. In music activity detection, beat stability is estimated as a ratio of a stable beat period in 3 seconds. When the time difference between the current tempo and the estimated beat interval is within 55 ms, the beat is estimated as stable. When the ratio is higher than 0.8, such a 3-second period is regarded as music. These

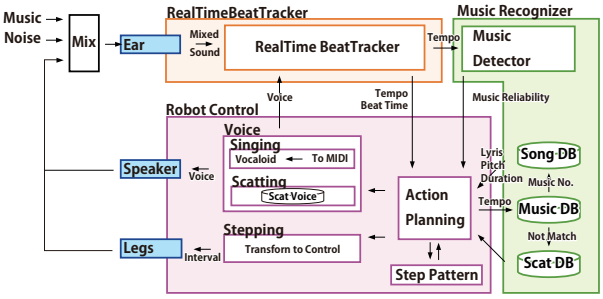

Figure 5. Architecture of a robot singer

thresholds were empirically obtained. Music retrieval returns music ID in the music database by retrieving music which has the closest beat to the estimated one. We simply used tempo information in this retrieval. Practically, when the tempo difference between music and the estimated tempo was within 11 ms, such music was selected. When such music was not found, "unknown music" was returned as a music ID. Music retrieval then obtained the lyrics and notes for the music ID from a song database. In case of unknown music, scatting sounds such as "Zun" and "Cha" were obtained from a scat database in order to utter them instead of singing. Finally, this information was sent to a Robot Control.

4.3 Robot Controller

Robot Controller controls ASIMO to time its steps to musical beats, and to synchronize singing or scatting with the beat. The voices are outputted from a loudspeaker inside ASIMO. The control of stepping is done by using a command via a TCP/IP network.

The stepping function is used to adjust step timings to musical beats only by using a command of specifying a step interval. Because an accurate target value is unavailable, it is theoretically difficult to control a robot even when sophisticated feedback control is used in this case. Thus, we used a simple feedback control to reduce the errors of step timing and interval.

Singing means that a robot sings according to musical beats. Thus, when a music tempo decreases, the robot can sing slowly. As prior information, the melody and lyrics of the music are given to the system as MIDI data. VOCALOID developed by YAMAHA is used as a singing engine. It achieves a singing function with around 200 ms latency. The robot outputs singing voices synchronizing to musical beats by taking such latency into account.

Scatting is used when any appropriate music is not found. Scatting means, here, that a robot outputs sounds according to a beat pattern. In this paper, "zun" was outputted for a strong beat, and "cha" for a weak beat. Since these words have some durations, synchronization between these words and beat time includes some ambiguities. When their correspondence is slightly changed, people easily feel that it is unnatural or the robot is tone deaf. We empirically decided

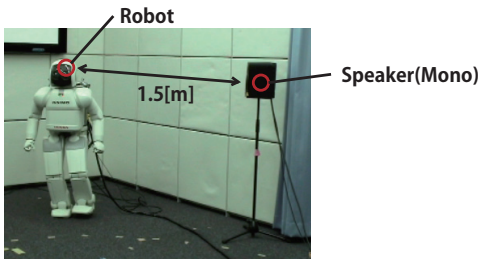

Figure 6. Overview of experimental condition: The system concerning to the robot is completely separated form that concerning to the music playback.

to use onset time of these words, which are detected by onset detection to synchronize with musical beats.

5 EVALUATION

We evaluated our beat tracking using our singer robot in the following three points: 1) adaptation speed to tempo changes, 2) noise-robustness using a beat prediction success rate, 3) music recognition in noisy environments. Three kinds of musical signals were used for these experiments.

T1 *musical signal including tempo changes*
T2 *musical signal with fixed tempo*
T3 *noisy music signals*

For **T1**, we prepared a 4-minute musical signal by selecting three songs (#11, #18, and #62) from the RWC music database (RWC-MDB-P-2001) developed by Goto *et al*. [4]. They include vocals and various instruments as commercial CDs do. Their tempos were 90, 112, and 81 M.M., respectively. We concatenated four 60-s segments that were extracted from the four pieces. For **T2**, we synthesized a musical signal of #62 by using MIDI data. MIDI data provides reference data of beat times. MIDI data is not used as a prior information for tempo and beat time detection. For **T3**, we prepared 10 minute data. The data includes five music signals, i.e., #4, #11, #17, #18 and #29. Each music appears with noises for 20 seconds, and only noise signals are included for the next 20 seconds. For noise data, we used exhibition noise in a booth included in JEIDA-NOISE database. A SNR in T3 was about -4 dB on average.

