ELSEVIER

Speech Communication 27 (1999) 311–335

SPEECH COMMUNICATION

Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions

Masataka Goto [*], Yoichi Muraoka

School of Science and Engineering, Waseda University, 3-4-1 Ohkubo, Shinjuku-ku, Tokyo 169-8555, Japan

Received 29 December 1997; received in revised form 18 September 1998

Abstract

This paper describes a real-time beat-tracking system that detects a hierarchical beat structure in musical audio signals without drum-sounds. Most previous systems have dealt with MIDI signals and had difficulty in applying, in real time, musical heuristics to audio signals containing sounds of various instruments and in tracking beats above the quarter-note level. Our system not only tracks beats at the quarter-note level but also detects beat structure at the half-note and measure levels. To make musical decisions about the audio signals, we propose a method of detecting chord changes that does not require chord names to be identified. The method enables the system to track beats at different rhythmic levels – for example, to find the beginnings of half notes and measures – and to select the best of various hypotheses about beat positions. Experimental results show that the proposed method was effective to detect the beat structure in real-world audio signals sampled from compact discs of popular music. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Beat tracking; Rhythm perception; Chord change detection; Music understanding; Computational auditory scene analysis

1. Introduction

One of the goals of computational auditory scene analysis is to implement a computational model that can understand musical audio signals in a human-like fashion. A popular approach to this goal is to build an automatic music transcription system, or a sound source separation system, which typically transforms audio signals into a symbolic representation such as a musical score or MIDI data. Although such detailed-transcription technologies are important, they have difficulty in dealing with compact disc audio signals in general. Because only a trained listener can identify the names of musical notes and chords, we can infer that musical transcription is a skill difficult even for human beings to acquire.

On the other hand, an untrained listener understands music to some extent without mentally representing audio signals as musical scores. For example, even a listener who cannot identify chord names can perceive harmony and chord changes. A listener who cannot segregate and identify every musical note can nevertheless track musical beats and keep time to music by hand-clapping or foot-tapping. We therefore

[*] Corresponding author. Present address: Machine Understanding Division, Electrotechnical Laboratory, 1-1-4 Ume zono, Tsukuba, Ibaraki 305-8568, Japan. Tel.: +81-298-54-5898; fax: +81-298-54-3313; e-mail: goto@etl.go.jp

think that it is important to first build a computer system that can understand music at the level untrained human listeners do, without relying on transcription, and then extend the system so that it can understand music at the higher level musicians do.

Our approach is to build a real-time beat-tracking system that detects a hierarchical beat structure of three rhythmic levels in real-world audio signals, such as those sampled from popular compact discs. Beat tracking is an important part of the computational modeling of music understanding because the beat is fundamental, for both trained and untrained listeners, to the perception of Western music. The purpose of this study is to build such a beat-tracking system that is practical from the engineering viewpoint, that gives suggestions to the modeling of higher-level music understanding systems, and that is useful in various applications, such as music-synchronized CG animation, video/audio editing, and human-computer improvisation in live ensemble. For this purpose it is desirable to detect a hierarchical beat structure, since a higher structure like the measure (bar-line) level can provide information more useful for modeling music understanding and for implementing beat-tracking applications. We therefore built a system that can track beats at three rhythmic levels: the quarter-note level, the half-note level, and the measure level. [1] The system not only finds the pulse sequence corresponding to the beats at the quarter-note level but also finds the beginnings of half notes and measures under the assumption that the time-signature is 4/4.

To build a real-time system that can output its beat interpretation along to a real-time input, it is necessary to utilize a beat-tracking algorithm that meets real-time requirements: it must process the input sequentially rather than in a back-and-forth or all-at-once manner. Although several previous systems (Desain and Honing, 1989, 1994, 1995; Smith, 1996) did not address the issue of predicting the next beat, a real-time beat-tracking algorithm needs to do just that. Several systems (Lee, 1985; Dannenberg and Mont-Reynaud, 1987; Allen and Dannenberg, 1990; Driesse, 1991; Rosenthal, 1992a,b; Desain, 1992; Rowe, 1993; Large, 1995) provide this capacity even if the real-time versions of their systems were not necessarily implemented. Since it is impossible to backtrack in performing beat tracking in real-time, several authors (Allen and Dannenberg, 1990; Rosenthal, 1992a,b) clarified the need for a strategy of pursuing multiple hypotheses in parallel and built their systems on such a strategy. Furthermore, Rosenthal (1992a,b) addressed the issue we are considering, detecting a hierarchical beat structure. Those systems, however, dealt with MIDI signals or clean onset times as their input. Since it is quite difficult to obtain complete MIDI representations from audio data, they cannot immediately be applied to complex audio signals.

Although several systems (Schloss, 1985; Katayose et al., 1989; Vercoe, 1994; Scheirer, 1996) dealt with audio signals, most of them did not consider higher-level musical structure such as the half-note and measure levels. Todd (1994, 1995) and Todd and Brown (1996) tackled this issue of detecting a hierarchical musical structure in a bottom-up fashion by using a multiscale smoothing model applied to onsets that were detected by a model of human auditory periphery. Previous practical MIDI-based systems (Allen and Dannenberg, 1990; Driesse, 1991; Rosenthal, 1992a,b) that employed musical heuristics to determine a more appropriate beat structure, in particular in situations when beat interpretation is ambiguous, have shown that a top-down process using musical heuristics provides more informative cues that a beat-tracking system can use to make appropriate musical decisions. It was difficult, however, to apply such musical heuristics to audio signals because of the difficulty of extracting musical elements such as chords and melodies in real-world audio signals.

We therefore developed a real-time beat-tracking system for audio signals (Goto and Muraoka, 1994, 1995a,b) under the assumption that the input contained drum-sounds (a bass drum and a snare drum). That system, though, was generally not able to track beats in audio signals without drum-sounds

[1] Although our system does not rely on score representation, for convenience here we use score-representing terminology like that used by Rosenthal (1992a,b). In our formulation the quarter-note level indicates the temporal basic unit that a human feels in music and that usually corresponds to a quarter note in scores.

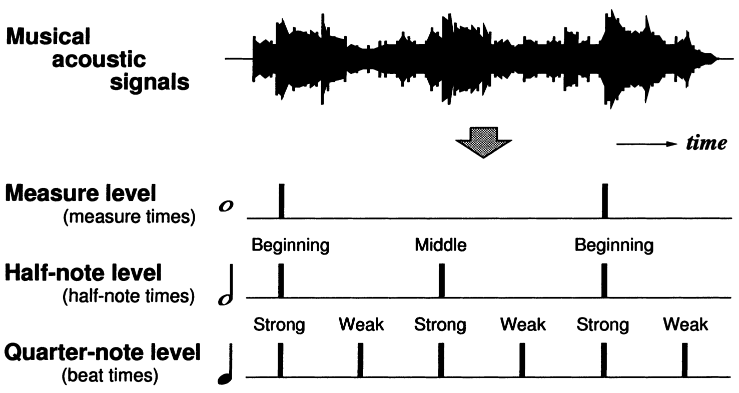

Musical acoustic signals
time
Measure level (measure times)
Beginning
Middle
Beginning
Half-note level (half-note times)
Strong
Weak
Strong
Weak
Strong
Weak
Quarter-note level (beat times)

Fig. 1. Beat-tracking problem.

because it relied on musical knowledge related to drum patterns. It was also unable to detect the beat structure at the measure level.

In the following sections we describe how we extended our previous system so that it can deal with drumless audio signals and detect the hierarchical beat structure comprising the three rhythmic levels in real time. We propose a method of detecting chord changes to make musical decisions about the audio signals by using heuristic musical knowledge. Because our method takes advantage of not requiring chord names to be identified, it can be applied to complex audio signals sampled from compact discs, signals in which chord identification is generally difficult.

2. Beat-tracking problem

In this section we specify the beat-tracking problem that we are dealing with and present the main difficulties of tracking beats.

2.1. Problem specification

In our formulation, beat tracking is defined as a process that organizes music into a hierarchical beat structure with three levels of rhythm: *the quarter-note level*, *the half-note level* and *the measure level* (Fig. 1). The first step in solving our beat-tracking problem is thus obtaining an appropriate sequence of *beat times* in an input musical audio signal. We define beat times as the temporal positions of almost regularly spaced beats corresponding to quarter notes, and the sequence of beat times is called the quarter-note level. The second step in solving our problem is then finding the beginnings of half notes and measures. The sequence of *half-note times* (temporal positions of strong beats [2]) is obtained by determining whether a beat is *strong* or *weak* (*half-note-level type*). The sequence of *measure times* (temporal positions of the beginnings of measures) is obtained by determining whether a half note is the *beginning* or the *middle* of a measure (*measure-level type*). The sequence of half-note times is called the half-note level

[2] Under the assumption that the time-signature of an input song is 4/4, in this paper a *strong beat* is either the first or third quarter note in a measure; a *weak beat* is the second or fourth.

and the sequence of measure times is called the measure level. Both half-note-level and measure-level types are called *beat types*.

To solve this problem, we assume that the time-signature of an input song is 4/4 and that its tempo is constrained to be between 61 M.M. (Mälzel's Metronome: the number of quarter notes per minute) and 120 M.M. and to be roughly constant. We also presuppose that a large class of popular music without drum-sounds has harmony transitions and chord changes.

