The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Robot Musical Accompaniment: Integrating Audio and Visual Cues for Real-time Synchronization with a Human Flutist

Angelica Lim, Takeshi Mizumoto, Louis-Kenzo Cahier, Takuma Otsuka,
Toru Takahashi, Kazunori Komatani, Tetsuya Ogata and Hiroshi G. Okuno
Graduate School of Informatics, Kyoto University, Japan
{angelica,mizumoto,kenzo,ohtsuka,tall,komatani,ogata,okuno}@kuis.kyoto-u.ac.jp

Abstract— **Musicians often have the following problem: they have a music score that requires 2 or more players, but they have no one with whom to practice. So far, score-playing music robots exist, but they lack adaptive abilities to synchronize with fellow players' tempo variations. In other words, if the human speeds up their play, the robot should also increase its speed. However, computer accompaniment systems allow exactly this kind of adaptive ability. We present a first step towards giving these accompaniment abilities to a music robot. We introduce a new paradigm of beat tracking using 2 types of sensory input – visual and audio – using our own visual cue recognition system and state-of-the-art acoustic onset detection techniques. Preliminary experiments suggest that by coupling these two modalities, a robot accompanist can start and stop a performance in synchrony with a flutist, and detect tempo changes within half a second.**

I. INTRODUCTION

Since the early 1980's, computer accompaniment programs have served as virtual musical partners for musicians around the world ([1][2][3], to name a few). These programs are more than minus-one CD players. They listen to a human's input via MIDI or a microphone, and adapt accompaniment music to match the soloist's score location and speed. For example, if the soloist plays faster, the accompaniment program also plays faster. We call this synchronization, and human musicians seem able to do this naturally.

In recent years, we've witnessed the next generation of synthetic musical partners: music robots. Humanoid music robots such as in [4][5][6] add a new dimension to computer music, allowing real acoustic instruments such as flute, theremin and piano to be played. In addition, embodiment such as head and arms, and built-in sensors such as cameras and microphones provide new interfaces for interaction. However, current music robots still lack the capability that the tried-and-true computer accompaniment programs have—to play an accompaniment or duet score with human-like synchronization.

Clearly, to give music robots this synchronization ability, we should draw on the knowledge gathered from computer accompaniment. Two possible approaches to computer accompaniment include score following and beat tracking. A score following accompanist listens to the soloist's notes and attempts to "follow" along with the soloist's score to localize itself within the piece. The second approach, beat tracking (e.g. [7][8][9]), does *not* require prior knowledge of the soloist score. Instead, beat trackers extract beats in the music, similar to a human tapping their foot, which lets it predict when the next beat will occur.

So far, few music robot systems [10] have implemented score following, though several robot systems use beat tracking for real-time interaction. Weinberg et al.'s interactive drum robots [11][12] use sophisticated beat trackers based on energy for both pitched and non-pitched percussive instruments. Murata et al.'s [13] robot system sings along to the beat of pop music. Goto's [14] beat tracking system does not require drum sounds, but uses a combination of note onsets and chord changes. To summarize, none of these beat tracking systems work with solo instruments such as violin or flute, whose drumless acoustic signals are more difficult to segment and track.

In this work, we create a robot accompanist that can perform simple beat tracking for this special class of non-percussive monophonic instruments; in particular, the classical flute. To achieve this, we take a different approach to all previous accompaniment systems which rely on audio input only. Music studies [15][16] suggest that human ensemble players both listen to and watch fellow players for temporal coordination. In fact, one study on conducting [17] suggested that visual cues were as important as audio cues in keeping musical synchrony. Therefore, we use a combination of audio note onsets and our new visual beat cue paradigm [18] to predict instantaneous tempo. In this way, by listening and watching a human flute player, our theremin-playing robot accompanist can synchronize its play quickly.

