Journal of New Music Research
2001, Vol. 30, No. 2, pp. 159–171

0929-8215/01/3002-159$16.00

An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds

Masataka Goto

National Institute of Advanced Industrial Science and Technology, Tsukuba, Ibaraki, Japan

Abstract

This paper describes a real-time beat tracking system that recognizes a hierarchical beat structure comprising the quarter-note, half-note, and measure levels in real-world audio signals sampled from popular-music compact discs. Most previous beat-tracking systems dealt with MIDI signals and had difficulty in processing, in real time, audio signals containing sounds of various instruments and in tracking beats above the quarter-note level. The system described here can process music with drums and music without drums and can recognize the hierarchical beat structure by using three kinds of musical knowledge: of onset times, of chord changes, and of drum patterns. This paper also describes several applications of beat tracking, such as beat-driven real-time computer graphics and lighting control.

1 Introduction

The goal of this study is to build a real-time system that can track musical beats in real-world audio signals, such as those sampled from compact discs. I think that building such a system that even in its preliminary implementation can work in real-world environments is an important initial step in the computational modeling of music understanding. This is because, as known from the scaling-up problem (Kitano, 1993) in the domain of artificial intelligence, it is hard to scale-up a system whose preliminary implementation works only in laboratory (toy-world) environments. This real-world oriented approach also facilitates the implementation of various practical applications in which music synchronization is necessary.

Most previous beat-tracking related systems had difficulty working in real-world acoustic environments. Most of them (Dannenberg & Mont-Reynaud, 1987; Desain & Honing, 1989, 1994; Allen & Dannenberg, 1990; Driesse, 1991; Rosenthal, 1992a, 1992b; Rowe, 1993; Large, 1995) used as their input MIDI-like representations, and their applications are limited because it is not easy to obtain complete MIDI representations from real-world audio signals. Some systems (Schloss, 1985; Katayose, Kato, Imai, & Inokuchi, 1989; Vercoe, 1994; Todd, 1994; Todd & Brown, 1996; Scheirer, 1998) dealt with audio signals, but they either did not consider the higher-level beat structure above the quarter-note level or did not process popular music sampled from compact discs in real time. Although I developed two beat-tracking systems for real-world audio signals, one for music with drums (Goto & Muraoka, 1994, 1995, 1998) and the other for music without drums (Goto & Muraoka, 1996, 1999), they were separate systems and the former was not able to recognize the measure level.

This paper describes a beat-tracking system that can deal with the audio signals of popular-music compact discs in real time regardless of whether or not those signals contain drum sounds. The system can recognize the *hierarchical beat structure* comprising the *quarter-note level* (almost regularly spaced beat times), the *half-note level*, and the *measure level* (bar-lines).[1] This structure is shown in Figure 1. It assumes that the time-signature of an input song is 4/4 and that the tempo is roughly constant and is either between 61 M.M.[2] and 185 M.M. (for music with drums) or between 61 M.M. and 120 M.M. (for music without drums). These assumptions fit a large class of popular music.

[1] Although this system does not rely on score representation, for convenience this paper uses score-representing terminology like that used by Rosenthal (1992a, 1992b). In this formulation the quarter-note level indicates the temporal basic unit that a human feels in music and that usually corresponds to a quarter note is scores.
[2] Mälzel's Metronome: the number of quarter notes per minute.

Accepted: 9 May, 2001

Correspondence: Dr. Masataka Goto, Information Technology Research Institute, National Institute of Advanced Industrial Science and Technology, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, JAPAN. Tel.: +81-298-61-5898, Fax: +81-298-61-3313, E-mail: m.goto@aist.go.jp

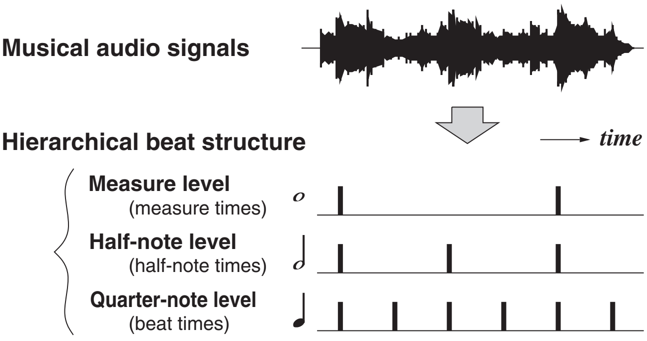


Fig. 1. Beat-tracking problem.

The main issues in recognizing the beat structure in real-world musical acoustic signals are (1) detecting beat-tracking cues in audio signals, (2) interpreting the cues to infer the beat structure, and (3) dealing with the ambiguity of interpretation. First, it is necessary to develop methods for detecting effective musical cues in audio signals. Although various cues – such as onset times, notes, melodies, chords, and repetitive note patterns – were used in previous score-based or MIDI-based systems (Dannenberg & Mont-Reynaud, 1987; Desain & Honing, 1989, 1994; Allen & Dannenberg, 1990; Driesse, 1991; Rosenthal, 1992a, 1992b; Rowe, 1993; Large, 1995), most of those cues are hard to detect in complex audio signals. Second, higher-level processing using musical knowledge is indispensable for inferring each level of the hierarchical beat structure from the detected cues. It is not easy, however, to make musical decisions about audio signals, and the previous audio-based systems (Schloss, 1985; Katayose et al., 1989; Vercoe, 1994; Todd, 1994; Todd & Brown, 1996; Scheirer, 1998) did not use such musical-knowledge-based processing for inferring the hierarchical beat structure. Although some of the above-mentioned MIDI-based systems used musical knowledge, the processing they used cannot be used in audio-based systems because the available cues are different. Third, it must be taken into account that multiple interpretations of beats are possible at any given time. Because there is not necessarily a single specific sound that directly indicates the beat position, there are various ambiguous situations. Two examples are those in which several detected cues may correspond to a beat and those in which different *inter-beat intervals* (the difference between the times of two successive beats) seem plausible.

The following sections introduce a new approach to the beat-tracking problem and describe a beat-tracking model that addresses the issues mentioned above. Experimental results obtained with a system based on that model are then shown, and several of its beat-tracking applications are described.

2 Beat-tracking problem (inverse problem)

In my formulation the beat-tracking problem is defined as a process that organizes musical audio signals into the hierarchical beat structure. As shown in Figure 2, this problem can be considered the *inverse problem* of the following three forward processes by music performers: the process of indicating or implying the beat structure in musical elements when performing music, the process of producing musical sounds (singing or playing musical instruments), and the process of acoustic transmission of those sounds. Although in the brains of performers music is temporally organized according to its hierarchical beat structure, this structure is not explicitly expressed in music; it is implied in the relations among various musical elements which are not fixed and which are dependent on musical genres or pieces. All the musical elements constituting music are then transformed into audio signals through the processes of musical sound production and acoustic transmission.