2.2. Acoustic beat-tracking issues

Problematic issues that must be dealt with when tracking the hierarchical beat structure in real-world musical acoustic signals are (1) detecting beat-tracking cues in audio signals, (2) examining multiple hypotheses about beat positions and (3) making musical decisions. The simple technique of peak-finding with a threshold is not sufficient because there are many energy peaks that are not directly related to beats. Multiple interpretations of beats are possible at any given point because there is not necessarily a single specific sound that directly indicates the beat position. There are various ambiguous situations, such as ones where several events obtained by frequency analysis may correspond to a beat and where different *inter-beat intervals* (the temporal difference between two successive beats) seem plausible. In addition, it is necessary to make context-dependent decisions, such as determining the half-note-level and measure-level types and determining which is the best interpretation in an ambiguous situation.

In detecting tracking cues, it is necessary to detect several cues for different purposes: finding beat times and tracking the higher-level beat structure. Our previous system (Goto and Muraoka, 1995b) found beat times by first using frequency analysis to detect onset times and then using autocorrelation and cross-correlation of the onset times. The cues for tracking the higher-level beat structure of drumless audio signals, however, were not dealt with.

The multiple-hypothesis issue was addressed in our previous system (Goto and Muraoka, 1994, 1995a, 1996) by managing multiple agents that, according to different strategies, examined parallel hypotheses about beat positions. This multiple-agent architecture enables the system to cope with difficult beat-tracking situations: even if some agents lose track of beats, the system will track beats correctly as long as other agents maintain the correct hypothesis.

In making musical decisions, our previous system (Goto and Muraoka, 1995a,b) made use of pre-stored drum patterns, matching them with the drum pattern currently detected in the input signal. Although this method was effective, it of course cannot be applied to the drumless audio signals we are considering here.

In this paper we address the main issue in extending our previous system to drumless audio signals and to higher-level beat structure. The issue is that higher-level processing using musical knowledge in addition to lower-level signal processing is indispensable for tracking the higher-level beat structure and determining which is the best interpretation of beat positions in an ambiguous situation. Musical knowledge that is useful for analyzing musical scores or MIDI signals, however, cannot be immediately applied to raw audio signals because of the difficulty of obtaining MIDI-like representations of those signals.

3. Chord change detection for musical decisions

To address the above-mentioned higher-level processing issue, we propose a method for making musical decisions based on chord changes. In the following sections, we first describe a method of obtaining beat-tracking cues for the higher-level beat structure by detecting chord changes (Section 3.1) and then explain a way of making semantic decisions (musical decisions) by using heuristic musical knowledge based on those chord changes (Section 3.2). The main variables used in this section are listed in Table 1.

Table 1
List of the main variables

Variable	Description
t	Time
f	Frequency
$p(t,f)$	Power of the frequency spectrum
$T_{\mathrm{Q}}(n)$	nth beat time (provisional beat time)
$T_{\mathrm{E}}(n)$	nth eighth-note time (Eq. (3))
$C_{\mathrm{Q}}(n)$	Quarter-note chord-change possibility (Eq. (12))
$C_{\mathrm{E}}(n)$	Eighth-note chord-change possibility (Eq. (12))
$S_{\mathrm{Q}}(n)$	Frequency spectrum sliced at $T_{\mathrm{Q}}(n)$ (Eq. (1))
$S_{\mathrm{E}}(n)$	Frequency spectrum sliced at $T_{\mathrm{E}}(n)$ (Eq. (2))
L	A symbol representing the quarter-note level ‘Q’ and the eighth-note level ‘E’
$H_L(n,f)$	Histogram in $S_L(n)$ (Eq. (4))
$P_{\mathrm{hist}\ L}(n,f)$	Peaks along the frequency axis in $H_L(n,f)$ (Eq. (5))
$P_{\mathrm{reg}\ L}(n,f)$	Regularized peaks $H_L(n,f)$ (range: 0–1) (Eq. (6))
$P_{\mathrm{tran}\ L}(n,f)$	Finally-transformed peaks $H_L(n,f)$ (Eq. (9))
$\mathrm{clip}(x)$	Clipping function passing the range of 0 to 1 of x (Eq. (8))
$\mathrm{tend}_{\mathrm{H}}(n)$	Past tendency of every other $C_{\mathrm{Q}}(n)$ (Eq. (13))
$\mathrm{tend}_{\mathrm{M}}(n)$	Past tendency of every four $C_{\mathrm{Q}}(n)$ (Eq. (14))
$r_{\mathrm{judge}\ \mathrm{H}}(n)$	Reliability of judging the half-note-level type (Eq. (15))
$r_{\mathrm{judge}\ \mathrm{M}}(n)$	Reliability of judging the measure-level type (Eq. (16))
$r_{\mathrm{judge}\ \mathrm{Q}}(n)$	Reliability of judging that the quarter-note level is appropriate (Eq. (17))

3.1. Chord change detection

By making use of provisional beat times obtained on the basis of onset times (i.e., making use of beat times of a beat-position hypothesis as top-down information), this detection method examines possibilities of chord changes in a frequency spectrum 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 [3] 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, dominant frequency components during a certain period of time can be roughly identified by using a histogram of frequency components.

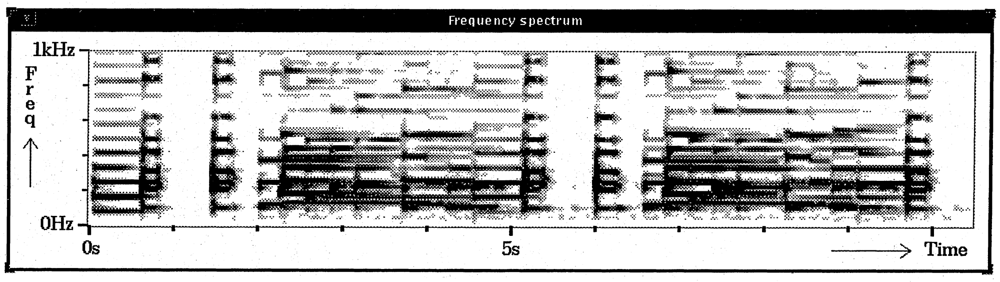
This method therefore calculates two kinds of possibilities of chord changes, one at the quarter-note level and the other at the eighth-note level, by slicing the frequency spectrum into strips at the provisional beat times (top-down information). We call the former the *quarter-note chord-change possibility* and the latter the *eighth-note chord-change possibility*. The quarter-note and eighth-note chord-change possibilities respectively represent how likely a chord is to change on each quarter-note position and on each eighth-note position under the current beat-position hypothesis. As described in Section 3.2, these possibilities are used for different purposes.

These possibilities are calculated as follows:

(1) *Slicing the frequency spectrum into spectrum strips*. The frequency spectrum (power spectrum) is calculated using the Fast Fourier Transform of the digitized audio signal (Section 4.1). In preparation for

[3] In the case of real-world songs, frequency components of a melody and other backing parts are also considered. These components tend to be in harmony with chord tones.

(a)



(b)

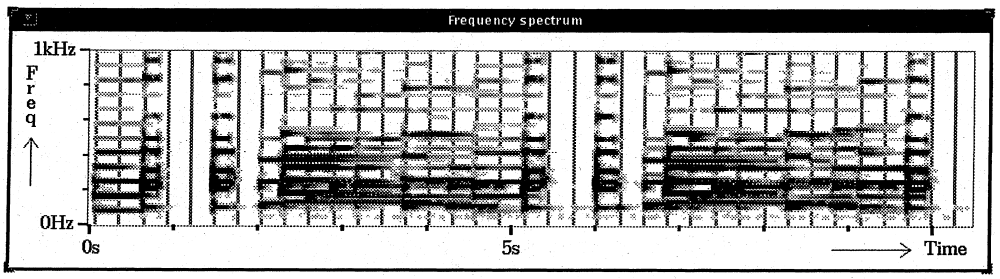


Fig. 2. Example of a frequency spectrum sliced into spectrum strips. (a) Frequency spectrum; (b) frequency spectrum sliced at the eighth-note times $T_{\mathrm{E}}(n)$.

calculating the quarter-note chord-change possibility $C_{\mathrm{Q}}(n)$, the frequency spectrum is sliced into spectrum strips $S_{\mathrm{Q}}(n)$ at the quarter-note times (beat times):

$$S_{\mathrm{Q}}(n) = \{p(t,f) \mid T_{\mathrm{Q}}(n) \leqslant t < T_{\mathrm{Q}}(n+1)\}, \tag{1}$$

where $T_{\mathrm{Q}}(n)$ is the nth beat time and $p(t,f)$ is the power of the spectrum of frequency f at time t. [4] In preparation for calculating the eighth-note chord-change possibility $C_{\mathrm{E}}(n)$, on the other hand, the spectrum is sliced into spectrum strips $S_{\mathrm{E}}(n)$ at the eighth-note times $T_{\mathrm{E}}(n)$ interpolated from $T_{\mathrm{Q}}(n)$:

$$S_{\mathrm{E}}(n) = \{p(t,f) \mid T_{\mathrm{E}}(n) \leqslant t < T_{\mathrm{E}}(n+1)\}, \tag{2}$$

$$T_{\mathrm{E}}(n) = \begin{cases} T_{\mathrm{Q}}(n/2) & (n \bmod 2 = 0), \\ (T_{\mathrm{Q}}((n-1)/2) + T_{\mathrm{Q}}((n+1)/2))/2 & (n \bmod 2 = 1). \end{cases} \tag{3}$$

Fig. 2 shows an example of a frequency spectrum sliced into spectrum strips. As shown in Fig. 2(b), the frequency spectrum shown in Fig. 2(a) is sliced at the eighth-note times interpolated from the provisional beat times.