A. Other Related Work

A few interactive music systems also integrate both audio and video modalities. For example, one multi-modal gestural system [19] tracks a flutist through audio and video, and plays back pre-recorded tracks when it detects certain cues such as when the flutist "points the flute downward and plays a low B". However, this doesn't seem to generalize to pieces other than those specifically composed for the system. The Shimon Interactive Marimba player [20] performs beat tracking while gazing at performers to indicate its interest and nodding its head to the beat. Human players use these visual cues to adapt to Shimon's play, as opposed to the robot adapting to the human. The Waseda flute and saxophone robot group also uses vision to control parameters such

as vibrato or tempo [21]. In [22], they perform rhythmic tracking in a call-and-answer context by comparing audio histograms to those stored in a database. However, this turn-based, asynchronous play is not applicable to the traditional accompaniment system where both players play at the same time.

II. A ROBOT ACCOMPANIST SYSTEM

Our robot accompanist uses audio and video to synchronize where applicable, from the beginning to the end of a piece. Our robot accompanist system can:

(1) Set its initial tempo by listening to human's beat cues (e.g. "one, two, three, four")
(2) Start playing when it sees a visual cue
(3) Change its tempo by seeing visual beat cues and listening to the flutist's notes
(4) End a held note (i.e. fermata) when visually indicated

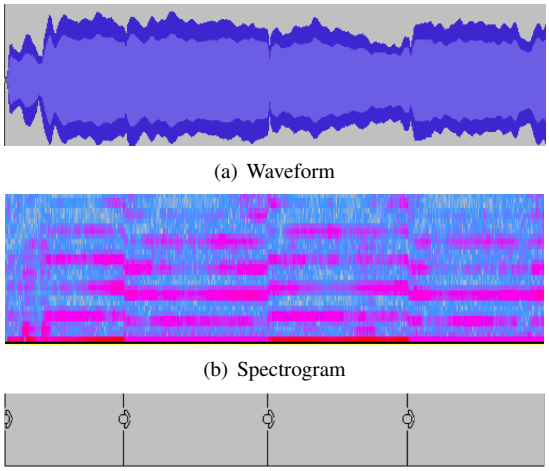
It uses audio and video in a complementary fashion. For initial tempo setting (1), it uses audio input only; to detect inter-player start and end cues (2,4) it uses vision only; and to change tempo (3) it uses both audio and video. We test the system on a robot thereminist, though this system is modular and can be placed on most ethernet-enabled musical robots. In this section, we first outline the note onset detection technique we chose for beat candidate extraction. Next, we give a brief review of our visual cue recognition algorithm. We then describe our technique for fusing these two sources of beat information. Finally, we provide an overall view of the robot system.

A. NOTE ONSET DETECTION

A note onset can be loosely defined as the beginning of a played note. In classical music, beats often coincide with the beginning of notes (e.g. on the quarter notes in a 4/4 piece). Thus, if we detect note onsets, we may have a set of possible beats in a musical performance.

To use note onset detection within our system, several constraints must be considered. First of all, the onset detection scheme must be fast, with a low complexity, to be reactive enough for a real-time performance. Secondly, in order for the system to play in musical ensembles containing string or woodwind instruments, note onset detection techniques should be sensitive to soft tonal onsets, such as those produced by a violin or flute (see Fig. 1(a)). Energy-based methods [23], which measure changes in volume to detect beats, are not sufficient unless percussive instruments (such as piano or drums) are used. We choose our onset detection techniques with these considerations in mind.

We informally tested several onset detection functions, all described in [24], to see which would work best with flute sounds. High Frequency Content [25], which detects large changes in the high frequency components of the spectrum, seemed to work well with notes with an explosive onset (i.e. attacked or tongued notes). However, it could not detect smooth legato note changes, such as those pictured in Fig. 1. As can be guessed by observing the spectrum in Fig. 1(b), calculating the frame-to-frame difference in spectral magnitude or "Spectral Difference" [26] is a typical way to address the problem of legato changes. A second technique for detecting these so-called tonal onsets is measuring temporal instabilities in spectral phase; this "Phase Deviation" [27] is useful for detecting situations where the spectral magnitude may remain similar but phase stability is perturbed, such as playing the same note twice in a row. We decided to use Complex Domain Difference [28], which looks for differences in both spectral magnitude and phase in the complex domain. As expected, this method seemed to detect both tongued notes as well as smooth note onsets. It should be noted, however, that as mentioned in [24], any method tracking changes in phase (including Phase Deviation and Complex Domain) is sensitive to noise, which we experienced when implementing this on robot audio setups of lower quality.