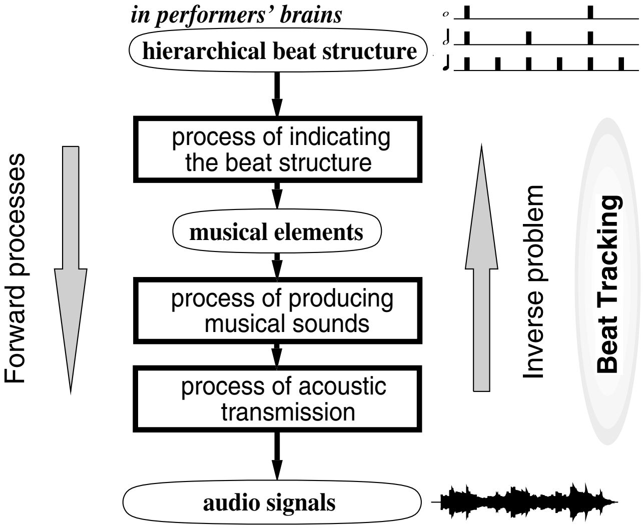

in performers' brains
hierarchical beat structure
process of indicating the beat structure
musical elements
process of producing musical sounds
process of acoustic transmission
audio signals
Forward processes
Inverse problem
Beat Tracking

Fig. 2. Beat tracking as an inverse problem.

The principal reason that beat tracking is intrinsically difficult is that it is the problem of inferring an original beat structure that is not expressed explicitly. The degree of beat-tracking difficulty is therefore not determined simply by the number of musical instruments performing a musical piece; it depends on how explicitly the beat structure is expressed in the piece. For example, it is very easy to track beats in a piece that has only a regular pulse sequence with a constant interval. The main reason that different musical genres and instruments have different tendencies with regard to beat-tracking difficulty is that they have different customary tendencies with regard to the explicitness with which their beat structure is indicated.

In audio-based beat tracking, furthermore, it is also difficult to detect the musical elements that are beat-tracking cues. In that case, the more musical instruments played simultaneously and the more complex the audio signal, the more difficult is the detection of those cues.

3 Beat-tracking model (inverse model)

To solve this inverse problem, I developed a beat-tracking model that consists of two component models: the model of

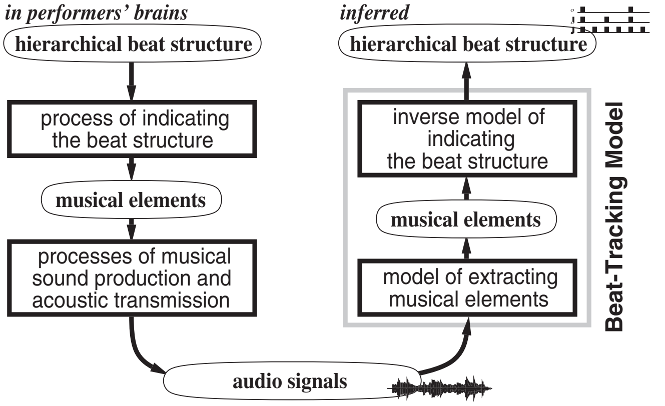

in performers' brains
hierarchical beat structure
process of indicating the beat structure
musical elements
processes of musical sound production and acoustic transmission
inferred
hierarchical beat structure
inverse model of indicating the beat structure
musical elements
model of extracting musical elements
Beat-Tracking Model
audio signals

Fig. 3. Beat-tracking model.

extracting musical elements from audio signals, and the *inverse model* of indicating the beat structure (Fig. 3). The three issues raised in the Introduction are addressed in this beat-tracking model as described in the following three sections.

3.1 Model of extracting musical elements: detecting beat-tracking cues in audio signals

In the model of extracting musical elements, the following three kinds of musical elements are detected as the beat-tracking cues:

1. Onset times
2. Chord changes
3. Drum patterns

As described in Section 3.2, these elements are useful when the hierarchical beat structure is inferred. In this model, onset times are represented by an *onset-time vector* whose dimensions correspond to the onset times of different frequency ranges. A chord change is represented by a *chord-change possibility* that indicates how much the dominant frequency components included in chord tones and their harmonic overtones change in a frequency spectrum. A drum pattern is represented by the temporal pattern of a bass drum and a snare drum.

These elements are extracted from the frequency spectrum calculated with the FFT (1024 samples) of the input (16 bit/22.05 kHz) using the Hanning window. Since the window is shifted by 256 samples, the frequency resolution is consequently 21.53 Hz and the discrete time step (1 *frame-time*[3]) is 11.61 ms. Hereafter $p(t, f)$ is the power of the spectrum of frequency f at time t.

3.1.1 Onset-time vector

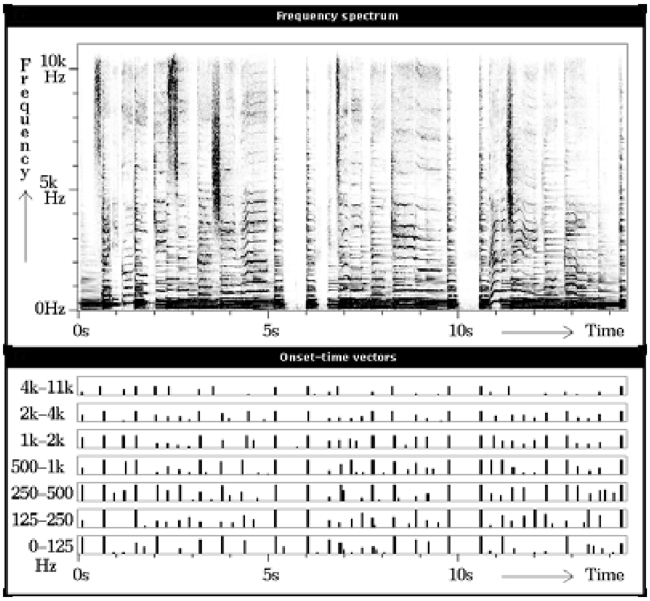


Fig. 4. Examples of a frequency spectrum and an onset-time vector sequence.

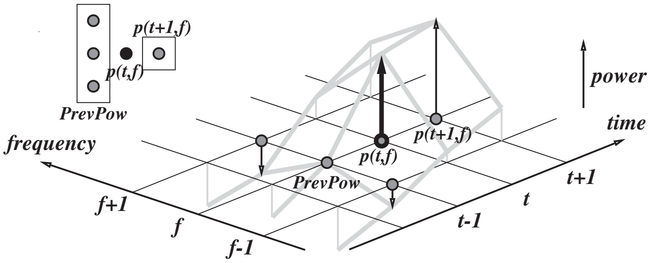


Fig. 5. Extracting an onset component.

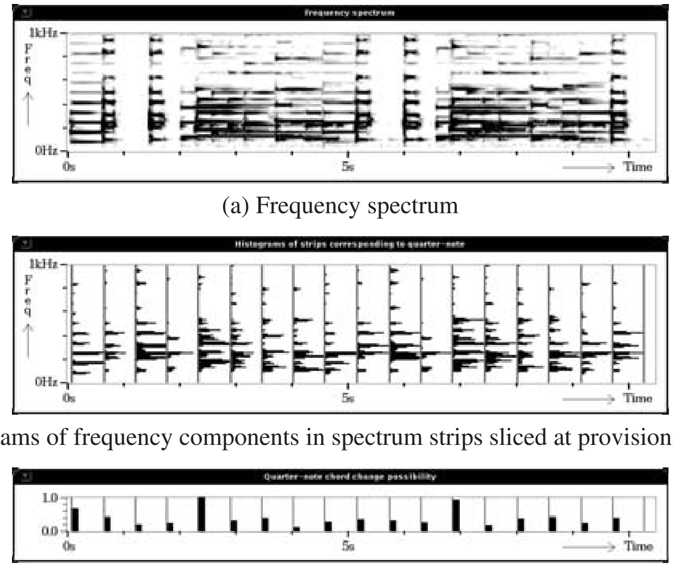
The onset-time vectors are obtained by an *onset-time vectorizer* that transforms the onset times of seven frequency ranges (0–125 Hz, 125–250 Hz, 250–500 Hz, 0.5–1 kHz, 1–2 kHz, 2–4 kHz, and 4–11 kHz) into seven-dimensional onset-time vectors (Fig. 4). This representation makes it possible to consider onset times of all the frequency ranges at the same time. The onset times can be detected by a frequency analysis process that takes into account such factors as the rapidity of an increase in power and the power present in nearby time-frequency regions as shown in Figure 5 (Goto & Muraoka, 1999). Each onset time is given by the peak time found by peak-picking[4] in a degree-of-onset function $D(t) = \Sigma_f d(t, f)$ where

$$d(t,f)=\begin{cases}\max(p(t,f),p(t+1,f))-PrevPow \\ \qquad (\min(p(t,f),p(t+1,f))>PrevPow), \\ 0 \quad (\text{otherwise}),\end{cases} \qquad (1)$$

$$PrevPow = \max(p(t-1,f), p(t-1,f\pm 1)). \qquad (2)$$

[3] The frame-time is the unit of time used in this system, and the term *time* in this paper is the time measured in units of the frame-time.