In every experiment, a loudspeaker was set in a 4 m×7 m room with 0.2 seconds of reverberation time (RT_{20}). The distance between the robot and the speaker was 1.5 m. The musical signals were played from the loudspeaker. This situation is outlined in Fig. 6.

For the first experiment, we used **T1**. The beat tracking delay was measured in five conditions, and was compared with a conventional self correlation based method in [3]. The beat tracking delay was defined as the time difference between when an actual tempo was changed and when the system adapted to the tempo change. Two conditions of the five were the ones with and without scatting when ASIMO was turned off. The other three conditions were the ones without scatting, with scatting and with singing when ASIMO was turned on and performed stepping.

For the second experiment, we used **T2**, and the beat prediction success rate was measured in five conditions. The beat prediction success rate r is defined by

$$r = \frac{100 \cdot N_{\text{success}}}{N_{\text{total}}}. \tag{9}$$

where N_{success} is the number of successfully predicted beats, and N_{total} is the number of total beats. When the error of a predicted beat time is within $\pm 0.35 I(t)$ as defined in [3], it is regarded as successfully predicted. Three conditions of the five are the ones when ASIMO was turned off. One was the condition without scatting and with echo canceling. Another two were the ones with and without canceling while scatting. The other two conditions of the five are the ones with and without echo canceling when ASIMO was turned on with stepping while scatting. For the last experiment, we used **T3**. As metrics for music activity detection, we used precision(P), recall(R), and F-measure(F) defined by

$$P = \frac{C}{N}, \quad R = \frac{C}{A}, \quad F = \frac{2 \cdot P \cdot R}{P + R} \tag{10}$$

where C is a period when music is successfully detected, N is the total period estimated as music, and A is the total music length. As a metric for music retrieval, we used music recognition rate (M) defined by

$$M = \frac{C'}{N} \tag{11}$$

where C' is a period when music was retrieved correctly.

5.1 Results

Table 1 shows the results for the first experiment. This shows that our proposed method adapted to the tempo changes 20 times faster than the conventional one when no voice exists, and it is still 10 times faster than when scatting voices exist. The self-correlation based system failed in beat tracking when singing voices existed, while the proposed was still robust. Table 2 shows the results of the second experiment. "Correct" means the beat tracking system correctly predicted beats, that is, strong beats. "Half-shifted" means that it predicted beats, but weak beats were predicted. This shows self-noises affected beat tracking due to its periodicity, and echo cancel drastically reduced the effect of such self-noises. However, other noises generated by robot's motors and fans were not suppressed explicitly in this paper. Such noise suppression will be attained by using microphone array techniques [18]. Table 3 shows the results of the last experiment. The average precision was around 10 points higher than the average recall. This is caused by the fact that music activity detection is unstable for 2.4 seconds (3 × 0.8) from the beginning of the music due to the

Table 1. Tracking Delay for Tempo Changes (in second)

	ASIMO power off		ASIMO with step		
scatting	off	on	off	on	off
singing	off	off	off	off	on
self-correlation	11.24	29.91	14.66	20.43	N/A
proposed	1.31	1.31	1.29	1.29	1.29

Table 2. Beat Prediction Success Rate

	ASIMO power off			ASIMO power on (with step)	
scatting	off	on		on	
echo cancel	off	on	off	on	off
Correct	95%	97%	68%	95%	64%
Half shifted	5%	1%	40%	4%	40%

Table 3. Music Recognition Result (P: precision, R: recall rate, F: f-measure)

		with noise			clean		
ID	bpm	P (%)	R (%)	F	P (%)	R (%)	F
#4	86	94.7	84.9	0.90	94.8	81.2	0.87
#11	90	74.3	67.3	0.71	96.1	72.1	0.82
#17	97	88.0	83.1	0.85	95.3	81.6	0.88
#29	103	93.4	81.5	0.87	95.9	82.2	0.88
#18	112	89.6	82.8	0.86	95.9	83.2	0.89

3-second window. In #11 and #17, precision was affected by noises. This is because the noise includes a periodic signal between 90 and 97 bpms. *M* was 95.8% for clean data, and 88.5% for noisy data. We can say that music recognition worked well for a small number of songs although using only tempo information. To improve the scalability of music recognition, we will use higher information such as rhythmic features such as [9].