(a) Waveform

(b) Spectrogram

(c) Onsets detected using Complex Domain Difference

Fig. 1. Four notes played on flute with legato onsets

We use the Aubio onset detection library [29] implementation of Complex Domain, which measures differences between frames using the Kullback Leibler distance [30]. Implemented in C with a dynamically thresholded peak picker, it can fulfill our real-time requirements. As a first step, we equip the flute player with a lapel microphone for input into this sound processing step, thus removing the need for sound separation. Ideally, the robot's built-in microphone should be used, with sound separation or frequency filtering to isolate the flute part.

Note onsets can represent beats in a few different ways. At the beginning of a piece, to set an initial tempo, the human can say "one, two, three, four", and a tempo can be set using the same note onset scheme described here. Tempo can be deduced simply by taking the average interval between word onset times. Mid-song, however, we detect note onsets that may or may not represent beats. For example, in Fig. 2, we show an excerpt from Pachelbel's Canon in D, where note onsets do not have a one-to-one correlation with beats; here, notes occur twice per beat. We also need to account for spurious detections of onsets which may arise from imperfect

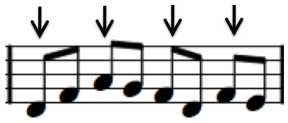
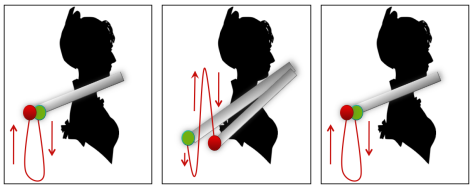


Fig. 2. One bar from Pachebel's Canon in D. Eight note onsets would be detected, though only 4 beats, here represented by arrows, should be considered for beat tracking.



(a) Beat Cue (b) Start Cue (c) End Cue

Fig. 3. Trajectories of flute visual cues

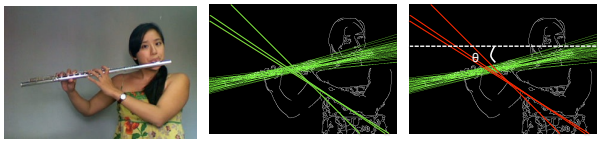
acoustic processing. Therefore, we need a method to improve our estimations of beat times—we use our visual cue method described next to perform this function.

B. DETECTING VISUAL CUES

Human musicians naturally use visual cues such as eye contact and instrument movement to coordinate with fellow ensemble players, similar to conductors who use their batons to indicate beats. In fact, a study on clarinetist's movements found that "movements related to structural characteristics of the piece (e.g. tempo)" [31] were consistently found among player subjects. Movements included "tapping of one's foot or the moving of the bell up and down to keep rhythm." Although we do not claim that all musicians use movements when performing, we believe that identifying common, natural gestures is a starting point to using vision as a human-robot interface.

Based on empirical observation, it appears that flutists also manifest a similar sort of up and down movement of the flute to keep rhythm. This movement, depicted in Fig. 3(a), is the one which we exploit here for beat tracking purposes. Aside from rhythmic beat movements, flute gestures may also be used within an ensemble to indicate the start of a passage (Fig. 3(b)), or the end of a held note (Fig. 3(c)). We refer to this set of movements as *visual cues* for synchronization.