[4] $D(t)$ is linearly smoothed with a convolution kernel before its peak time is calculated.



(a) Frequency spectrum

(b) Histograms of frequency components in spectrum strips sliced at provisional beat times

(c) Quarter-note chord-change possibilities

Fig. 6. Example of obtaining a chord-change possibility on the basis of provisional beat times.

Because *PrevPow* considers $p(t-1, f \pm 1)$, a false non-onset power increase from $p(t-1, f)$ to $p(t, f)$ is not picked up even if there is a rising frequency component holding high power on both $p(t-1, f-1)$ and $p(t, f)$. The onset times in the different frequency ranges are found by limiting the frequency range of Σ_f.

3.1.2 Chord-change possibility

Because it is difficult to detect chord changes when using only a bottom-up frequency analysis, I developed a method for detecting them by making use of top-down information, *provisional beat times* (Goto & Muraoka, 1996, 1999). The provisional beat times are a hypothesis of the quarter-note level and are inferred from the onset times as described in Section 3.2.1. Possibilities of chord changes in a frequency spectrum are examined without identifying musical notes or chords by name. The idea for this method came from the observation that a listener who cannot identify chord names can nevertheless perceive chord changes. When all frequency components included in chord tones and their harmonic overtones are considered, they are found to tend to change significantly when a chord is changed and to be relatively stable when a chord is not changed. Although it is generally difficult to extract all frequency components from audio signals correctly, the frequency components dominant during a certain period of time can be roughly identified by using a histogram of frequency components. The frequency spectrum is therefore sliced into strips at the provisional beat times and the dominant frequencies of each strip are estimated by using a histogram of frequency components in the strip (Fig. 6). Chord-change possibilities are then obtained by comparing dominant frequencies between adjacent strips. Because the method takes advantage of not requiring musical notes to be identified, it can detect chord changes in real-world audio signals, where chord identification is generally difficult.

For different purposes, the model uses two kinds of possibilities of chord changes, one at the quarter-note level and the other at the eighth-note level, by slicing the frequency spectrum into strips at the provisional beat times and by slicing it at the interpolatd eighth-note times. The one obtained by slicing at the provisional beat times is called the *quarter-note chord-change possibility* and the one obtained by slicing at the eighth-note times is called the *eighth-note chord-change possibility*. They respectively represent how likely a chord is, under the current beat-position hypothesis, to change on each quarter-note position and on each eighth-note position. The detailed equations used in this method are described in a paper focusing on beat tracking for music without drum-sounds (Goto & Muraoka, 1999).

3.1.3 Drum pattern

A drum-sound finder detects the onset time of a bass drum (BD) by using onset components and the onset time of a snare drum (SD) by using noise components. Those onset times are then formed into the drum patterns by making use of the provisional beat times (top-down information) (Fig. 7).

[Detecting BD onset times]

Because the sound of a BD is not known in advance, the drum-sound finder learns the characteristic frequency of a BD by examining the extracted onset components $d(t, f)$

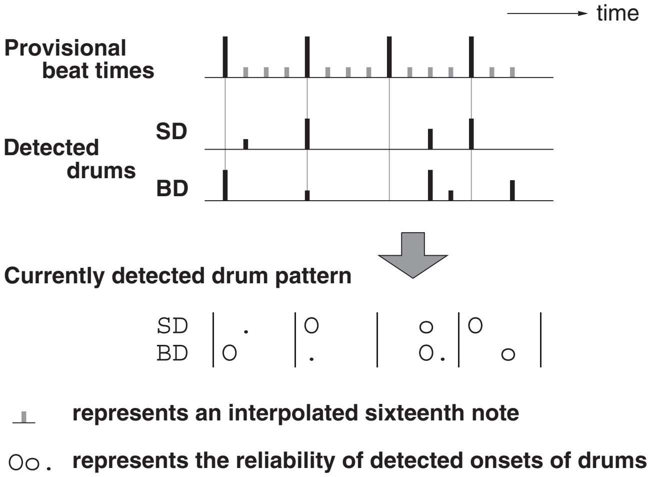

time
Provisional beat times
Detected drums
SD
BD
Currently detected drum pattern
SD | . |O | o |O |
BD |O |. | O.| o |
represents an interpolated sixteenth note
Oo. represents the reliability of detected onsets of drums

Fig. 7. Forming a drum pattern by making use of provisional beat times.

(Equation (1)). For times at which onset components are found, the finder picks peaks along the frequency axis and makes a histogram of them (Fig. 8). The finder then judges that a BD has sounded at times when an onset's peak frequency coincides with the characteristic frequency that is given by the lowest-frequency peak of the histogram.

[Detecting SD onset times]

Since the sound of a SD typically has noise components widely distributed along the frequency axis, the finder needs to detect such components. First, the noise components $n(t, f)$ are given by the following equations:

$$n(t, f) = \begin{cases} p(t, f) & \left(\min(HightFreqAve, LowFreqAve) \right. \\ & \left. > \frac{1}{2}\, p(t, f)\right), \\ 0 & \text{(otherwise)}, \end{cases} \tag{3}$$

$$HighFreqAve = \frac{1}{4}\left(p(t, f+2) + \sum_{i=-1}^{1} p(t+i, f+1)\right), \tag{4}$$

$$LowFreqAve = \frac{1}{4}\left(p(t, f-2) + \sum_{i=-1}^{1} p(t+i, f-1)\right), \tag{5}$$

where *HighFreqAve* and *LowFreqAve* respectively represent the local averages of the power in higher and lower regions of $p(t, f)$. When the surrounding *High Freq Ave* and *Low Freq Ave* are both larger than half of $p(t, f)$, the component $p(t, f)$ is not considered a peaked component but a noise component distributed almost uniformly. As shown in Figure 8, the noise components $n(t, f)$ are quantized: the frequency axis of the noise components is divided into subbands (the number of subbands is 16) and the mean of $n(t, f)$ in each subband is calculated. The finder then calculates $c(t)$, which represents how widely noise components are distributed along the frequency axis: $c(t)$ is calculated as the product of all quantized components within the middle frequency range (from 1.4 kHz to 7.5 kHz). Finally, the SD onset time is obtained by peak-picking of $c(t)$ in the same way as in the onset-time finder.

3.2 Inverse model of indicating the beat structure: interpreting beat-tracking cues to infer the hierarchical beat structure

Each level of the beat structure is inferred by using the inverse model of indicating the beat structure. The inverse model is represented by the following three kinds of musical knowledge (heuristics) corresponding to the three kinds of musical elements.