(2) *Forming histograms*. The system forms histograms $H_{\mathrm{Q}}(n,f)$ and $H_{\mathrm{E}}(n,f)$ (after this, we will use abbreviations such as $H_L(n,f)$ $(L = \mathrm{Q},\mathrm{E})$) summed up along the time axis of the corresponding strip $S_{\mathrm{Q}}(n)$ and $S_{\mathrm{E}}(n)$:

$$H_L(n,f) = \sum_{t=T_L(n)+\mathrm{gap}_L(n)}^{T_L(n+1)-\mathrm{gap}_L(n)} p(t,f), \tag{4}$$

[4] f and t are integers, and $1f$ and $1t$ are respectively equal to the frequency resolution (10.77 Hz) and the discrete time step (11.61 ms).

where $\mathrm{gap}_L(n)$ is a margin that was introduced in order to avoid influences of noises and unstable frequency components around the note onset and that was empirically determined as $\mathrm{gap}_L(n) = (T_L(n+1) - T_L(n))/5$. Fig. 3(a) shows the histograms formed from the spectrum strips shown in Fig. 2(b).

(3) *Detecting dominant frequencies.* First, peaks $P_{\mathrm{hist}\ L}(n,f)$ along the frequency axis in $H_L(n,f)$ are given by

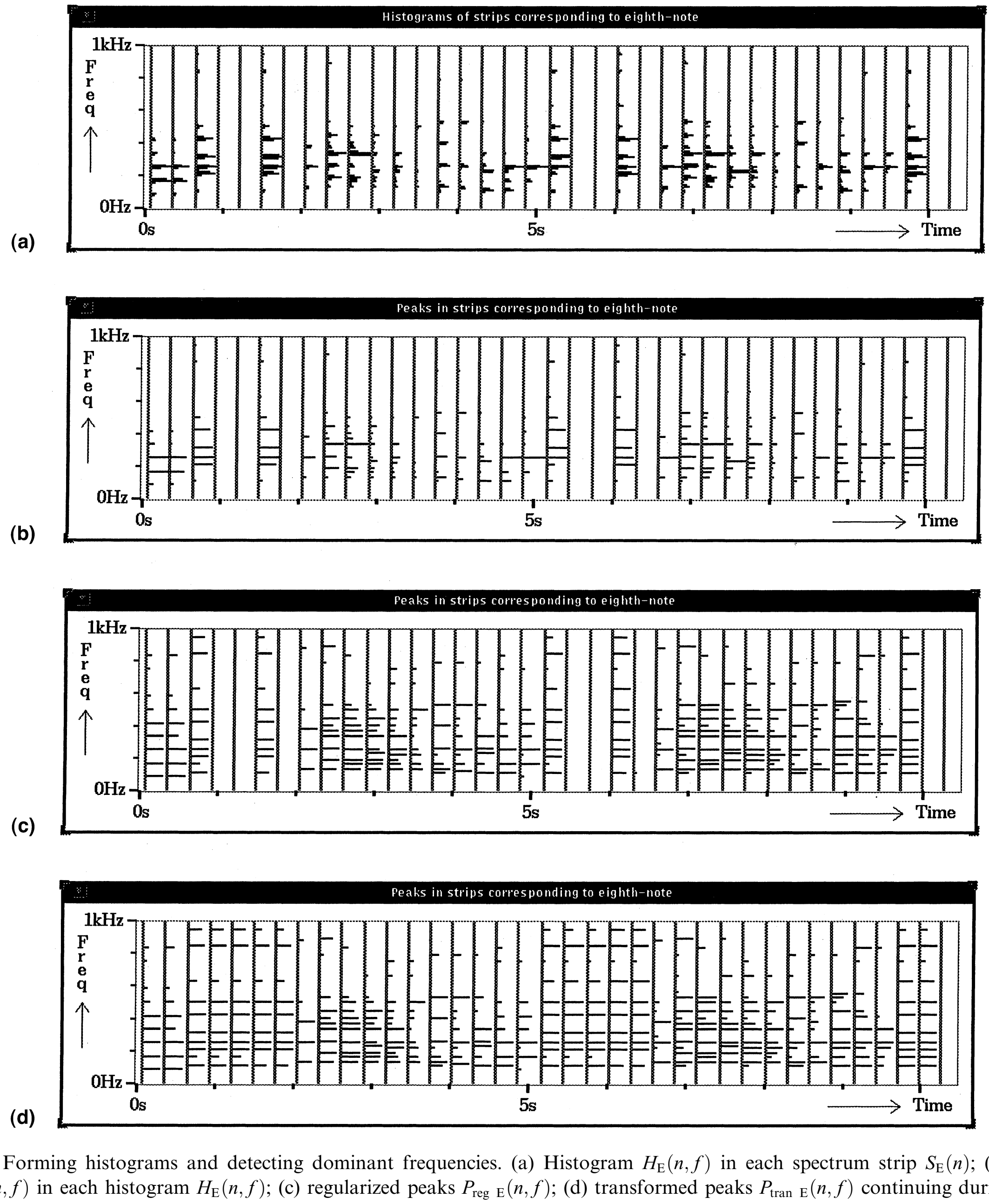


Fig. 3. Forming histograms and detecting dominant frequencies. (a) Histogram $H_{\mathrm{E}}(n,f)$ in each spectrum strip $S_{\mathrm{E}}(n)$; (b) peaks $P_{\mathrm{hist}\ \mathrm{E}}(n,f)$ in each histogram $H_{\mathrm{E}}(n,f)$; (c) regularized peaks $P_{\mathrm{reg}\ \mathrm{E}}(n,f)$; (d) transformed peaks $P_{\mathrm{tran}\ \mathrm{E}}(n,f)$ continuing during silent periods.

$$P_{\text{hist }L}(n,f) = \begin{cases} H_L(n,f) & \text{if } H_L(n,f) \geqslant H_L(n,f \pm 1), \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Our current implementation considers only peaks whose frequency is between 10 Hz and 1 kHz. These peaks can be considered the frequencies of the dominant tones in each strip and tend to correspond to frequency components of a chord or a melody. Fig. 3(b) shows peaks along the frequency axis in each histogram shown in Fig. 3(a).

The peaks $P_{\text{hist }L}(n,f)$ are then regularized into $P_{\text{reg }L}(n,f)$, which take values between 0 and 1. To avoid amplifying unnecessary noise peaks that appear during a musically silent period such as a rest, the system calculates $P_{\text{reg }L}(n,f)$ as a value relative to the recent maximum $M_{\text{recent }L}(n)$ of $P_{\text{hist }L}(n,f)$. In addition, the clipping function clip(x) (Eq. (8)) is applied after multiplying this relative value by a constant gain ratio GainRatio so that the absolute values of dominant peaks of $P_{\text{reg }L}(n,f)$ are large enough. We can thus express the regularized peaks $P_{\text{reg }L}(n,f)$ as

$$P_{\text{reg }L}(n,f) = \text{clip}\left(\text{GainRatio}\frac{P_{\text{hist }L}(n,f)}{M_{\text{recent }L}(n)}\right), \tag{6}$$

$$M_{\text{recent }L}(n) = \max\left(\max_f(P_{\text{hist }L}(n,f)),\ \text{AttnRatio}\ M_{\text{recent }L}(n-1)\right), \tag{7}$$

$$\text{clip}(x) = \begin{cases} 0 & (x < 0), \\ x & (0 \leqslant x \leqslant 1), \\ 1 & (1 < x), \end{cases} \tag{8}$$

where AttnRatio is a constant attenuation ratio which determines how long the previous local maximum affects the current value of the recent maximum and which will also be utilized in Eq. (11). The value of GainRatio was constrained to be at least 1 and the value of AttnRatio was constrained to be at least 0 and less than 1; those values were empirically set at GainRatio = 5 and AttnRatio = 0.99. Fig. 3(c) shows the regularized peaks calculated from the peaks shown in Fig. 3(b).

Finally the transformed peaks $P_{\text{tran }L}(n,f)$ in each strip are calculated so that the previous peaks $P_{\text{tran }L}(n-1,f)$ can be regarded as continuing during a relatively silent period in which the sum of the peaks $P_{\text{reg }L}(n,f)$ is low:

$$P_{\text{tran }L}(n,f) = \begin{cases} P_{\text{reg }L}(n,f) & \text{if } \sum_f P_{\text{reg }L}(n,f) \geqslant \text{SilentThres} \sum_f P_{\text{tran }L}(n-1,f), \\ P_{\text{tran }L}(n-1,f) & \text{otherwise,} \end{cases} \tag{9}$$

where SilentThres is a constant threshold used as the criterion for the silent period and was empirically set at 0.1. This transformation makes it possible to prevent the chord-change possibilities from increasing rapidly after every silent period. The transformed peaks shown in Fig. 3(d) were obtained from Fig. 3(c) and continue during silent periods.