To detect these visual cues using a robot's camera, we position the flutist to face the robot, producing input images such as Fig. 4(a). We can then locate and track the flute to



(a) (b) (c)

Fig. 4. (a) Original input image, (b) processed image with detected Hough lines and (c) outliers marked in red, with the flute angle to track in white

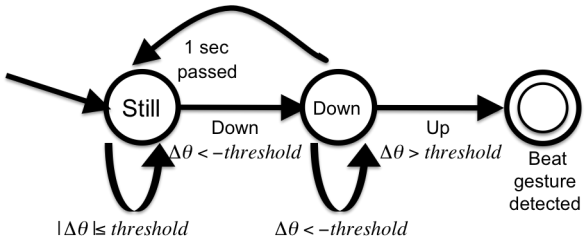


Fig. 5. State machine to detect Beat Cue

recognize the three visual cues described earlier. Our simple method uses the Hough Transform algorithm to locate the straight flute throughout a stream of input images.

1) Hough Line Detection: We first perform Canny edge detection [32] and the Hough Transform [33] on each image. This outputs multiple lines with approximately the same angle of the flute, as shown in Fig. 4(b).

2) Outlier Line Pruning: Spurious lines may also be detected, due to background clutter or patterns on clothing. Thus, we use the RANSAC algorithm [34], a popular outlier detector, to prune these unwanted lines. Once outliers are pruned, we extract the remaining lines' mean angle θ to get the estimated orientation of the flute, as depicted in Fig. 4(c), and input it into gesture recognizers.

3) Finite State Machine: At each time step, we determine the instantaneous change in θ (derived from the image processing stage) between two subsequent video frames F at time $t-1$ and t.

$$\Delta\theta = \theta(F_t) - \theta(F_{t-1}) \tag{1}$$

To recognize the visual beat cue, we input $\Delta\theta$ into the finite state machine (FSM) depicted in Fig. 5. In this case, the FSM state with respect to $\Delta\theta$ is defined as follows.

$$STATE(\Delta\theta) = \begin{cases} DOWN & \text{if } \Delta\theta < -threshold \\ UP & \text{if } \Delta\theta > threshold \\ STILL & \text{otherwise} \end{cases} \tag{2}$$

The *threshold* acts as a rudimentary smoother, and a DOWN state means the end of the flute is moving downwards, and so on. By defining an FSM for each of the three types of gestures depicted in Fig. 3, we may recognize not only when the flute player moves their flute to the beat, but also when they make start and end visual cues.

The accompanist system only searches for start and end cues during appropriate places in the score. However, it tries to detect visual beat cues continuously throughout a piece. Next, we describe how we use these detected visual beat cues along with note onset information to extract an instantaneous tempo.

C. PERCEPTUAL MODULE: AUDIO & VISUAL BEAT MATCHING

Our perceptual module attempts to find matches between audio onset events with visual cue events. By relying on two sources of information, we can have a satisfactory level of confidence that a beat was detected.

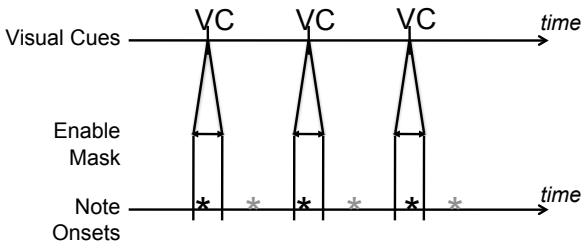


Fig. 6. Our audio-visual matching scheme. Visual cues act as an enabler; detected note onsets which fall into a pre-specified range around visual cues are considered as matched beats.

We make the following assumptions. First, a human player makes beat gestures on two consecutive beats, and also plays notes on those beats. This is not too uncommon, especially in the cases where a player insists a tempo by attracting attention with a visual cue. Secondly, we know the approximate tempo we are looking for, based on the current tempo. This is consistent with how humans play - they do not, for example, double their speed suddenly, unless it is already marked in the score. Finally, we assume that instantaneous tempo can be expressed as the latest Inter-Onset Interval (IOI) detected, the time between the start of the two most recent consecutive beats.