3.2.1 Musical knowledge of onset times

To infer the quarter-note level (i.e., to determine the provisional beat times), the model uses the following heuristic knowledge:

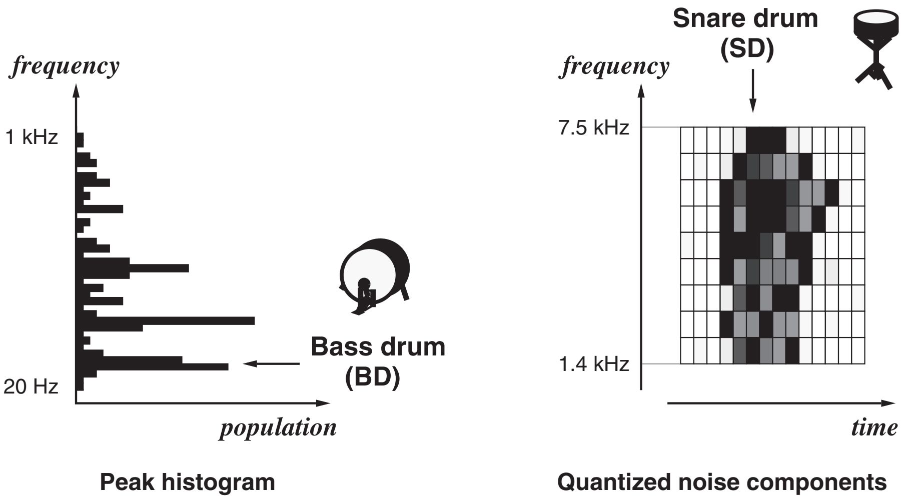


Fig. 8. Detecting a bass drum (BD) and a snare drum (SD).

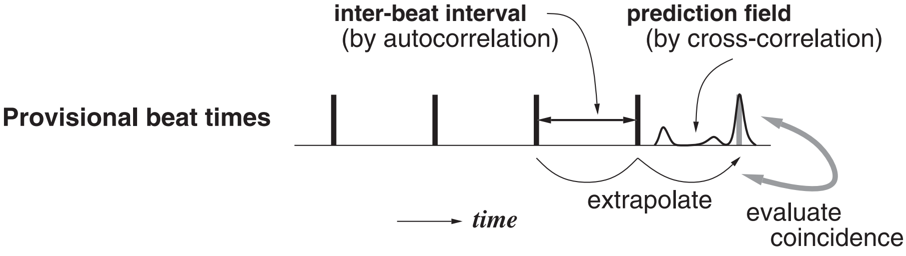


Fig. 9. Predicting the next beat.

(a-1) "A frequent inter-onset interval is likely to be the inter-beat interval."
(a-2) "Onset times tend to coincide with beat times (i.e., sounds are likely to occur on beats)."

The reason the term *the provisional beat times* is used is that the sequence of beat times obtained below is just a single hypothesis of the quarter-note level: multiple hypotheses are considered as explained in Section 3.3.

By using autocorrelation and cross-correlation of the onset-time vectors, the model determines the inter-beat interval and predicts the next beat time. The inter-beat interval is determined by calculating the windowed and normalized vectorial autocorrelation function $Ac(\tau)$ of the onset-time vectors:[5]

$$Ac(\tau) = \frac{\sum_{t=c-AcPeriod}^{c} \mathrm{win}(c-t, AcPeriod)(\vec{o}(t)\cdot\vec{o}(t-\tau))}{\sum_{t=c-AcPeriod}^{c} \mathrm{win}(c-t, AcPeriod)(\vec{o}(t)\cdot\vec{o}(t))}, \tag{6}$$

where $\vec{o}(t)$ is the onset-time vector at time t, c is the current time, and *AcPeriod* is the autocorrelation period. The window function $\mathrm{win}(t, s)$ whose window size is s is

$$\mathrm{win}(t, s) = \begin{cases} 1.0 - 0.5\dfrac{t}{s} & 0 \le t \le s, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

According to the knowledge **(a-1)**, the inter-beat interval is given by the τ with the maximum height in $Ac(\tau)$ within an appropriate inter-beat interval range. To predict the next beat time by using the knowledge **(a-2)**, the model forms a prediction field (Fig. 9) by calculating the windowed cross-correlation function $Cc(\tau)$ between the sum $O(t)$ of all dimensions of $\vec{o}(t)$ and a tentative beat-time sequence $T_{tmp}(t, m)$ whose interval is the inter-beat interval obtained using Equation (6):

$$Cc(\tau) = \sum_{t=c-CcPeriod}^{c} \left(\mathrm{win}(c-t, CcPeriod)O(t) \sum_{m=1}^{\mathrm{CcNumBeats}} \delta(t - T_{tmp}(c+\tau, m)) \right), \tag{8}$$

$$T_{tmp}(t, m) = \begin{cases} t - I(t) & (m = 1), \\ T_{tmp}(t, m-1) - I(T_{tmp}(t, m-1)) & (m > 1), \end{cases} \tag{9}$$

$$\delta(x) = \begin{cases} 1 & (x = 0), \\ 0 & (x \neq 0), \end{cases} \tag{10}$$

where $I(t)$ is the inter-beat interval at time t, *CcPeriod* (= CcNumBeats $I(c)$) is the window size for calculating the cross-correlation, and CcNumBeats (= 12) is a constant factor that determines how many previous beats are considered in calculating the cross-correlation. The prediction field is thus given by $Cc(\tau)$ where $0 \le \tau \le I(c) - 1$. Finally, the local-maximum peak in the prediction field is selected as the next beat time while considering to pursue the peak close to the sum of the previously selected one and the inter-beat interval.

The reliability of each hypothesis of the provisional beat times is then evaluated according to how closely the next beat time predicted from the onset times coincides with the time extrapolated from the past beat times (Fig. 9).

3.2.2 Musical knowledge of chord changes

To infer each level of the structure, the model uses the following knowledge:

(b-1) "Chords are more likely to change on beat times than on other positions."
(b-2) "Chords are more likely to change on half-note times than on other positions of beat times."
(b-3) "Chords are more likely to change at the beginnings of measures than at other positions of half-note times."

Figure 10 shows a sketch of how the half-note and measure times are inferred from the chord-change possibilities. According to the knowledge **(b-2)**, if the quarter-note chord-change possibility is high enough, its time is considered to indicate the position of the half-note times. According to the knowledge **(b-3)**, if the quarter-note chord-change possibility of a half-note time is higher than that of adjacent half-note times, its time is considered to indicate the position of the measure times (bar-lines).

The knowledge **(b-1)** is used for reevaluating the reliability of the current hypothesis: if the eighth-note chord-change possibility tends to be higher on beat times than on eighth-note displacement positions, the reliability is increased.

3.2.3 Musical knowledge of drum patterns

For music with drum-sounds, eight prestored drum patterns, like those illustrated in Figure 11, are prepared. They repre-

[5] Vercoe (1994) also proposed the use of a variant of autocorrelation for rhythmic analysis.

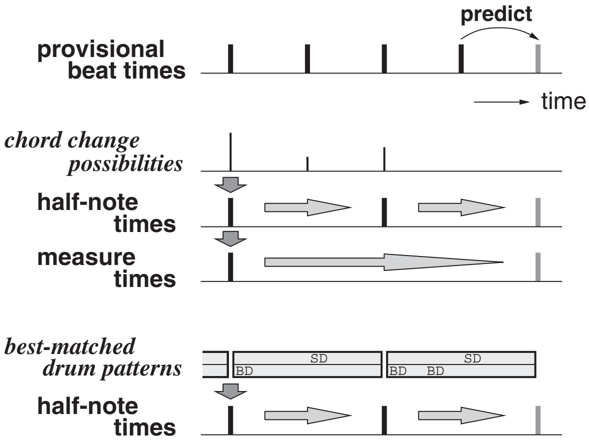


Fig. 10. Knowledge-based inferring.