(4) *Comparing frequencies between adjacent strips.* The chord-change possibilities are calculated by comparing peaks between adjacent strips: $P_{\text{tran }L}(n-1,f)$ and $P_{\text{tran }L}(n,f)$. When a chord is changed at the boundary time $T_L(n)$ between those strips, the peaks in $P_{\text{tran }L}(n,f)$ tend to differ from those in $P_{\text{tran }L}(n-1,f)$. Therefore, the chord-change possibility $C_L(n)$ is obtained as the result of normalizing the positive peak difference $P_{\text{diff }L}(n)$ given by Eq. (10). In order to normalize $P_{\text{diff }L}(n)$ into the range of 0–1, the system calculates $P_{\text{diff }L}(n)$ as a value relative to the recent maximum $M_{\text{diff }L}(n)$ of $P_{\text{diff }L}(n)$. Thus both the quarter-note chord-change possibility $C_{\text{Q}}(n)$ and the eighth-note chord-change possibility $C_{\text{E}}(n)$ are given by Eq. (12).

$$P_{\mathrm{diff}\ L}(n) = \sum_f \mathrm{clip}(P_{\mathrm{tran}\ L}(n,f) - P_{\mathrm{tran}\ L}(n-1,f)), \tag{10}$$

$$M_{\mathrm{diff}\ L}(n) = \max(P_{\mathrm{diff}\ L}(n),\ \ \mathrm{AttnRatio} M_{\mathrm{diff}\ L}(n-1)), \tag{11}$$

$$C_L(n) = \frac{P_{\mathrm{diff}\ L}(n)}{M_{\mathrm{diff}\ L}(n)}. \tag{12}$$

Fig. 4 shows examples of two kinds of chord-change possibilities obtained by the above method. The thin vertical lines in (a) represent the quarter-note times $T_Q(n)$ and those in (b) represent the eighth-note times $T_E(n)$. The beginning of measure occurs at every four quarter-note times from the extreme left in (a), and the beat occurs at every two eighth-note times from the extreme left in (b). In both (a) and (b), the horizontal lines above represent the peaks $P_{\mathrm{tran}\ L}(n,f)$ in each strip and the thick vertical lines below show the chord-change possibility $C_L(n)$.

3.2. Musical decisions

Utilizing the two kinds of chord-change possibilities, the system tracks the higher-level beat structure (i.e., determines the half-note times and the measure times) and selects the best of various agent-generated

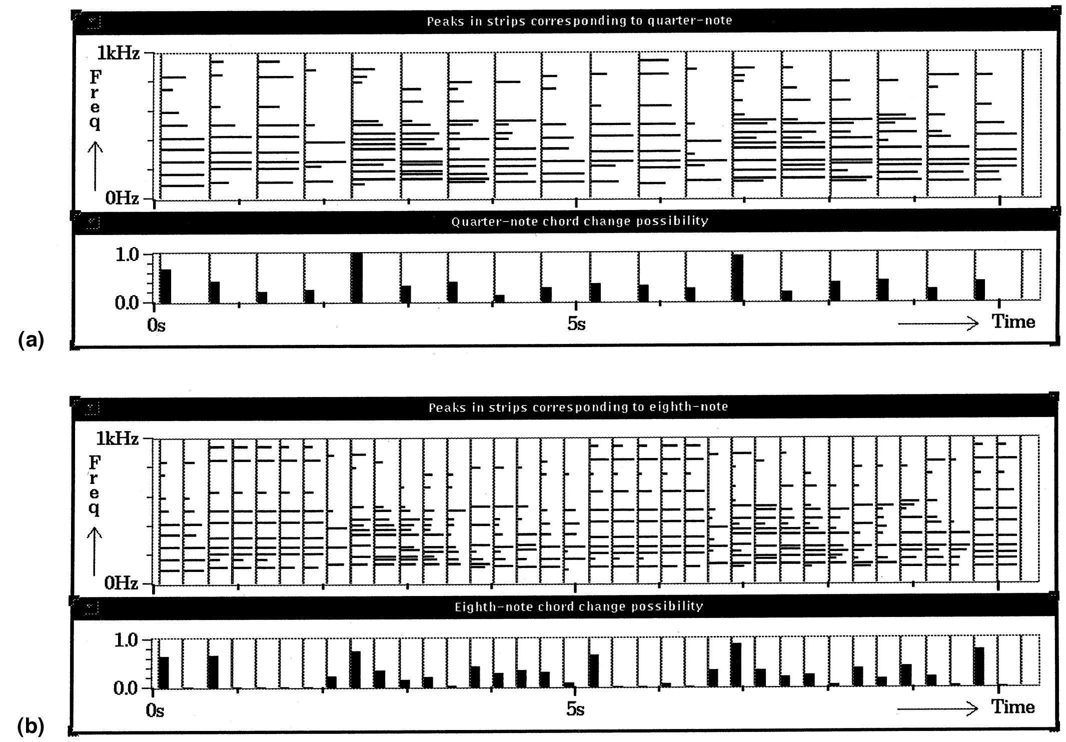


Fig. 4. Examples of peaks in sliced frequency spectrum and of chord-change possibilities. (a) Examining quarter-note chord-change possibility; (b) examining eighth-note chord-change possibility.

hypotheses about beat positions. For these purposes, we introduce the following two kinds of heuristic musical knowledge.

(1) *Quarter-note-level knowledge*. Chords are more likely to change at the beginnings of measures than at other positions. In other words, the quarter-note chord-change possibility tends to be higher on a strong beat than on a weak beat and higher on the strong beat at the beginning of a measure than on the other strong beat in the measure.

(2) *Eighth-note-level knowledge*. Chords are more likely to change on beats (quarter notes) than between adjacent beats. In other words, the eighth-note chord-change possibility tends to be higher on beats than on eighth-note displacement positions.

The system utilizes the quarter-note-level knowledge to detect the higher-level beat structure. It first calculates $\mathrm{tend}_{\mathrm{H}}(n)$, which represents a past tendency of every other quarter-note chord-change possibility, and $\mathrm{tend}_{\mathrm{M}}(n)$, which represents a past tendency of every four quarter-note chord-change possibility:

$$\mathrm{tend}_{\mathrm{H}}(n) = \mathrm{PastWeight}\ \ \mathrm{tend}_{\mathrm{H}}(n-2) + \mathrm{NowWeight}\ \ C_{\mathrm{Q}}(n), \tag{13}$$

$$\mathrm{tend}_{\mathrm{M}}(n) = \mathrm{PastWeight}\ \ \mathrm{tend}_{\mathrm{M}}(n-4) + \mathrm{NowWeight}\ \ C_{\mathrm{Q}}(n), \tag{14}$$

where PastWeight is a weight factor determining how much the past values (of $C_{\mathrm{Q}}(n)$) are taken into consideration and NowWeight is a weight factor determining how much the current value (of $C_{\mathrm{Q}}(n)$) is taken into consideration. Those constant values were empirically set at $\mathrm{PastWeight} = 0.99$ and $\mathrm{NowWeight} = 0.2$. The value of $\mathrm{tend}_{\mathrm{H}}(n)$ thus becomes higher when $C_{\mathrm{Q}}(n)$ tends to be higher on a strong beat, which occurs on every other quarter note, and the value of $\mathrm{tend}_{\mathrm{M}}(n)$ becomes higher when $C_{\mathrm{Q}}(n)$ tends to be higher on the beginning of a measure, which occurs on every fourth quarter note.

If $\mathrm{tend}_{\mathrm{H}}(n) - \mathrm{tend}_{\mathrm{H}}(n-1) > \mathrm{TendThres}_{\mathrm{H}}$, the system judges that the position of a half-note time is $T_{\mathrm{Q}}(n)$, where $\mathrm{TendThres}_{\mathrm{H}}$ $(=0.3)$ is a constant threshold for this judgement. If $T_{\mathrm{Q}}(n)$ is a half-note time and $\mathrm{tend}_{\mathrm{M}}(n) - \mathrm{tend}_{\mathrm{M}}(n-2) > \mathrm{TendThres}_{\mathrm{M}}$, the system judges that the position of a measure time is $T_{\mathrm{Q}}(n)$, where $\mathrm{TendThres}_{\mathrm{M}}$ $(=0.2)$ is a constant threshold. The reliabilities of these judgements are defined as

$$r_{\mathrm{judge\ H}}(n) = \mathrm{clip}(|\mathrm{tend}_{\mathrm{H}}(n) - \mathrm{tend}_{\mathrm{H}}(n-1)|), \tag{15}$$

$$r_{\mathrm{judge\ M}}(n) = \mathrm{clip}(|\mathrm{tend}_{\mathrm{M}}(n) - \mathrm{tend}_{\mathrm{M}}(n-2)|). \tag{16}$$

Using the previous positions of a half-note time and a measure time, the system determines the following beat types (half-note-level type and measure-level type) under the assumptions that *strong* and *weak* alternate on beat times and that *beginning* and *middle* alternate on half-note times.