Our algorithm for IOI extraction works as follows. Let V and A respectively be sets containing previously observed video and audio cue events, M be a temporally ordered list of *matched beat* times, δ_1 be the maximum offset between a matched audio and video cue, the current tempo IOI be IOI_c, and δ_2 be the tempo change threshold. Whenever an audio or visual cue event at time e is detected at time t_e, we run this function to return an instantaneous tempo IOI if applicable.

if e is $audio$ **and** $\exists v \in V, |t_e - t_v| < \delta_1$ **then**
  $M \leftarrow M + t_e$
  if $|S| \geq 2$ **and** $||M[last] - M[last-1]| - IOI_c| < \delta_2$ **then**
    return $M[last] - M[last-1]$
else
  if e is $video$ **and** $\exists a \in A, |t_e, t_a| < \delta_1$ **then**
    $M \leftarrow min(\{t_a | a \in A, |t_e - t_a| < \delta_1\})$
    if $|S| \geq 2$ $||M[last] - M[last-1]| - IOI_c| < \delta_2$ **then**
      return $M[last] - M[last-1]$

Visual beat cues simply act an enable mask (see Fig. 6) with a window width of $2 * \delta_1$, and a *matched beat* corresponds to the note onset event that falls within that window. We experimentally set our threshold here to 150 ms, which gives a detection window of 300 ms around each visual beat cue. If more than one audio note onset is detected within this window, the first onset is chosen - the earliest onset detected. Notice that the final IOI is determined solely by the audio note onset times. Visual cue timing is not used in the final IOI calculation because audio has a much higher sampling rate, and is thus more precise. Whereas audio has a typical sampling rate of 44100 samples per second, video camera frame rates are on the order of only 30 frames per second.

(a) 175 lux (b) 100 lux (c) 50 lux

Fig. 8. Actual input images from robot's camera for our three experimental conditions.

In order for this simple fusion algorithm to be valid, a highly precise timing scheme is essential. The Carnegie Mellon laptop orchestra [35] used a central hub from which laptop instruments queried the current time. We decided to use Network Time Protocol [36] to synchronize the clocks of all our modules, some of which were connected through ethernet. In addition to precise clock synchronization, this event-driven formulation of the algorithm is required because the data from two data sources may not arrive in sequence, due to delays in network data transfers.

D. SYSTEM OVERVIEW

This system was implemented for the HRP-2 theremin-playing robot first introduced in [5]. Fig. 7 overviews the accompaniment system. The HRP-2's Point Grey Fly camera is used to take greyscale images at 1024x728 resolution, at a maximum of 30 fps. When start and end cues are detected from the vision module, these commands are sent to the theremin robot to start a piece or end a held note, depending on the robot's current location in the score. A 2.13 GHz MacBook with an external microphone was used as our note onset detection module. If there is no current tempo (such as before starting a piece), the tempo detection module uses audio onsets only to derive an initial tempo. Otherwise, it attempts to match input from its two input modalities within the perceptual module, and sends on detected tempos to the theremin player.

III. EXPERIMENTS AND RESULTS

We performed two experiments to determine the viability of our accompaniment system. The first experiment attempts to evaluate the start and end cues of our visual cue recognition system. The second evaluates the combination of note onset detection augmented with visual beat cues for tempo detection.

A. Experiment 1: Visual Start and End Cues

We evaluate the accuracy of our gesture recognizer by recording its output given 30 samples of the start and end gesture, performed at 3 different brightness levels, as shown in Fig. 8. These gestures were performed by an intermediate-level flutist familiar with the gestures as depicted in Fig. 3. The results are shown in Table 1.

