Multimodal gesture recognition for robot musical accompaniment*

*Angelica LIM, Takeshi MIZUMOTO, Louis-Kenzo CAHIER, Takuma OTSUKA, Toru TAKAHASHI, Tetsuya OGATA, Hiroshi G. OKUNO (Kyoto University)

Abstract— Listening and watching are important skills for accompanists to play in time with fellow musicians. By detecting subtle visual cues and listening to other players, musicians can start together, stop together, and follow faster or slower changes in tempo. In this paper, we formalize this non-verbal language for the case of flutists, and describe how our thereminist robot accompanist system detects them. Initial experiments show over 83% detection rates for our 3 types of visual cues. Additionally, by coupling visual cues and acoustic beat detection, the robot can extract tempo in less than half a second.

Key Words: music-playing robots, gesture recognition, multimodal integration, robot accompanist

1. Introduction

Can robots understand the hidden meaning behind our words? In daily life, we wave to say hello, shake our head to show disagreement, and give a thumbs up to approve. These non-verbal movements are called gestures, and they convey important meaning. In addition, we may speak fast and loudly when irritated, and slowly and softly when timid. These audio features also help convey a message. Clearly, if robots could react appropriately to these subtle visual gestures and acoustic cues, interaction would be more natural, smooth, and human-like.

Music-playing robots such as [1] [2], are a perfect test bed for processing both gestural and aural information in real-time. Consider that even amateur musicians possess a very important musical skill: they must listen and watch co-players, continuously adapting to match their play. Fredrickson [3] showed that band musicians synchronize best by both watching the conductor and listening to their co-players. Even without a conductor, musicians still communicate visually. Piano duet studies such as [4] found that head movements, exaggerated finger lifts and eye contact are used to communicate synchronization events between players. Can we give robot musicians these same visual and acoustic sensing abilities?

Some robots can already listen to the beat. For example, Georgia Tech's HAILE drum robot [5] detects human drumbeats using energy-based beat trackers. Using the beats, it can detect speed and perform improvisation accordingly. In [6], a robot can listen to pop music and sing along to the beat. All of these systems track percussive beats, where large changes in volume indicate the starts of the notes. As we will see later, processing the acoustic cues of continuous instruments such as violin or flute is a more difficult task.

*This research is partially supported by Kakenhi (S) and InfoExplosion.

Fig.1 Thereminist robot plays while adapting to co-player's visual and audio cues.

Visual cues in music have been less researched. The Shimon interactive marimba player [7] looks at fellow human players to indicate solo changes, acting as a leader of the ensemble. However, it cannot yet detect the same movements when humans take the lead role. Waseda's flute and saxophone robots [8] can change volume and vibrato by detecting linear movements of a human saxophone player. Similarly, a multi-modal accompaniment system [9] tracks a flutist through audio and video, automatically playing back pre-recorded music such as when the flutist points the flute downward and plays a low note. What is missing from the latter two approaches is that the detected cues are not based on naturally occurring gestures. In our present work, we first try to define the musical equivalent of "waving hello" for three musical situations:

1) starting the piece
2) ending a held note
3) changing tempo

In this paper, we describe a Theremin-playing robot accompanist (Fig. 1) that can:

- recognize 3 types of visual gestures,

- detect audio cues known as note onsets,
- and extract a human's playing speed by fusing the information.

In particular, we treat the case of flutists' instrument movement and acoustic fingerprint, though it is hoped that multimodal gesture recognition could be extended to other instruments and perhaps other fields.

2. A robot accompanist

Our robot accompanist uses audio and vision in a complementary fashion. It uses visual cues to start and end in sync with the human, and a combination of both audio and video to adapt to tempo changes. In the following sections, we first describe our visual cue recognition algorithm. We then outline our note onset technique for audio processing. Finally, we give a technique for fusing these two sources of information in real-time.