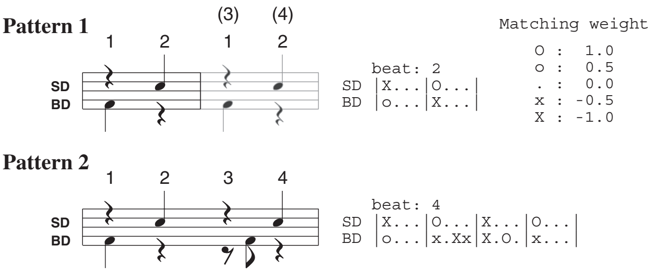


Fig. 11. Examples of prestored drum patterns.

sent the ways drum-sounds are typically used in a lot of popular music. The beginning of a pattern should be a half-note time, and the length of the pattern is restricted to a half note or a measure. In the case of a half note, patterns repeated twice are considered to form a measure.

When an input drum pattern that is currently detected in the audio signal matches one of the prestored drum patterns well, the model uses the following knowledge to infer the quarter-note and half-note levels:

(c-1) "The beginning of the input drum pattern indicates a half-note time."
(c-2) "The input drum pattern has the appropriate inter-beat interval."

Figure 10 also shows a sketch of how the half-note times are inferred from the best-matched drum pattern: according to the knowledge **(c-1)**, the beginning of the best-matched pattern is considered to indicate the position of a half-note time. Note that the measure level cannot be determined this way: the measure level is determined by using the quarter-note chord-change possibilities as described in Section 3.2.2.

The knowledge **(c-2)** is used for reevaluating the reliability of the current hypothesis: the reliability is increased according to how well an input drum pattern matches one of the prestored drum patterns.

3.2.4 Musical knowledge selection based on the presence of drum-sounds

To infer the quarter-note and half-note levels, the musical knowledge of chord changes (**(b-1)** and **(b-2)**) and the musical knowledge of drum patterns (**(c-1)** and **(c-2)**) should be selectively applied according to the presence or absence of drum-sounds as shown in Table 1. I therefore developed a method for judging whether or not the input audio signal contains drum-sounds. This judgement could not be made simply by using the detected results because the detection of the drum-sounds is noisy. According to the fact that in popular music a snare drum is typically played on the second and fourth quarter notes in a measure, the method judges that the input audio signal contains drum-sounds only when autocorrelation of the snare drum's onset times is high enough.

3.3 Dealing with ambiguity of interpretation

To enable ambiguous situations to be handled when the beat-tracking cues are interpreted, a multiple-agent model in which multiple agents examine various hypotheses of the beat structure in parallel as illustrated in Figure 12 was developed (Goto & Muraoka, 1996, 1999). Each agent uses its own strategy and makes a hypothesis by using the inverse model described in Section 3.2. An agent interacts with another agent to track beats cooperatively and adapts to the current situation by adjusting its strategy. It then evaluates the reliability of its own hypothesis according to

Table 1. Musical knowledge selection for music with drum-sounds and music without drum-sounds.

Beat structure	Without drums	With drums
Measure level	quarter-note chord-change possibility (knowledge (**b-3**))	quarter-note chord-change possibility (knowledge (**b-3**))
Half-note level	quarter-note chord-change possibility (knowledge (**b-2**))	drum pattern (knowledge (**c-1**))
Quarter-note level	eighth-note chord-change possibility (knowledge (**b-1**))	drum pattern (knowledge (**c-2**))

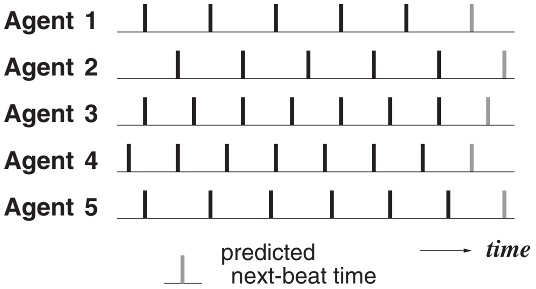


Fig. 12. Multiple hypotheses maintained by multiple agents.

how well the inverse model can be applied. The final beat-tracking result is determined on the basis of the most reliable hypothesis.

3.4 System overview

Figure 13 shows an overview of the system based on the beat-tracking model. In the frequency-analysis stage, the system detects the onset-time vectors (Section 3.1.1), detects onset times of bass drum and snare drum sounds (Section 3.1.3), and judges the presence or absence of drum-sounds (Section 3.2.4). In the beat-prediction stage, each agent infers the quarter-note level by using the autocorrelation and cross-correlation of the onset-time vectors (Section 3.2.1). Each higher-level checker corresponding to each agent then detects chord changes (Section 3.1.2) and drum patterns (Section 3.1.3) by using the quarter-note level as the top-down information. Using those detected results, each agent infers the higher levels (Section 3.2.2 and Section 3.2.3) and evaluates the reliability of its hypothesis. The agent manager gathers all hypotheses and then determines the final output on the basis of the most reliable one. Finally, the beat-tracking result is transmitted to other application programs via a computer network.

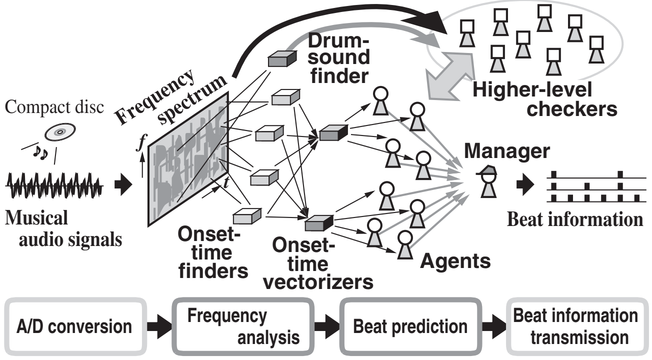


Fig. 13. Overview of the beat-tracking system.

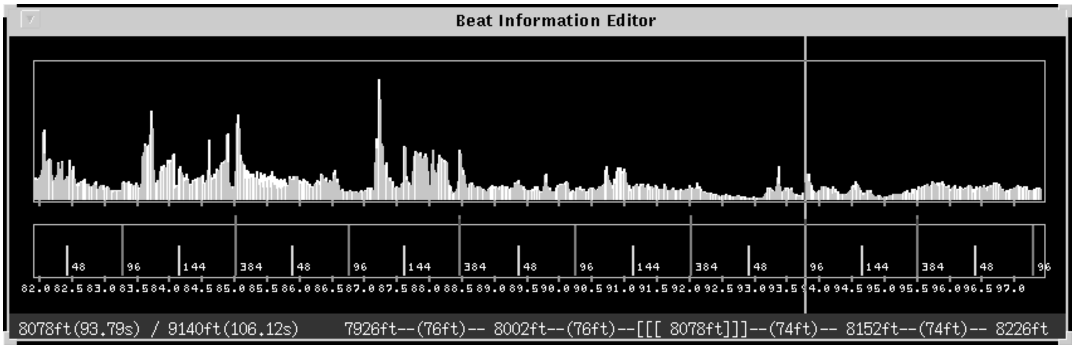
4 Experiments and results

The system was tested on monaural audio signals sampled from commercial compact discs of popular music. Eighty-five songs, each at least one minute long, were used as the inputs. Forty-five of the songs had drum-sounds (32 artists, tempo range: 67–185 M.M.) and forty did not (28 artists, tempo range: 62–116 M.M.). Each song had the 4/4 time-signature and a roughly constant tempo.