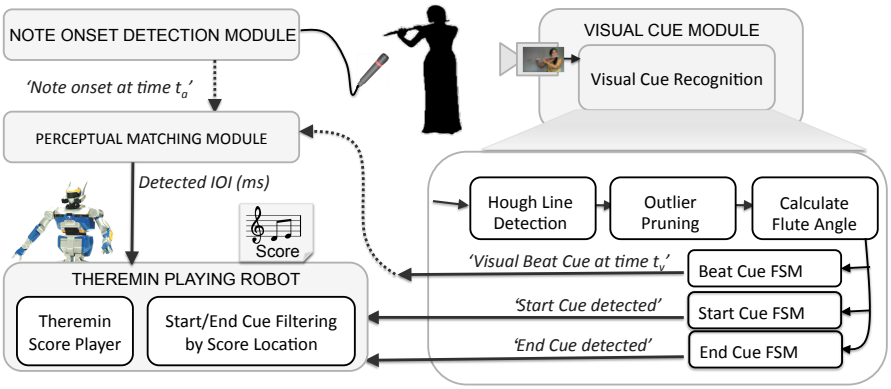


Fig. 7. Overview of our robot accompanist system

Visual Cue to Detect	175 lux	100 lux	50 lux
Start Cue (%)	97	100	83
End Cue (%)	100	97	100

TABLE I
RECOGNITION RATES OF EACH TYPE OF GESTURE (PRECISION).

B. Experiment 2: Audio Onset Detection + Visual Beat Cues

In this experiment, lighting was fixed at 175 lux. The same flute player, equipped with a lapel microphone, played two legato notes in alternation, with no tonguing: A2 and B♭2, the same legato notes shown in Fig. 1. With each change in note, the flutist performed a visual beat cue. A secondary observer, a classically trained intermediate-level clarinet player, tapped a computer key along with the changes in notes to provide a human-detected tempo (measured in IOI) for comparison.

Visual and Audio Beat Detections: Fig. 9 shows the resulting timeline of our experiment. Over 75 notes played, the system detected 75 visual beat cues correctly, 3 false positive note onsets, 3 false negative note onsets, and 72 matched beats. The average error between our system and the human detected IOI was 40 ms, so we can conclude that our system detects tempo comparably to humans in real-time. The remaining error may be explained when considering the relative errors as a histogram: Fig. 9 shows a Gaussian-like distribution similar to white noise. Humans tap with timing error patterns similar to white/pink noise [37], so further experiments not involving humans for ground truth may be needed for more precise measurements.

Tempo Change Delay: Previous beat-tracking methods extract a tempo based on a history of past notes, for example, using cross-correlation. For example, Murata's beat tracker [13] required 2 seconds to change tempo, due to taking 1 second windows for its pattern matching method. If our matching algorithm can detect instantaneous tempos in less than 2 seconds, it may be used as a fast tempo initializer during these precious seconds of unsynchronized play.

We have found that our method may be useful for changing tempos within half a second. In our experiment, a tempo IOI is calculated whenever two consecutive matched beats are detected. We calculate the average delay in tempo change by finding the difference between the time this second audio beat was input into the microphone, and when the internal tempo of the robot accompanist was changed. In our preliminary experiments, the average delay was 231 ms.

C. Discussion

An interesting observation noted by our human observer was that he watched the visual cue to predict the beat onset. This may imply that we should track visual cues with a higher temporal resolution, and try to predict the visual onset before it happens, rather than use it in hindsight. One weakness noted while using the system is that the accuracy of the tempo detection is largely dependent on the flutist's proficiency. The matching threshold may need to be widened to compensate for precision error. Future work should include experimentally setting these thresholds based on usage by multiple participants of varying expertise.

In the case of widening the threshold, our audio-visual technique would be limited to slower passages, with few notes. As noted in [31], musicians typically stop all movement during highly technical passages, and increase motions during easier parts of the score. In future work, we would hope that we can offset this lack of visual stimulus by taking advantage of the rhythmic nature of many notes in a short period of time.

IV. CONCLUSION AND FUTURE WORK

Our ultimate goal is to create a robot that can play music with human-like expressiveness. As a first step, we have allowed a theremin-playing robot to listen and watch a human, to mimic timing and speed in the context of a duet. By using both audio and visual cues, it can synchronize with a human flutist within half a second.