2·1 Visual Module

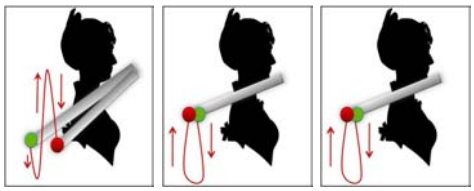
The first step to detecting gestures is to define the problem: which gestures are performed, and how? A study on clarinet players' movements [10] found that players move the bell up and down to keep rhythm. Based on our empirical observation, flutists also seem to move their instrument up and down to the beat, so we identified distinct three gestures, described below. We do not claim that all flutists use these movements when performing, but we believe that trying to identify these common, symbolic gestures is a starting point to using vision for natural robot interaction.

The *start cue* is used to synchronize the first note of a piece or musical section. We define it as a down-up-down movement of the far end of the flute (Fig. 2a). It is preceded by a lack of motion, while making eye contact with the rest of the ensemble players.

The *end cue* is used by the leader to "cut off" a held note. For example, the final note of a score is often notated with a fermata, and the leader must indicate when all players should stop. Among flutists, the stop cue is a circular motion of the end of the flute. We define it here simply as a down-up motion, when viewed from the front (Fig. 2b). This gesture is also preceded by a lack of motion while playing the held note.

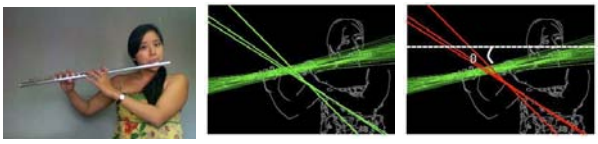
The *beat cue* is used to indicate rhythmic beats in the music. From beat cues, we can infer tempo; i.e., if the leader performs beat gestures closer together, players should speed up. We define it as a down-up motion of the flute (Fig. 3b). As opposed to the end gesture, we do not assume it is preceded by stillness.

To detect these three gestures, it is natural to track the flute itself. We position the flutist in front of our robot's camera, as shown in Fig. 3(a). The system then locates the flute using a line-detection technique called the Hough Transform, which detects the straight flute throughout a stream of video images (Fig. 3(b)). Because the classical flute is characterized by many key-connecting rods, the resulting output is multiple lines with approximately the same angle of the flute. In addition to these set of lines, some background clutter or clothing may produce spurious lines. We use the RANSAC outlier detection algorithm [11] to prune these unwanted lines, as shown in Fig. 3(c). By extracting the mean angle θ of the remaining set of lines, we can approximate the position of the flute. This localization is performed at each time step, to determine in which direction the flute moved. Formally, we determine the instantaneous change in θ between the previous video frame F at time $t-1$, and the current frame at time t.



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

Fig.2 Trajectories of flute visual cues



(a) (b) (c)

Fig.3 (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

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

From this $\Delta\theta$ we can decide which state the flute is currently in: down, up, or still.

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

The *threshold* ensures that the change in flute angle is sufficiently large, to ignore small player fluctuations. Three state machines (corresponding to start, end, and beat cues) track the flute's current state in parallel. These finite state machines (FSM) are shown in Fig. 4. Notice that beat and end cues FSM appear identical; to overcome this similarity, we add a constraint that end cues require a minimum amount of time in the "still" state, whereas beat cues do not.

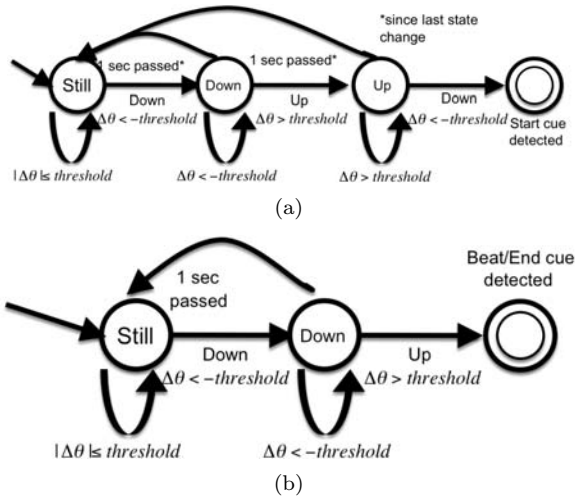


Fig.4 FSMs for start cue (a) and end/beat cues (b).