To select the best hypothesis, the system utilizes the eighth-note-level knowledge. As described in Section 4.2, the final output is determined on the basis of the appropriate hypothesis that has the highest reliability. To evaluate the reliability of a hypothesis, the system calculates $r_{\mathrm{judgeQ}}(n)$, which is the reliability of the judgement that $T_{\mathrm{Q}}(n)$ $(=\ T_{\mathrm{E}}(2n))$ is the position of a beat:

$$r_{\mathrm{judge\ Q}}(n) = \mathrm{PastWeight}\ \ r_{\mathrm{judge\ Q}}(n-1) + \mathrm{NowWeight}(C_{\mathrm{E}}(2n) - C_{\mathrm{E}}(2n+1)). \tag{17}$$

If $r_{\mathrm{judge\ Q}}(n)$ becomes high enough (i.e., the eighth-note chord-change possibility tends to be higher on beats than on other positions), the reliability value is increased so that the system can select the hypothesis under which the appropriate $C_{\mathrm{E}}(n)$ is obtained. The reliability is also evaluated from different viewpoints as described in Section 4.2.

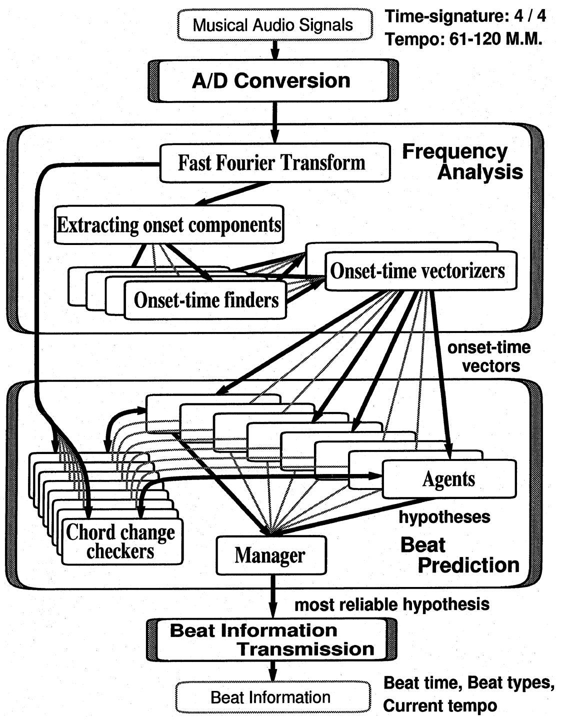


Fig. 5. Overview of our beat-tracking system.

4. System description

This section briefly describes our beat-tracking system for musical audio signals without drum-sounds. [5] It provides, as real-time output, a description called *beat information* (BI) that consists of the beat time, its beat types, and the current tempo. Fig. 5 shows an overview of the system. The system first digitizes an input audio signal in the *A/D conversion* stage. Then in the *frequency analysis* stage, multiple onset-time finders detect onset times in different ranges of the frequency spectrum, and those results are transformed into vectorial representation (called *onset-time vectors*) by *onset-time vectorizers*. In the *beat prediction* stage, the system manages multiple agents that, according to different strategies, make parallel hypotheses based on those onset-time vectors. Each agent first calculates the inter-beat interval and predicts the next beat time. By communicating with a *chord change checker*, it then determines the beat types and evaluates the reliability of its own hypothesis. A *hypotheses manager* gathers all hypotheses and then determines the final output on the basis of the most reliable one. Finally, in the *BI Transmission* stage, the system transmits BI to application programs via a computer network.

4.1. Frequency analysis

In the *frequency analysis* stage, the frequency spectrum and several sequences of N-dimensional onset-time vectors are obtained for later processing (Fig. 6). The full frequency band is split into several

[5] For detailed descriptions of our beat-tracking system for audio signals that include drum-sounds, see (Goto and Muraoka, 1995a,b).

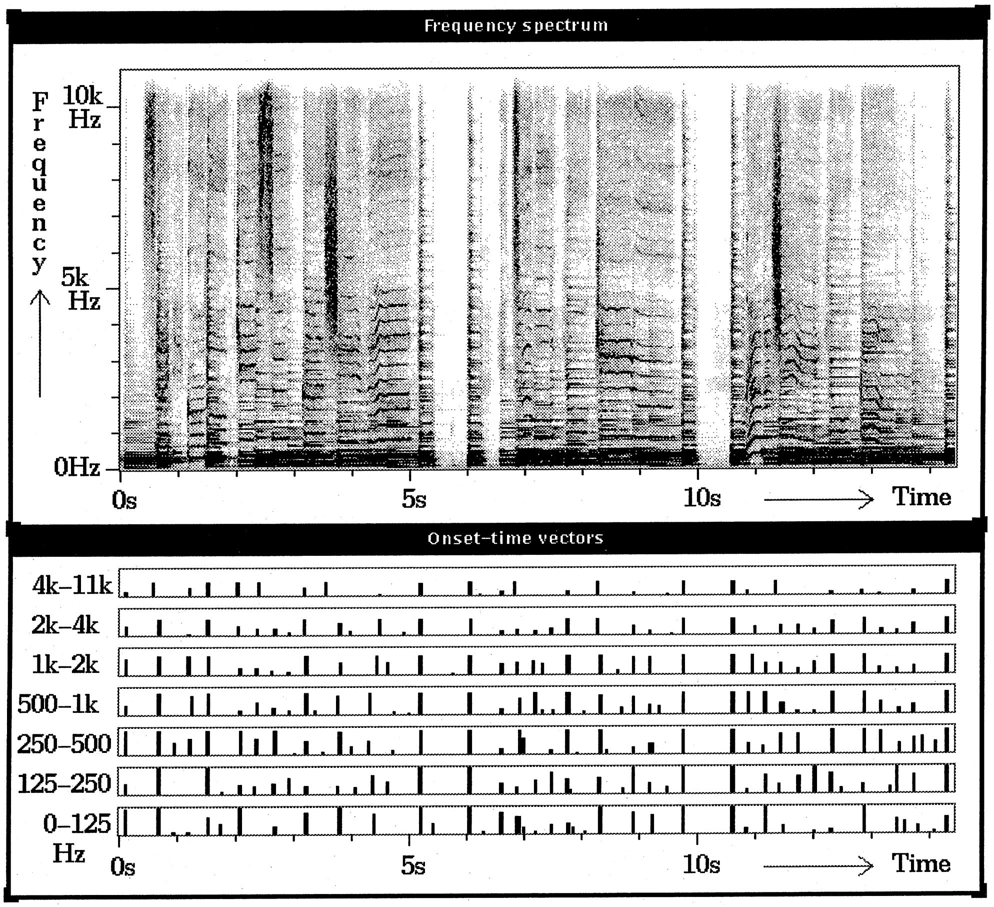


Fig. 6. Examples of a frequency spectrum and of an onset-time vector sequence.

frequency ranges, and each dimension of the onset-time vectors corresponds to a different frequency range. This representation makes it possible to consider onset times of all the frequency ranges at the same time.

4.1.1. Fast Fourier transform (FFT)

The frequency spectrum is calculated with the FFT using the Hanning window. Each time the FFT is applied to the input signal, the window is shifted to the next frame.

In our current implementation, the input signal is digitized at 16 bit/22.05 kHz, and two kinds of FFT are calculated. One FFT, for extracting onset components in the Frequency analysis stage, is calculated with a window size of 1024 samples, and the window is shifted by 256 samples. The frequency resolution is consequently 21.53 Hz and the discrete time step (1 *frame-time* [6]) is 11.61 ms. The other FFT, for examining chord changes in the Beat prediction stage, is simultaneously calculated in audio down-sampled at 16 bit/

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

11.025 kHz with a window size of 1024 samples, and the window is shifted by 128 samples. The frequency resolution and the time step are consequently 10.77 Hz and 1 frame-time.

4.1.2. Extracting onset components

The frequency component $p(t,f)$ that meets the following Condition (18) is extracted as an onset component.

$$\min(p(t,f), p(t+1,f)) > pp, \tag{18}$$

$$pp = \max(p(t-1,f), p(t-1,f \pm 1)). \tag{19}$$

The degree of onset $d(t,f)$ (rapidity of increase in power) is given by

$$d(t,f) = \begin{cases} \max(p(t,f), p(t+1,f)) - pp & \text{if Condition (18) is fulfilled,} \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

4.1.3. Onset-time finders

Multiple onset-time finders (seven in our current implementation) detect onset times in several frequency ranges (0–125 Hz, 125–250 Hz, 250–500 Hz, 500 Hz–1 kHz, 1–2 kHz, 2–4 kHz, and 4–11 kHz). Each onset time is given by the peak time found by peak-picking in the sum $D(t)$ along the time axis, where $D(t) = \sum_f d(t,f)$. The sum $D(t)$ is linearly smoothed with a convolution kernel before its peak time is calculated. Limiting the frequency range of $\sum_f$ makes it possible to find onset times in the different frequency ranges.

4.1.4. Onset-time vectorizers

Each onset-time vectorizer transforms the results of all onset-time finders into a sequence of onset-time vectors: the same onset times in all the frequency ranges are put together into one vector. In the current system, three vectorizers transform onset times from seven finders into three sequences of seven-dimensional onset-time vectors with the different sets of frequency weights (focusing on all, low, and middle frequency ranges) (Goto and Muraoka, 1996). These results are sent to agents of the beat prediction stage.