In the future, these audio and visual capabilities may lay the foundation for more interesting applications. For example, a robot could learn by demonstration. By watching and listening a human perform, a robot musician may learn how to make gestures that correspond musically with the music it plays. Or, it may learn how to play music expressively not only by mimicking a human's pitches and rhythms, but

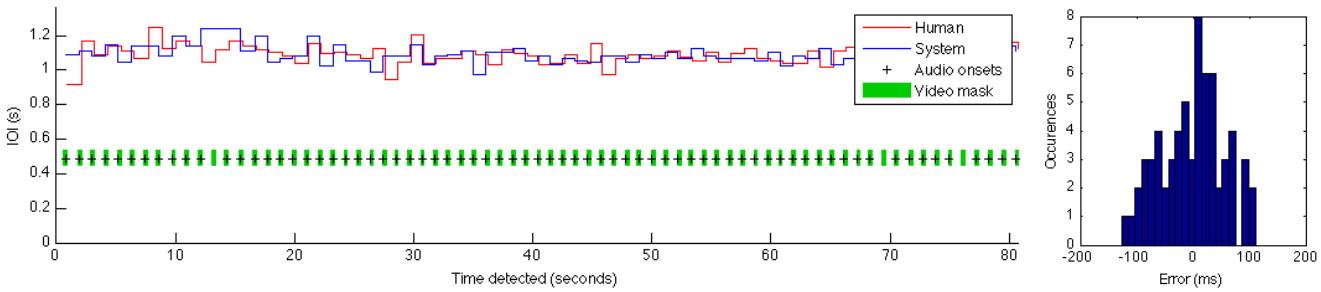


Fig. 9. (Left) Experiment timeline: Over 75 notes played, both human and system detected tempos (IOI) remained around 1.1 s; average absolute error between human and system is 40 ms. (Right) A histogram of deviation between the human and system shows that most errors were less than 100 ms.

also minute volume and tempo variations. Our short-term future work includes using the robot's built-in microphone as opposed to an external mic and integrating sound separation to track multiple instruments simultaneously. Other possibilities include extending visual cue recognition to other instruments, or implementing a form of score following, as opposed to beat tracking, for monophonic instruments.

V. ACKNOWLEDGMENTS

This work was supported by GCOE and KAKENHI.