Every time the robot detects a cue, the system decides whether or not to use the information, based on context. In particular, we filter cues with respect to the current score location. Start cues only control the robot accompanist at the start of the piece, and end cues only when the note is currently being held. Beat cues are valid throughout the piece, so to avoid overdetection, we check for concurrent audio cues, which will be described next.

2·2 Audio Module

As mentioned previously, changes in tempo correspond to beats spreading further apart or closer together. How do we detect these beats? One way is to detect the start of played notes, known as note onsets. Since beats often coincide with note onsets, we can derive a set of possible beats by performing note onset detection.

Several note onset detectors exist, but we require that a) it be fast enough for real-time performance, and b) it can detect soft tonal onsets, such as those produced by a violin or flute. The technique we chose is known as "Complex Domain Difference" [12], which looks for differences in both a sound's spectral magnitude and phase in the complex domain. The result is that both attacked notes (a change in power) and legato notes (perturbation in phase) are detected. We use the Aubio onset detection library [13] which is implemented in C and can fulfill our real-time requirements.

2·3 Fusion Module

The fusion module perceives tempo changes by looking for simultaneous matches between visual beat cues and audio beat cues. When beat cues are detected simultaneously from both modalities, we can reliably say that ensemble leader was trying to indicate a beat (and therefore probably wants to change tempo). As shown in Fig. 5, visual cues act as an enabler. Using an enable mask of $\delta_1 = 150$ ms around each visual cue, the system attempts to align note onsets. Once two alignments (matched beats) are detected, their difference, known as Inter-Onset-Interval (IOI), is calculated, giving us an instantaneous tempo. We assume that the leader will not make sudden, large tempo changes, so if the tempo change is less than a given threshold δ_2, we accept this tempo change and the new tempo is sent to the robot's playing module.

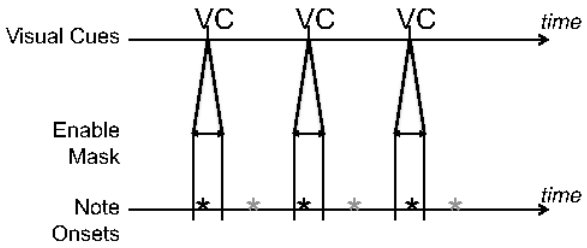


Fig.5 Our audio-visual matching scheme. Visual cues act as an enabler; detected note onsets which occur within +/- δ_1 around visual cues are considered as matched beats.

Our fusion algorithm for tempo (IOI) detection is as follows. Let V and A respectively be the sets of previously observed video and audio cue events, M be a temporally ordered list of matched beat times, δ_1 be the maximum difference between a matched events in V and A, the current tempo IOI be IOI_c, and δ_2 be the tempo change threshold. In our event driven formulation, whenever an event e from V or A is detected at time t_e, we run the following function to return a new tempo IOI if applicable:

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

In lines 2 and 7, we can see that only timings from audio events are used to estimate the tempo. This is because audio is sampled at a very high rate (44100 samples per second) and video only at 30 frames per second.

An essential point in making this fusion scheme work is our use of Network Time Protocol (NTP) [14]. A small lag of even 100 ms can affect tempo greatly. NTP synchronizes to the millisecond the clocks of all our modules, which were connected through Ethernet.

3. Experiments

We implemented our system for the HRP-2 theremin playing robot first introduced in [1]. Grayscale images were taken at 1024x728 resolution at 30 fps using the robot's built-in camera. A 2.13

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

Fig.6 Actual input images from robot's camera for our three experimental conditions.

Table 1 Recognition rates of each type of gesture.

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

GHz MacBook performed the note onset detection, with an external microphone clipped to the flutist's lapel. In our preliminary experiments, only one intermediate flutist was used as a participant. Readers should thus take care in interpreting the results; further experiments involving more flutists are needed.

In our first experiment, we evaluated the accuracy of the start and end cue recognition module. The flutist performed 30 instances of each gesture at 3 different brightness levels, as shown in Fig. 6. Results are shown in Table 1.