In this experiment the system output was compared with the hand-labeled hierarchical beat structure. To label the correct beat structure, I developed a beat-structure editor program that enables a user to mark the beat positions in a digitized audio signal while listening to the audio and watching its waveform (Fig. 14). The positions can be finely adjusted by playing back the audio with click tones at beat times, and the half-note and measure levels can also be labeled. The recognition rates were evaluated by using the quantitative evaluation measures for analyzing the beat-tracking accuracy that were proposed in earlier papers (Goto & Muraoka, 1997, 1999). Unstably tracked songs (those for which correct beats were obtained just temporarily) were not considered to be tracked correctly.

4.1 Results of evaluating recognition rates

The results of evaluating the recognition rates are listed in Table 2. I also evaluated how quickly the system started to track the correct beats stably at each level of the hierarchical beat structure, and the start time of tracking the correct beat



```
8078ft(93.79s) / 9140ft(106.12s)     7926ft--(76ft)-- 8002ft--(76ft)--[[[ 8078ft]]]--(74ft)-- 8152ft--(74ft)-- 8226ft
```

Fig. 14. Beat-structure editor program.

Table 2. Results of evaluating recognition rates at each level of the beat structure.

Beat structure	Without drums	With drums
Measure level	32 of 34 songs (94.1%)	34 of 39 songs (87.2%)
Half-note level	34 of 35 songs (97.1%)	39 of 39 songs (100%)
Quarter-note level	35 of 40 songs (87.5%)	39 of 45 songs (86.7%)

Table 3. Start time of tracking the correct beat structure (music without drums).

Beat structure	mean	min	max
Measure level	18.47 s	3.42 s	42.56 s
Half-note level	13.74 s	3.42 s	36.75 s
Quarter-note level	10.99 s	0.79 s	36.75 s

Table 4. Start time of tracking the correct beat structure (music with drums).

Beat structure	mean	min	max
Measure level	22.00 s	6.32 s	40.05 s
Half-note level	17.15 s	4.20 s	41.89 s
Quarter-note level	13.87 s	0.52 s	41.89 s

structure is shown in Figure 15. The horizontal axis represents the song numbers (#) arranged in order of the start time of the quarter-note level up to song #32 (for music without drums) and #34 (for music with drums). The mean, minimum, and maximum of the start time of all the correctly tracked songs are listed in Table 3 and Table 4. These results show that in each song where the beat structure was eventually determined correctly, the system initially had trouble determining a higher rhythmic level even though a lower level was correct.

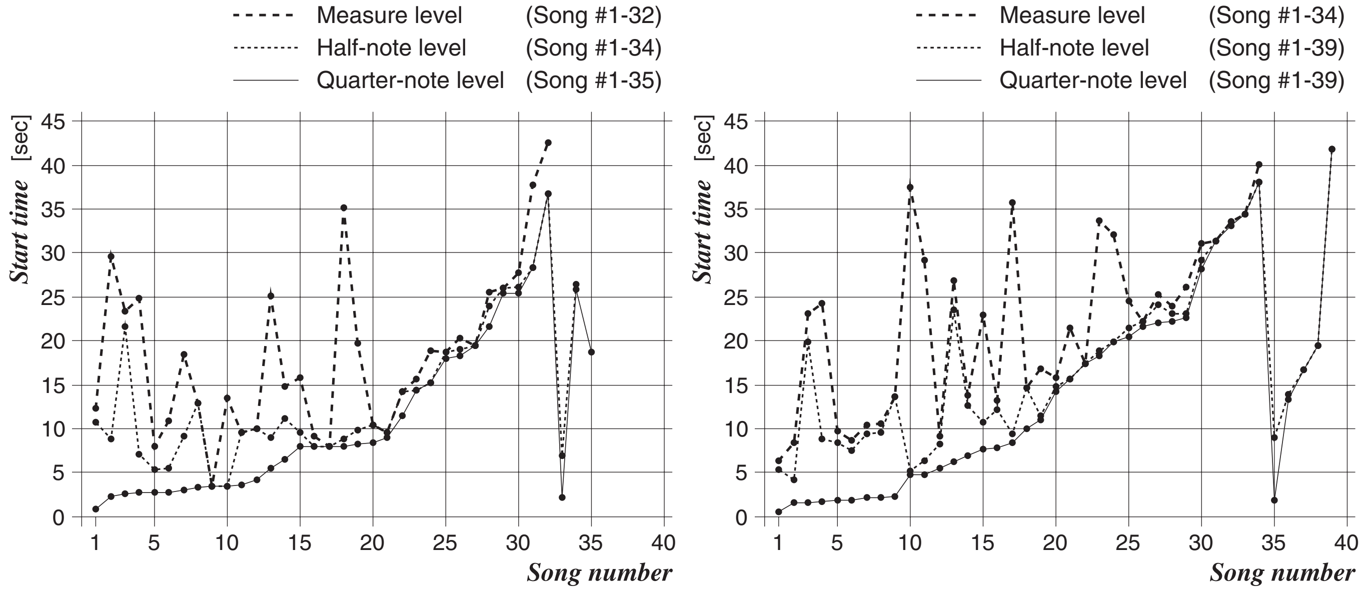
The following are the results of analyzing the reasons the system made mistakes:

[Music without drums]

The quarter-note level was not determined correctly in five songs. In one of them the system tracked eighth-note displacement positions because there were too many syncopations in the basic accompaniment rhythm. In three of the other songs, although the system tracked correct beats temporarily (during from 14 to 24 s), it sometimes got out of position because the onset times were very few and irregular. In the other song the tracked beat times deviated too much during a measure, although the quarter-note level was determined correctly during most of the song.

In a song where the half-note level was wrong, the system failed to apply the musical knowledge of chord changes because chords were often changed at the fourth quarter note in a measure.

In two songs where only the measure level was mistaken, chords were often changed at every other quarter-note and the system was not able to determine the beginnings of measures.



(music without drums)

(music with drums)

Fig. 15. Start time of tracking the correct beat structure.

[Music with drums]

The quarter-note level was not determined correctly in six songs. In two of them the system correctly tracked beats in the first half of the song, but the inter-beat interval became 0.75 or 1.5 times of the correct one in the middle of the song. In two of the other songs the quarter-note level was determined correctly except that the start times were too late: 45.3 s and 51.8 s (the start time had to be less than 45 s for the tracking to be considered correct). In the other two songs the tracked beat times deviated too much temporarily, although the system tracked beat times correctly during most of the song.

The system made mistakes at the measure level in five songs. In one of them the system was not able to determine the beginnings of measures because chords were often changed at every quarter-note or every other quarter-note. In two of the other songs the quarter-note chord-change possibilities were not obtained appropriately because the frequency components corresponding to the chords were too weak. In the other two songs the system determined the measure level correctly except that the start times were too late: 48.3 s and 49.9 s.

The results mentioned above show that the recognition rates at each level of the beat structure were more than 86.7 percent and that the system is robust enough to deal in real time with real-world audio signals containing sounds of various instruments.