REFERENCES

[1] R.B. Dannenberg, "An On-Line Algorithm for Real-Time Accompaniment," *Proceedings of ICMC*, 1984, pp. 193-198.
[2] B. Vercoe and M. Puckette, "Synthetic Rehearsal: Training the Synthetic Performer," *Proceedings of ICMC*, 1985, pp. 275-278.
[3] C. Raphael, "Synthesizing Musical Accompaniments with Bayesian Belief Networks," *Journal of New Music Research*, vol. 30, 2001, pp. 59–67.
[4] K. Chida et al., "Development of a New Anthropomorphic Flutist Robot WF-4." *Proceedings of ICRA*, 2004, pp. 152-157.
[5] T.Mizumoto et al., "Thereminist Robot: Development of a Robot Theremin Player with Feedforward and Feedback Arm Control based on a Theremin's Pitch Model", *Proc. of IROS*, 2009.
[6] I. Kato et al., "The robot musician wabot-2," *Robotics*, vol. 3, Jun. 1987, pp. 143-155.
[7] M.E. Davies et al. "Beat tracking towards automatic musical accompaniment," *Proceedings of the Audio Engineering Society 118th convention*, 2005.
[8] P. Toiviainen, "An interactive MIDI accompanist.," *Computer Music Journal*, vol. 22, Winter 98. 1998, p. 63.
[9] P.E. Allen and R.B. Dannenberg, "Tracking Musical Beats in Real Time," *Proceedings of ICMC*, 1990, pp. 140–143.
[10] T. Otsuka et al., "Incremental Polyphonic Audio to Score Alignment using Beat Tracking for Singer Robots," *Proceedings of IROS 2009*, pp.2289-2296
[11] G. Weinberg et al., "Musical Interactions with a Perceptual Robotic Percussionist." *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication*, 2005.
[12] G. Weinberg, S. Driscoll, "The Design of a Robotic Marimba Player - Introducing Pitch into Robotic Musicianship", *Proceedings of NIME*, 2007, pp. 228-233.
[13] K. Murata et al., "A beat-tracking robot for human-robot interaction and its evaluation," *Proceedings of Humanoids 2008*, 2008, pp. 79-84.
[14] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," *Journal of New Music Research*, vol. 30, no. 2, pp. 159-171, 2001.
[15] W. Goebl and C. Palmer, "Synchronization of Timing and Motion Among Performing Musicians," *Music Perception*, vol. 26, 2009, pp. 427-438.
[16] K. Katahira et al., "The Role of Body Movement in Co-Performers' Temporal Coordination", *Proceedings of ICoMCS* December, 2007, p. 72.
[17] W.E. Fredrickson, "Band Musicians' Performance and Eye Contact as Influenced by Loss of a Visual and/or Aural Stimulus," *Journal of Research in Music Education*, vol. 42, Jan. 1994, pp. 306-317.
[18] Lim et al., "Robot Musical Accompaniment: Integrating Audio and Visual Cues for Real-time Synchronization with a Human Flutist", *Proceedings of IPSJ*, 2010
[19] D. Overholt et al., "A multimodal system for gesture recognition in interactive music performance," *Computer Music Journal*, vol. 33, 2009, pp. 69-82.
[20] G. Weinberg et al., "Interactive jamming with Shimon: a social robotic musician," *Proceedings of HRI*, 2009, pp. 233?234.
[21] K. Petersen et al., "Development of a Real-Time Instrument Tracking System for Enabling the Musical Interaction with the WF-4RIV," *IROS 2008*, pp. 313-318
[22] K. Petersen et al., "Development of a Aural Real-Time Rhythmical and Harmonic Tracking to Enable the Musical Interaction with the Waseda Flutist Robot," *Proceedings of IROS 2009*, pp. 2303-2308
[23] A.W. Schloss, "On the Automatic Transcription of Percussive Music - From Acoustic Signal to High-Level Analysis." *PhD thesis, Department of Hearing and Speech, Stanford University*, 1985.
[24] J. Bello et al., "A Tutorial on Onset Detection in Music Signals," *Speech and Audio Processing, IEEE Transactions on* vol. 13, 2005, pp. 1035-1047.
[25] P. Masri, "Computer Modeling of Sound for Transformation and Synthesis of Musical Signal," *Ph.D. dissertation, Univ. of Bristol*, Bristol, U.K., 1996.
[26] J. Foote and S. Uchihashi, "The beat spectrum: a new approach to rhythm analysis," *Proceedings of the IEEE ICME* 2001, pages 881-884, 2001.
[27] J. Bello et al., "Phase-based note onset detection for music signals," *Proceedings of the IEEE ICASSP*, 2003, pages 441-444.
[28] C. Duxbury et al., "Complex domain onset detection for musical signals," *Proceedings of DAFx*, 2003, pages 90-93.
[29] P.M. Brossier, "Automatic Annotation of Musical Audio for Interactive Applications." *Ph.D Thesis, Centre for Digital Music Queen Mary, University of London*, 2006
[30] S. Hainsworth and M. Macleod, "Onset detection in music audio signals," *Proceedings of the ICMC*, pages 163-166, 2003.
[31] M. Wanderley et al., "The Musical Significance of Clarinetists' Ancillary Gestures: An Exploration of the Field," *Journal of New Music Research*, vol. 34, 2005, pp. 97-113.
[32] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. on Pattern Analysis and Machine Intell.,* vol. 8, 1986, pp. 679-698.
[33] R.O. Duda and P.E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, 1972, pp. 11-15.
[34] R.C. Bolles et al., "A RANSAC-based approach to model fitting and its application to finding cylinders in range data" *Proc. of IJCAI*, 1981, pp. 637-643.
[35] R.B. Dannenberg et al., "The Carnegie Mellon Laptop Orchestra," *Proceedings of ICMC, 2007*, pp. 340-343.
[36] D. Mills, "Network Time Protocol (Version 3) specification, implementation and analysis", *RFC 1305*, 1992.
[37] D.L. Gilden, T. Thornton, and M.W. Mallon, "1/f noise in human cognition," Science, vol. 267, 1995, p. 1837.