In the second experiment, we attempted to evaluate our fusion module's tempo change detection accuracy, compared to a human. The flutist played 2 legato notes in alternation, and with each change in note, performed a visual beat cue. Between each successive beat, we expect that a new IOI should be detected. A secondary observer tapped a computer key along with the changes in notes.

Over 75 beats played, 75 visual beat cues were correctly detected. However, 3 false positive note onsets and 3 false negative note onsets were detected, resulting in 72 matched beats. The average error between our system and the human-detected IOI was 40 ms, so we can say that our system detects tempo comparably to humans.

As for tempo change delay, we define a tempo change as occurring once the 2nd beat has occurred. Between the time each second beat was input into the microphone, and the new tempo was set in the robot, the average delay was 231 ms. With a delay of less than half a second, our method appears sufficiently fast for real-time applications.

4. Future Work

During our experiments, we noticed that the human observer used the visual cue to predict the beat onset. In the future, perhaps vision could be used to predict the event before it happens, instead of using it in hindsight as we did here. Another major direction is to give expressiveness to the robot. As it stands, the robot must follow the lead performer as closely as possible. If the robot had its own sense of timing, it could anticipate timing changes independently, only needing to synchronize slightly with the human. Further experiments with many players should also be performed, and the lapel microphone should be replaced by the robot's internal microphone. Other instruments such as clarinet or violin could also be investigated.

References

[1] T. Mizumoto, H. Tsujino, T. Takahashi, T. Ogata, and H. G. Okuno, "Thereminist robot : development of a robot theremin player with feedforward and feedback arm control based on a theremin's pitch model," in *IROS*, pp. 2297–2302, 2009.

[2] J. Solis, K. Chida, K. Taniguchi, S. M. Hashimoto, K. Suefuji, and A. Takanishi, "The Waseda flutist robot WF-4RII in comparison with a professional flutist.," *Computer Music Journal*, vol. 30, no. 4, pp. 12–27, 2006.

[3] 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, pp. 306–317, Jan. 1994.

[4] Werner Goebl and Caroline Palmer, "Synchronization of timing and motion among performing musicians," *Music Perception*, vol. 26, pp. 427–438, May 2009.

[5] G. Weinberg and S. Driscoll, "Robot-human interaction with an anthropomorphic percussionist," in *SIGCHI*, pp. 1229–1232, 2006.

[6] K. et al. Murata, "A beat-tracking robot for human-robot interaction and its evaluation," in *Humanoids*, pp. 79–84, 2008.

[7] G. Weinberg, A. Raman, and T. Mallikarjuna, "Interactive jamming with Shimon: a social robotic musician," in *HRI*, pp. 233–234, 2009.

[8] K. Petersen, J. Solis, and A. Takanishi, "Development of a real-time instrument tracking system for enabling the musical interaction with the Waseda Flutist Robot," in *IROS*, pp. 313–318, 2008.

[9] D. Overholt, J. Thompson, L. Putnam, B. Bell, and J, "A multimodal system for gesture recognition in interactive music performance," *Computer Music*, vol. 33, no. 4, pp. 69–82, 2009.

[10] M. Wanderley, B. Vines, N. Middleton, C. McKay, and W. Hatch, "The musical significance of clarinetists' ancillary gestures: an exploration of the field," *Journal of New Music Research*, vol. 34, no. 1, pp. 97–113, 2005.

[11] R. C. Bolles and M. A. Fischler, "A RANSAC-based approach to model fitting and its application to finding cylinders in range data," in *IJCAI*, pp. 637–643, 1981.

[12] C. Duxbury, J. P. Bello, M. Davies, and M. Sandler, "A combined phase and amplitude based approach to onset detection for audio segmentation," in *WIAMIS*, pp. 275–280, 2003.

[13] P. M. Brossier, *Automatic Annotation of Musical Audio for Interactive Applications*. PhD thesis, Queen Mary University of London, 2006.

[14] D. Mills, "Network Time Protocol (Version 3) specification, implementation and analysis," 1992.