4.2 Results of measuring rhythmic difficulty

It is important to measure the degree of beat-tracking difficulty for the songs that were used in testing the beat-tracking system. As discussed in Section 2, the degree of beat-tracking difficulty depends on how explicitly the beat structure is expressed. It is very difficult, however, to measure its explicitness because it is influenced from various aspects of the songs. In fact, most previous beat-tracking studies have not dealt with this issue. I therefore tried, as a first step, to evaluate the power transition of the input audio signals. In terms of the power transition, it is more difficult to track beats of a song in which the power tends to be lower on beats than between adjacent beats. In other words, the larger the number of syncopations, the greater the difficulty of tracking beats.

I thus proposed a quantitative measure of the rhythmic difficulty, called the *power-difference measure*,[6] that considers differences between the power on beats and the power on other positions. This measure is defined as the mean of all the normalized power difference $diff_{pow}(n)$ in the song:

$$diff_{pow}(n) = 0.5\frac{pow_{other}(n) - pow_{beat}(n)}{\max(pow_{other}(n),\ pow_{beat}(n))} + 0.5, \quad (11)$$

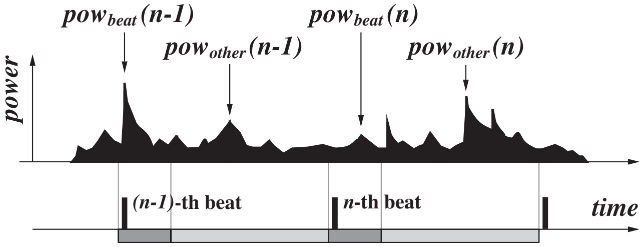


Fig. 16. Finding the local maximum of the power.

where $pow_{beat}(n)$ represents the local maximum power on the n-th beat[7] and $pow_{other}(n)$ represents the local maximum power on positions between the n-th beat and $(n + 1)$-th beat (Fig. 16). The power-difference measure takes a value between 0 (easiest) and 1 (most difficult). For a regular pulse sequence with a constant interval, for example, this measure takes a value of 0.

Using this power-difference measure, I evaluated the rhythmic difficulty of each of the songs used in testing the system. Figure 17 shows two histograms of the measure, one for songs without drum-sounds and the other for songs with drum-sounds. Comparison between these two histograms indicates that the power-difference measure tends to be higher for songs without drum-sounds than with drum-sounds. In particular, it is interesting that the measure exceeded 0.5 in more than half of the songs without drum-sounds; this indicates that the power on beats is often lower than the power on other positions in those songs. This also suggests that a simple idea of tracking beats by regarding large power peaks of the input audio signal as beat positions is not feasible.

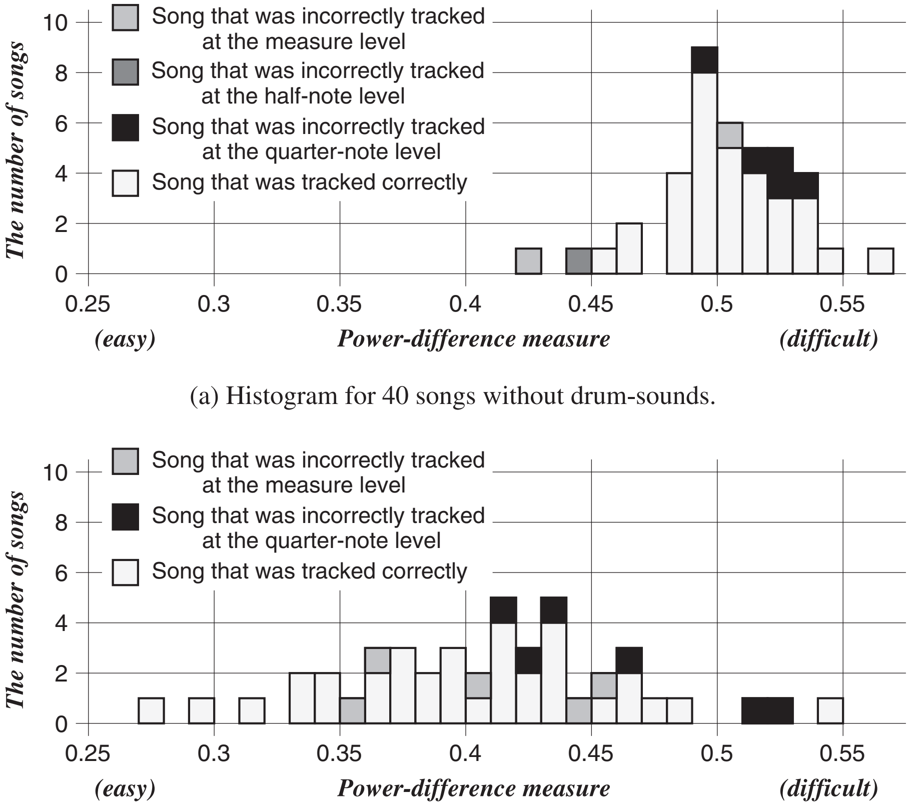
Figure 17 also indicates the songs that were incorrectly tracked at each level of the beat structure. While the power-difference measure tends to be higher for the songs that were incorrectly tracked at the quarter-note level, it's value is not clearly related to the songs that were incorrectly tracked at the half-note and measure levels: the influence from various other aspects besides the power transition is dominant in inferring the half-note and measure levels. Although this measure is not perfect for evaluating the rhythmic difficulty and other aspects should be taken into consideration, it should be a meaningful step on the road to measuring the beat-tracking difficulty in an objective way.

5 Applications

Since beat tracking can be used to automate the time-consuming tasks that must be completed in order to synchronize events with music, it is useful in various applications, such as video editing, audio edition, and human-computer improvisation. The development of applications

[6] The detailed equations of the power-difference measure are described by Goto and Muraoka (1999).

[7] The hand-labeled correct quarter-note level is used for this evaluation.



(a) Histogram for 40 songs without drum-sounds.

(b) Histogram for 45 songs with drum-sounds.

Fig. 17. Evaluating beat-tracking difficulty: histograms of the evaluated power-difference measure.

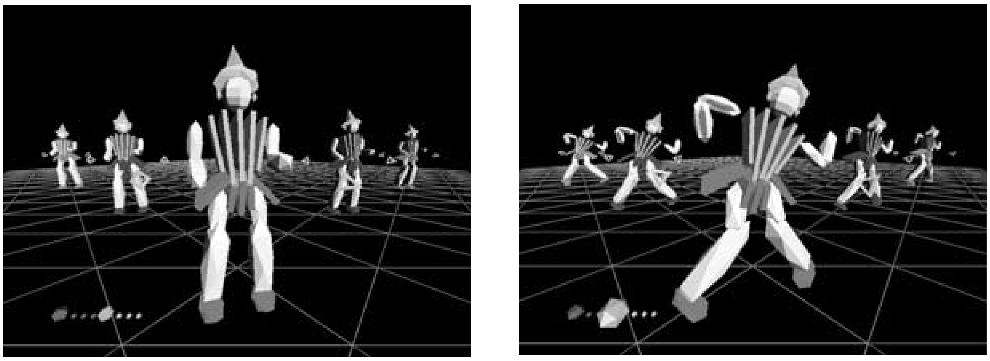


Fig. 18. Virtual dancer "Cindy".

is facilitated by using a network protocol called *RMCP* (*Remote Music Control Protocol*) (Goto, Neyama, & Muraoka, 1997) for sharing the beat-tracking result among several distributed processes. RMCP is designed to share symbolized musical information through networks and it supports time-scheduling using time stamps and broadcast-based information sharing without the overhead of multiple transmission.