4.2. Beat prediction

Multiple agents interpret the sequences of onset-time vectors according to different strategies and maintain their own hypotheses (Goto and Muraoka, 1994, 1995a, 1996). Each hypothesis consists of a predicted next-beat time, its beat types (half-note-level type and measure-level type), and the current inter-beat interval (Fig. 7). These hypotheses are gathered by the manager and the most reliable one is considered the system output.

All agents are grouped into pairs. [7] The two agents in a pair examine the same inter-beat interval and cooperatively predict the time of the next beat; their two predictions will always differ by half the inter-beat interval. For this purpose, one agent interacts with the other through a *prediction field*, which is an expectancy curve [8] that represents the time that the next beat is expected to occur (Fig. 8). The height of each local peak in the prediction field can be interpreted as the probability that the next beat is at that position. The two agents interact with each other by inhibiting each other's prediction field: the beat time of the

[7] In our current implementation there are twelve agents grouped into six pairs.

[8] Other systems (Desain, 1992; Desain and Honing, 1994; Vercoe, 1994) have used a similar expectancy-curve concept for predicting future events but not for managing interactions between agents.

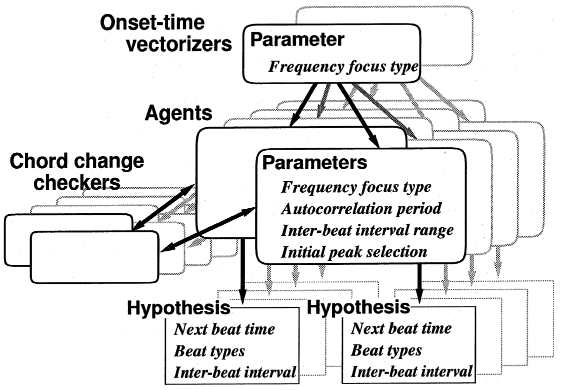


Fig. 7. Relations between onset-time vectorizers, agents, and chord change checkers.

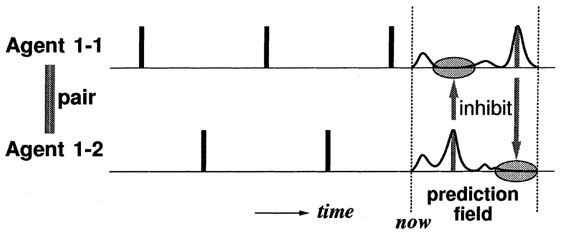


Fig. 8. Agent interaction through a prediction field.

hypothesis of each agent reduces the probability of a beat in the temporally corresponding neighborhood in the other's field.

For each agent, the following four parameters determine its strategy for making the hypothesis (Fig. 7). Initial settings of the parameters are listed in Table 2.

Table 2
Initial settings of the strategy parameters

Pair-agent	Frequency focus type	Auto-correlation period (f.t.)	Inter-beat interval range (f.t.)	Initial peak selection
1–1	Type-all	500	43–85	Primary
1–2	Type-all	500	43–85	Secondary
2–1	Type-all	1000	43–85	Primary
2–2	Type-all	1000	43–85	Secondary
3–1	Type-low	500	43–85	Primary
3–2	Type-low	500	43–85	Secondary
4–1	Type-low	1000	43–85	Primary
4–2	Type-low	1000	43–85	Secondary
5–1	Type-mid	500	43–85	Primary
5–2	Type-mid	500	43–85	Secondary
6–1	Type-mid	1000	43–85	Primary
6–2	Type-mid	1000	43–85	Secondary

'f.t.' is the abbreviation of frame-time (11.61 ms).

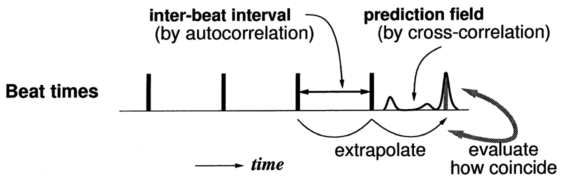


Fig. 9. Predicting the next beat.

(1) *Frequency focus type.* This parameter determines which vectorizer an agent receives onset-time vectors from. Its value is chosen from among *type-all*, *type-low* and *type-mid*, respectively corresponding to vectorizers focusing on all frequency ranges, low frequency ranges and middle frequency ranges.

(2) *Autocorrelation period.* This parameter determines the window size for calculating the vectorial autocorrelation (described later) of the onset-time vector sequence. The greater its value, the older the onset-time information considered.

(3) *Inter-beat interval range.* This parameter controls the range of possible inter-beat intervals. As described later, it limits the range in the result of the vectorial autocorrelation, within which a peak is selected.

(4) *Initial peak selection.* This parameter takes a value of either *primary* or *secondary*. When the value is primary, the largest peak in the prediction field is initially selected and considered the next beat time; when the value is secondary, the second-largest peak is initially selected. This selection helps generate a variety of hypotheses.

4.2.1. Beat-predicting agents

Each agent makes a hypothesis as follows and sends it to both the one-to-one corresponding chord-change checker and the manager.

(1) *Determining the inter-beat interval.* To determine the inter-beat interval, each agent receives the sequence of onset-time vectors and calculates its vectorial autocorrelation. [9] The windowed and normalized vectorial autocorrelation function Ac(τ) is defined as

$$\mathrm{Ac}(\tau) = \frac{\sum_{t=c-\mathrm{AcPeriod}}^{c} \mathrm{win}(c-t, \mathrm{AcPeriod})(\vec{o}(t) \cdot \vec{o}(t-\tau))}{\sum_{t=c-\mathrm{AcPeriod}}^{c} \mathrm{win}(c-t, \mathrm{AcPeriod})(\vec{o}(t) \cdot \vec{o}(t))}, \tag{21}$$

where $\vec{o}(t)$ is the N-dimensional onset-time vector at time t, c is the current time and AcPeriod is the strategy parameter *autocorrelation period* (Goto and Muraoka, 1996). The window function $\mathrm{win}(t, s)$ whose window size s is given by

$$\mathrm{win}(t, s) = \begin{cases} 1.0 - 0.5t/s & 0 \leqslant t \leqslant s, \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

The inter-beat interval is given by τ with the maximum height in Ac(τ) within the range limited by the parameter *inter-beat interval range*. If the reliability of a hypothesis becomes high enough, its agent tunes this parameter to narrow the range of possible inter-beat intervals so that it examines only a neighborhood of the current appropriate one. This is effective in stabilizing the beat-tracking output because the autocorrelation result tends to contain several unnecessary and confusing peaks around the correct peak pursued by an agent whose hypothesis has a high reliability.

(2) *Predicting the next beat time.* To predict the next beat time, each agent forms a prediction field (Fig. 9). The prediction field is the result of calculating the windowed cross-correlation function Cc(τ)

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

between the sum $O(t)$ of all dimensions of $\vec{o}\,(t)$ and the provisional beat-time sequence $T_{\mathrm{tmp}}(t,m)$ whose interval is the inter-beat interval obtained using Eq. (21):

$$\mathrm{Cc}(\tau) = \sum_{t=c-\mathrm{CcPeriod}}^{c} \left(\mathrm{win}(c-t, \mathrm{CcPeriod})\, O(t) \sum_{m=1}^{\mathrm{CcNumBeats}} \delta(t - T_{\mathrm{tmp}}(c+\tau, m)) \right), \tag{23}$$

$$T_{\mathrm{tmp}}(t,m) = \begin{cases} t - I(t) & (m = 1), \\ T_{\mathrm{tmp}}(t, m-1) - I(T_{\mathrm{tmp}}(t, m-1)) & (m > 1), \end{cases} \tag{24}$$

$$\delta(x) = \begin{cases} 1 & (x = 0), \\ 0 & (x \neq 0), \end{cases} \tag{25}$$

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 $\mathrm{Cc}(\tau)$ where $0 \leqslant \tau \leqslant I(c) - 1$.

Each agent then selects the next beat time from local peaks in the prediction field after the field is inhibited by its paired agent. When the reliability of a hypothesis is low, the agent initially selects the peak in the prediction field according to the parameter *initial peak selection* and then tries to pursue the peak close to the sum of the previously selected one and the inter-beat interval.

(3) *Judging the beat types*. Each agent determines the beat types of the predicted beat time according to the half-note time and the measure time. As described in Section 3.2, these times are obtained from the quarter-note chord-change possibility received from the corresponding chord-change checker.

(4) *Evaluating the reliability of its own hypothesis*. Each agent finally evaluates the reliability of its hypothesis in the following three steps. First, the reliability is evaluated according to how the next beat time predicted on the basis of the onset times coincides with the time extrapolated from the past beat times (Fig. 9). If they coincide, the reliability is increased; otherwise, the reliability is decreased. Second, the reliability is evaluated according to how appropriate the eighth-note chord-change possibility is. If $r_{\mathrm{judge\ Q}}(n)$ (defined in Section 3.2) is high enough, the reliability is increased; otherwise, the reliability is decreased. Third, the reliability is evaluated according to how appropriate the quarter-note chord-change possibility is. If $r_{\mathrm{judge\ H}}(n)$ is high enough, the reliability is increased a little.

4.2.2. Chord change checkers

Each chord-change checker examines the two kinds of chord-change possibilities as described in Section 3.1. It analyzes the frequency spectrum on the basis of beat times (top-down information) received from the one-to-one corresponding agent, and it sends the possibilities back to the agent (Fig. 7).