6 CONCLUSIONS

We presented a real-time beat-tracking method for robots which is noise-robust and quickly-adaptable to musical beat changes. The method uses spectro-temporal pattern matching to improve the adaptation speed against tempo changes, and echo canceling based on semi-blind independent component analysis to suppress self periodic noises such as scatting and singing. We showed a singer robot using Honda ASIMO as an application of the proposed beat-tracking. It sings or scats while stepping synchronized to musical beats detected by using robot-embedded microphones, and also it has a simple function to recognize music based on musical beat information. Performance evaluation of the proposed beat tracking method showed high noise-robustness, quick adaptation to tempo changes, high music recognition performance. We believe that the proposed method and its extension will help to realize more active and enjoyable user interface through music, although further evaluation with benchmark datasets is necessary to know its performance precisely. More sophisticated robot motions such as dancing, improvements of robustness of beat tracking, introduction of other music information processing are remaining future work.

7 REFERENCES

[1] J. J. Aucouturier *et al*. Cheek to Chip: Dancing Robots and AI's Future. *Intelligent Systems, IEEE*, 23(2):74–84, 2008.

[2] A. Cemgil and B. Kappen. Monte carlo methods for tempo tracking and rhythm quantization. *J. of Artificial Intelligence Research*, 18:45–81, 2003.

[3] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *J. of New Music Research*, 30(2):159–171, 2001.

[4] M. Goto *et al*. RWC music database: Popular, classical, and jazz music databases. In *Int. Conf. Music Info. Retrieval*, pages 287–288, 2002.

[5] M. Goto and Y. Muraoka. A real-time beat tracking system for audio signals. In *Proc. of the Int'l Computer Music Conf.*, pages 171–174, San Francisco CA, 1995. International Computer Music Association.

[6] F. Gouyon *et al*. An experimental comparison of audio tempo induction algorithms. *IEEE Trans. Audio, Speech and Language Processing*, 14(5):1832–1844, 2006.

[7] I. Hara *et al*. Robust speech interface based on audio and video information fusion for humanoid HRP-2. In *Proc. of IEEE/RSJ Int'l Conf. on Intel. Robots and Systems (IROS 2004)*, pages 2404–2410. IEEE, 2004.

[8] K. Jensen and T.H. Andersen. Real-time beat estimation using feature extraction. *Proc. Computer Music Modeling and Retrieval Symposium, Lecture Notes in Computer Science. Springer Verlag*, 2003.

[9] D. Kirovski and H. Attias. Beat-ID: Identifying music via beat analysis.

[10] A. P. Klapuri *et al*. Analysis of the meter of acoustic musical signals. *IEEE Trans. Audio, Speech, and Language Processing*, 14(1), 2006.

[11] S. Kotosaka and S. Schaal. Synchronized robot drumming by neural oscillators. In *Proc. of Int'l Sympo. Adaptive Motion of Animals and Machines*, 2000.

[12] T. Kurozumi *et al*. A robust audio searching method for cellular-phone-based music information retrieval. In *Proc. of Int'l Conf. on Pattern Recognition (ICPR'02)*, volume 3, page 30991, 2002.

[13] M. P. Michalowski *et al*. A dancing robot for rhythmic social interaction. In *Proc. of ACM/IEEE Int'l Conf. on Human-Robot Interaction (HRI 2007)*, pages 89–96. IEEE, 2007.

[14] K. Nakadai *et al*. Active audition for humanoid. In *Proc. of National Conf. on Artificial Intelligence (AAAI-2000)*, pages 832–839. AAAI, 2000.

[15] A. Nakazawa *et al*. Imitating human dance motions through motion structure analysis. In *Proc. of IEEE/RSJ Int'l Conf. on Intel. Robot s and Systems (IROS-2002)*, pages 2539–2544, 2002.

[16] R. Takeda *et al*. Exploiting known sound sources to improve ica-based robot audition in speech separation and recognition. In *Proc. of IEEE/RSJ Int'l Conf. on Intel. Robots and Systems (IROS-2007)*, pages 1757–1762, 2007.

[17] T. Takeda *et al*. Hmm-based error detection of dance step selection for dance partner robot –MS DanceR–. In *Proc. of IEEE/RSJ Int'l Conf. on Intel. Robots and Systems (IROS-2006)*, pages 5631–5636, 2006.

[18] S. Yamamoto *et al*., T. Ogata, and H. G. Okuno. Real-time robot audition system that recognizes simultaneous speech in the real world. In *Proc. of IEEE/RSJ Int'l Conf. on Intel. Robots and Systems (IROS 2006)*, pages 5333–5338. IEEE, 2006.

[19] K. Yoshii *et al*. A biped robot that keeps steps in time with musical beats while listening to music with its own ears. In *Proc. of IEEE/RSJ Int'l Conf. on Intel. Robots and Systems (IROS-2007)*, pages 1743–1750, 2007.