- Beat-driven real-time computer graphics
  The beat-tracking system makes it easy to create real-time computer graphics synchronized with music and has been used to develop a system that displays virtual dancers and several graphic objects whose motions and positions change in time to beats (Fig. 18). This system has several dance sequences, each for a different mood of dance motions. While a user selects a dance sequence manually, the timing of each motion in the selected sequence is determined automatically on the basis of the beat-tracking results. Such a computer graphics system is suitable for live stage, TV program, and *Karaoke* uses.
- Stage-lighting control
  Beat tracking facilitates the automatic synchronization of computer-controlled stage lighting with the beats in a

musical performance. Various properties of lighting – such as color, brightness, and direction – can be changed in time to music. At the moment this application is simulated on a computer graphics display with virtual dancers.

- Intelligent drum machine
  A preliminary system that can play drum patterns in time to input musical audio signals without drum-sounds has been implemented. This application is potentially useful for automatic MIDI-audio synchronization and intelligent computer accompaniment.

The beat-structure editor program mentioned in Section 4 is also useful in practical applications. A user can correct or adjust the output beat structure when the system output includes mistakes and can make the whole hierarchical beat structure for a certain application from scratch.

6 Conclusion

This paper has described the beat-tracking problem in dealing with real-world audio signals, a beat-tracking model that is a solution to that problem, and applications based on a real-time beat-tracking system. Experimental results show that the system can recognize the hierarchical beat structure comprising the quarter-note, half-note, and measure levels in audio signals of compact disc recordings. The system has also been shown to be effective in practical applications.

The main contributions of this paper are to provide a view in which the beat-tracking problem is regarded as an inverse problem and to provide a new computations model that can recognize, in real time, the hierarchical beat structure in audio signals regardless of whether or not those signals contain drum-sounds. The model uses sophisticated frequency-analysis processes based on top-down information and uses a higher-level processing based on three kinds of musical knowledge that are selectively applied according to the presence or absence of drum-sounds. These features made it possible to overcome difficulties in making musical decisions about complex audio signals and to infer the hierarchical beat structure.

The system will be upgraded by enabling it to follow tempo changes and by generalizing it to other musical genres. Future work will include integration of the beat-tracking model described here and other music-understanding models, such as one detecting melody and bass lines (Goto, 1999, 2000).

Acknowledgments

This paper is based on my doctoral dissertation supervised by Professor Yoichi Muraoka at Waseda University. I thank Professor Yoichi Muraoka for his guidance and support and for providing an ideal research environment full of freedom.

References

Allen, P.E. & Dannenberg, R.B. (1990). Tracking Musical Beats in Real Time. In *Proceedings of the 1990 International Computer Music Conference*, pp. 140–143. Glasgow: ICMA.

Dannenberg, R.B. & Mont-Reynaud, B. (1987). Following an Improvisation in Real Time. In *Proceedings of the 1987 International Computer Music Conference* (pp. 241–248). Champaign/Urbana: ICMA.

Desain, P. & Honing, H. (1989). The Quantization of Musical Time: A Connectionist Approach. *Computer Music Journal*, *13*(3), 56–66.

Desain, P. & Honing, H. (1994). Advanced issues in beat induction modeling: syncopation, tempo and timing. In *Proceedings of the 1994 International Computer Music Conference* (pp. 92–94). Aarhus: ICMA.

Driesse, A. (1991). Real-Time Tempo Tracking Using Rules to Analyze Rhythmic Qualities. In *Proceedings of the 1991 International Computer Music Conference* (pp. 578–581). Montreal: ICMA.

Goto, M. (1999). A Real-time Music Scene Description System: Detecting Melody and Bass Lines in Audio Signals. In *Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis* (pp. 31–40). Stockholm: IJCAII.

Goto, M. (2000). A Robust Predominant-F0 Estimation Method for Real-time Detection of Melody and Bass Lines in CD Recording. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. II–757–760). Stanbul: IEEE.

Goto, M. & Muraoka, Y. (1994). A Beat Tracking System for Acoustic Signals of Music. In *Proceedings of the Second ACM International Conference on Multimedia* (pp. 365–372). San Francisco: ACM.

Goto, M. & Muraoka, Y. (1995). A Real-time Beat Tracking System for Audio Signals. In *Proceedings of the 1995 International Computer Music Conference* (pp. 171–174). Banff: ICMA.

Goto, M. & Muraoka, Y. (1996). Beat Tracking based on Multiple-agent Architecture – A Real-time Beat Tracking System for Audio Signals –. In *Proceedings of the Second International Conference on Multiagent Systems* (pp. 103–110). Kyoto: AAAI Press.

Goto, M. & Muraoka, Y. (1997). Issues in Evaluating Beat Tracking Systems. In *Working Notes of the IJCAI-97 Workshop on Issues in AI and Music* (pp. 9–16). Nagoya: IJCAII.

Goto, M. & Muraoka, Y. (1998). Music Understanding At The Beat Level – Real-time Beat Tracking For Audio Signals. In D.F. Rosenthal & H.G. Okuno (Eds.), *Computational Auditory Scene Analysis* (pp. 157–176). New Jersey: Lawrence Erlbaum Associates, Publishers.

Goto, M. & Muraoka, Y. (1999). Real-time Beat Tracking for Drumless Audio Signals: Chord Change Detection for Musical Decisions. *Speech Communication*, *27*(3–4), 311–335.

Goto, M., Neyama, R., & Muraoka, Y. (1997). RMCP: Remote Music Control Protocol – Design and Applications –. In *Proceedings of the 1997 International Computer Music Conference* (pp. 446–449). Thessaloniki: ICMA.

Katayose, H., Kato, H., Imai, M., & Inokuchi, S. (1989). "An Approach to an Artificial Music Expert," In *Proceedings of the 1989 International Computer Music Conference* (pp. 139–146). Columbus: ICMA.

Kitano, H. (1993). "Challenges of Massive Parallelism," In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* (pp. 813–834). Chambery: IJCAII.

Large, E.W. (1995). Beat Tracking with a Nonlinear Oscillator. In *Working Notes of the IJCAI-95 Workshop on Artificial Intelligence and Music* (pp. 24–31). Montreal: IJCAII.

Rosenthal, D. (1992a). "Emulation of Human Rhythm Perception," *Computer Music Journal*, *16*(1), 64–76.

Rosenthal, D. (1992b). *Machine Rhythm: Computer Emulation of Human Rhythm Perception*. Ph.D. thesis, Massachusetts Institute of Technology.

Rowe, R. (1993). *Interactive Music Systems*. Massachusetts: MIT Press.

Scheirer, E.D. (1998). "Tempo and beat analysis of acoustic musical signals," *Journal of the Acoustical Society America*, *103*(1), 588–601.

Schloss, W.A. (1985). *On The Automatic Transcription of Percussive Music – From Acoustic Signal to High-Level Analysis*. Ph.D. thesis, CCRMA, Stanford University.

Todd, N.P.M. (1994). "The Auditory 'Primal Sketch": A Multiscale Model of Rhythmic Grouping," *Journal of New Music Research*, *23*(1), 25–70.

Todd, N.P.M. & Brown, G.J. (1996). "Visualization of Rhythm, Time and Metre," *Artificial Intelligence Review*, *10*, 253–273.

Vercoe, B. (1994). "Perceptually-based music pattern recognition and response," In *Proceedings of the Third International Conference for the Perception and Cognition of Music* (pp. 59–60). Liège: ESCOM.