4.2.3. Hypotheses manager

The manager classifies all agent-generated hypotheses into groups according to beat time and inter-beat interval. Each group has an overall reliability given by the sum of the reliabilities of the group's hypotheses. The manager then selects the dominant group that has the highest reliability. Since an incorrect group could be selected if temporarily unstable beat times split the appropriate dominant group, the manager repeats grouping and selecting three times while narrowing the margin of beat times allowable for being classified into the same group. The reliable hypothesis in the most dominant group is thus selected as the output and sent to the BI Transmission stage.

The manager updates the beat types in the output using only the beat types that were labeled when $r_{\mathrm{judge\ H}}(n)$ and $r_{\mathrm{judge\ M}}(n)$ were high compared with the recent maximum, since the beat types labeled by each agent might be incorrect because of a local irregularity of chord changes or a detection error.

Table 3
Results of testing chord change detection

	$C_Q(n)$		$C_E(n)$	
	CH	NC	on $T_Q(n)$ (CH, NC)	off $T_Q(n)$
mean	0.73	0.01	0.56 (0.81, 0.30)	0.03
SD	0.22	0.02	0.29 (0.18, 0.08)	0.05
max.	1.00	0.10	1.00 (1.00, 0.48)	0.21
min.	0.28	0.00	0.12 (0.37, 0.12)	0.00

CH: chord change. NC: no chord change.

5. Experiments and results

In the following, we first describe an experimental result of testing the proposed method of detecting chord changes (Section 5.1) and then show the overall beat detection rates of the system implemented on a parallel computer, the Fujitsu AP1000 [10] (Section 5.2). We then report the result of our attempt to evaluate the difficulty of tracking beats in an input audio signal (Section 5.3) and describe an experimental result of evaluating the contribution of the proposed method of making musical decisions based on chord changes (Section 5.4). Finally, we summarize those results and introduce a beat-tracking application (Section 5.5).

5.1. Testing chord change detection

We tested the basic performance of the method of chord-change detection proposed in Section 3.1 by using a random chord progression. This chord progression consisted of one hundred chord transitions of 101 chords that were randomly selected from sixty kinds of chords: the twelve kinds of root (A, A♯, B, C, C♯, D, D♯, E, F, F♯, G, G♯) with the five chord types (major triad, minor triad (m), dominant 7th chord (7), minor 7th chord (m7), major 7th chord (M7)). These chords were so selected that the adjacent chords were different. Using a synthesizer's piano tone, we played them in the basic root position (close position voicing). The fundamental frequency of the chord root note was between 110 Hz and 208 Hz. To examine the case in which the chord did not change, we played each chord twice with the duration of a quarter note (600 ms) under the tempo 100 MM.

The mean, standard deviation (SD), maximum, and minimum of the quarter-note chord-change possibility $C_Q(n)$ and the eighth-note chord-change possibility $C_E(n)$ obtained when the appropriate beat times were provided for slicing the frequency spectrum are listed in Table 3. The 'CH' and 'NC' of the $C_Q(n)$ in Table 3 are respectively the $C_Q(n)$ when a chord was changed at $T_Q(n)$ and the $C_Q(n)$ when a chord was not changed at $T_Q(n)$. The values listed in these columns indicate that the $C_Q(n)$ at chord changes (CH) were appropriately high.

On the other hand, the 'on $T_Q(n)$' and 'off $T_Q(n)$' of the $C_E(n)$ respectively mean the $C_E(n)$ on beats ($n \bmod 2 = 0$) and the $C_E(n)$ on eighth-note displacement positions ($n \bmod 2 = 1$). In the case of the 'on $T_Q(n)$', because the chord-change case (CH) alternated with the no-chord-change case (NC), these cases were also analyzed separately. The values listed in these columns indicate that chord changes were appropriately detected using $C_E(n)$. The $C_E(n)$ of NC of 'on $T_Q(n)$' tended to be higher than the $C_E(n)$ of 'off $T_Q(n)$' because the chord notes were always played at a beat time, whereas all frequency components present on an eighth-note displacement position persisted from the previous beat time.

[10] The AP1000 (Ishihata et al., 1991) consists of 64 processing elements and its performance is at most 960 MIPS and 356 MFLOPS. Although the AP1000 had relatively huge computing power when we started our research five years ago, those values of the AP1000's performance imply that our system might now be implemented on an up-to-date personal computer.

5.2. Overall result

We first introduce the method we used for evaluating our system and then report beat detection rates of songs, how quickly the system started to track the correct beats, and how accurately the system obtained the beat, half-note, and measure times.

5.2.1. Evaluation method

We designed a measure for analyzing the beat-tracking accuracies at the quarter-note, half-note, and measure levels. The basic concept of this measure is to compare the beat times of the system output (*the examined times*) [11] with the hand-labeled beat times (*the correct times*). In other words, we considered subjective hand-labeled beat positions to be the correct beat times. Since the beat is a perceptual concept that a person feels in music, it is generally difficult to define the correct beat in an objective way.

To label the correct beat positions, we developed a beat-position 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. The positions can be finely adjusted by playing back the audio with click tones at beat times, and the user also defines a hierarchical beat structure – the quarter-note, half-note, and measure levels – corresponding to the audio signal. This enables the correct beat times to be more accurate than the results of human tapping containing relatively large timing deviations.

The beat-tracking accuracies are each represented as a measurement set {Q, H, M}$[\zeta, \mu, \sigma, M]$, where Q[ ], H[ ], and M[ ] respectively represent the measures at the quarter-note, half-note, and measure levels. The term $\zeta = [A_{Ns}$ s, A_{Ne} s] is the correctly tracked period (the period between A_{Ns} and A_{Ne} in which the beat is tracked correctly). In particular, $[A_{Ns}, -]$ means that the beat-tracking system keeps on tracking the correct beat if once it starts to track the correct one at A_{Ns}. The terms μ, σ and M are respectively the mean, standard deviation, and maximum of the normalized difference (deviation error) between the correct time and the examined time. If the normalized difference is 1, it indicates that the difference is half the correct inter-beat interval. [12]

5.2.2. Beat detection rates

We tested the system on 40 songs, each at least one minute long, performed by 28 artists (Table 4). The input monaural audio signals were sampled from commercial compact discs of popular music and contained the sounds of various instruments (but not drums). The time-signature was 4/4 and the tempi ranged from 62–116 MM and were roughly constant.

We judged that a song was tracked correctly at a certain rhythmic level if the corresponding measurement set of the song fulfilled {Q, H, M}$[[A_{Ns} < 45.0$ s, $A_{Ne} = -]$, $\mu < 0.2$, $\sigma < 0.2$, $M < 0.35]$. In our experiment the system correctly tracked beats at the quarter-note level in 35 of the 40 songs (87.5%) [13] and correctly tracked the half-note level in 34 of the 35 songs in which the correct beat times were obtained (97.1%) Moreover, it correctly tracked the measure level in 32 of the 34 songs in which the correct half-note times were obtained (94.1%) (Table 4).

The beat times were not obtained correctly in five songs because onset times were very few and irregular or the tempo fluctuated temporarily. Consequently, the chord-change possibilities in those songs could not be obtained appropriately because those possibilities depend on the beat times. The main reason that the half-note-level or measure-level type was incorrect in the other mistaken songs was inconsistency of chord

[11] In evaluating the quarter-note, half-note and measure levels, we respectively use the beat, half-note and measure times.

[12] The detailed definition of this measure is described by Goto and Muraoka (1997).

[13] In evaluating the tracking accuracy of our system, we did not count unstably tracked songs (those for which correct beats were obtained just temporarily).

Table 4
Song list

Title (Artist)	Result [a]	Tempo (M.M.)
Ame (Chisato Moritaka)	o o o	62
Konoyoruni (Yumi Tanimura)	o o o	64
Suki (DREAMS COME TRUE)	o o o	65
Anatawo Mitsumete (K.ODA)	× × ×	65
For You (Katsumi)	o o o	68
Futari (Maki Ohguro)	o o o	68
Mayonaka no Love Song (T-BOLAN)	o o o	68
Kimini Aete (Tetsuya Komuro)	o o o	70
Futarino Natsu (ZARD)	o o o	72
Blue Star (Mayo Okamoto)	o o o	72
Listen to me (ZARD)	o o o	73
Harunohi (Miki Imai)	× × ×	74
No More Rhyme [Acoustic Mix] (Debbie Gibson)	o o o	74
My Heart Ballad (Yoko Minamino)	o o o	75
Love is... (B'z)	o o o	75
Fubukino Nakawo (Yumi Matsutoya)	o o o	76
Roots of The Tree (Naoto Kine)	o o o	76
Itukano Merry Christmas [Reprise] (B'z)	o o o	78
Now and Forever (Richard Marx)	o o o	78
Dandelion – *Osozakino Tanpopo* (Yumi Matsutoya)	o o o	78
Afureru Omoino Subetewo... (Miho Morikawa)	o o o	81
You're My Life (Komi Hirose)	o o o	82
Alone (Heart)	o o ×	88
Ruriirono Chikyuu (Seiko Matsuda)	× × ×	88
Love – *Nemurezuni Kiminoyokogao Zuttomiteita* – (ZARD)	o o o	89
Right Here Waiting (Richard Marx)	× × ×	89
Seasons (B'z)	o o o	90
Strangers Of The Heart (Heart)	o o o	91
Mitsumeteitaine (ZARD)	o o o	92
Mia Maria (ORIGINAL LOVE)	o o o	95
Anatani Aiwo (Yumi Tanimura)	o o o	100
I Wish (Misato Watanabe)	o o o	100
I Won't Hold You Back (TOTO)	o o o	102
amour au chocolat (Miki Imai)	o o o	106
Lazy Afternoon (STARDUST REVUE)	o o o	108
Whispers (Fairground Attraction)	o o ×	111
Nijiwo Watarou (Hitomi Yuki)	o o o	112
Too far gone (Incognito)	× × ×	112
Resistance (Tetsuya Komuro)	o o o	115
Do You Want To Know A Secret (Fairground Attraction)	o ××	116

[a] o o o: Song that was tracked correctly; o o ×: Song that was not tracked at the measure level; o × ×: Song that was not tracked at the half-note level; × × ×: Song that was not tracked at the quarter-note level.

changes with the heuristic musical knowledge; there were songs where chords changed at every quarter-note or every other quarter-note.

We then compared the performance of this system with that of our previous system (Goto and Muraoka, 1995b) for music with drum-sounds. The previous system was also tested on the same 40 songs, and the results of this comparison are listed in Table 5, which shows that the beat detection rates were remarkably improved by our system extension.

Table 5
Performance improvement compared to our previous system

Beat structure	Proposed system	Previous system (Goto and Muraoka, 1995b)
Measure level	32/34 songs (94.1%)	0/2 songs (0.0%)
Half-note level	34/35 songs (97.1%)	2/9 songs (22.2%)
Quarter-note level	35/40 songs (87.5%)	9/40 songs (22.5%)

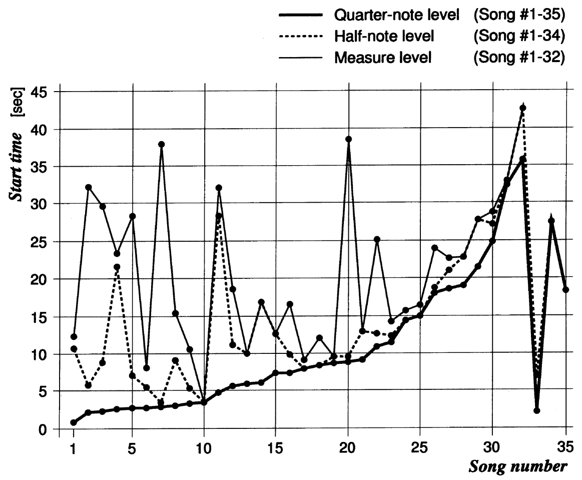


Fig. 10. Start time (A_{Ns}) of tracking the correct beats at the quarter-note, half-note and measure levels.

5.2.3. Tracking quickness

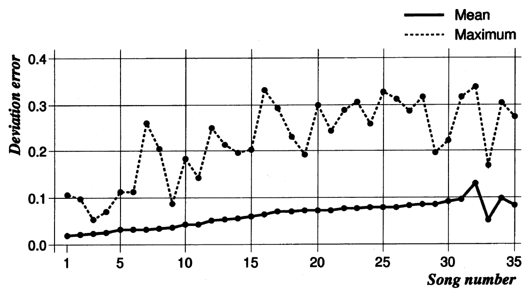
We evaluated how quickly the system started to track the correct beats stably at each rhythmic level by using the A_{Ns} of each measurement set. Fig. 10 shows the A_{Ns} of the songs correctly tracked at the quarter-note, half-note, and measure levels. The horizontal axis represents the song numbers (#) arranged in order of A_{Ns} of the quarter-note level up to song #32. The mean, minimum, and maximum of the A_{Ns} of all the correctly tracked songs are listed in Table 6. 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.

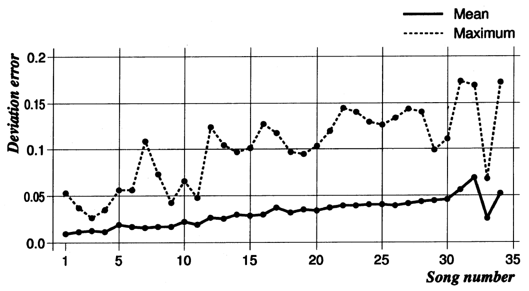
5.2.4. Tracking accuracy

We evaluated how accurate the examined times (the system output) were by using the μ, σ, and M of each measurement set. Fig. 11 shows the μ and M of the correctly-tracked songs at the three rhythmic

Table 6
Start time (A_{Ns}) of tracking the correct beats at the quarter-note, half-note and measure levels

Rhythmic level	mean	min	max
Quarter-note level	10.71 s	0.79 s	35.77 s
Half-note level	14.70 s	3.42 s	42.56 s
Measure level	20.70 s	3.42 s	42.56 s

(a)


(b)


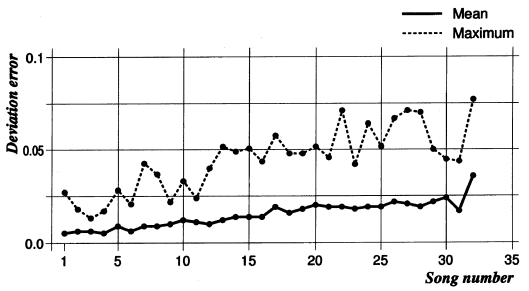
(c)


Fig. 11. Mean (μ) and maximum (M) of the deviation error at the quarter-note, half-note, and measure levels. (a) Quarter-note level (Song # 1-35), (b) half-note level (Song # 1-34), (c) measure level (Song # 1-32).

levels. The horizontal axis represents the song numbers rearranged in order of μ of the quarter-note level up to song #32. The mean, minimum, and maximum of μ and M of all the correctly-tracked songs are listed in Tables 7 and 8. The deviation error was at most 0.339 and those mean values were relatively small on the average.

Table 7
Mean, minimum and maximum of μ at the quarter-note, half-note and measure levels

Rhythmic level	mean	min	max
Quarter-note level	0.062	0.019	0.130
Half-note level	0.031	0.009	0.069
Measure level	0.015	0.005	0.036

Table 8
Mean, minimum and maximum of M at the quarter-note, half-note and measure levels

Rhythmic level	mean	min	max
Quarter-note level	0.223	0.053	0.339
Half-note level	0.101	0.026	0.174
Measure level	0.045	0.013	0.077

5.3. Measuring rhythmic difficulty

It is important but very difficult to measure the rhythmic difficulty of a real-world song from the viewpoint of the input of beat tracking, because the difficulty can be influenced from various aspects of songs. We tried, as a first step, to evaluate the power transition of the input audio signals. In terms of the power transition of the audio signals, 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.

We therefore consider differences between the power on beats and the power on other positions as a measure of the rhythmic difficulty. This measure, called the *power-difference measure*, is calculated as follows:

(1) *Finding the local maximum of the power*. Let $\text{power}_{\text{input}}(t)$ be the power of the input audio signal at time t.[14] We first calculate two kinds of local maximum of the power, $\text{pow}_{\text{beat}}(n)$ representing the maximum power on the nth beat and $\text{pow}_{\text{other}}(n)$ representing the maximum power on positions between the nth beat and $(n+1)$th beat:

$$\text{pow}_{\text{beat}}(n) = \max_{t\,=\,T_{\text{ans}}(n)-\text{PwMargin}}^{T_{\text{ans}}(n)+I_{\text{ans}}(n)/4-\text{PwMargin}} (\text{power}_{\text{input}}(t)), \tag{26}$$

$$\text{pow}_{\text{other}}(n) = \max_{t\,=\,T_{\text{ans}}(n)+I_{\text{ans}}(n)/4-\text{PwMargin}}^{T_{\text{ans}}(n+1)-\text{PwMargin}} (\text{power}_{\text{input}}(t)), \tag{27}$$

where $T_{\text{ans}}(n)$ is the nth correct beat time of the quarter-note level and $I_{\text{ans}}(n)$ is the nth inter-beat interval (Fig. 12). PwMargin is a margin to obtain appropriate maximum values because the power of sounds played on the beat sometimes increases just before its beat time. In our current implementation, PwMargin is equal to 2 (23.22 ms).

(2) *Calculating the normalized power difference of each beat*. The normalized power difference $\text{diff}_{\text{pow}}(n)$ between $\text{pow}_{\text{other}}(n)$ and $\text{pow}_{\text{beat}}(n)$ is then calculated:

$$\text{diff}_{\text{pow}}(n) = 0.5\,\frac{\text{pow}_{\text{other}}(n) - \text{pow}_{\text{beat}}(n)}{\max(\text{pow}_{\text{other}}(n), \text{pow}_{\text{beat}}(n))} + 0.5. \tag{28}$$

(3) *Calculating the mean of the normalized power differences in a song*. We finally calculate the power-difference measure $\text{AveDiff}_{\text{pow}}$, which is the mean of all the $\text{diff}_{\text{pow}}(n)$ in the song:

$$\text{AveDiff}_{\text{pow}} = \frac{1}{N-1}\sum_{n=1}^{N-1} \text{diff}_{\text{pow}}(n), \tag{29}$$

[14] The time resolution of this measurement in our current implementation is 11.61 ms (= 256/22050 Hz).

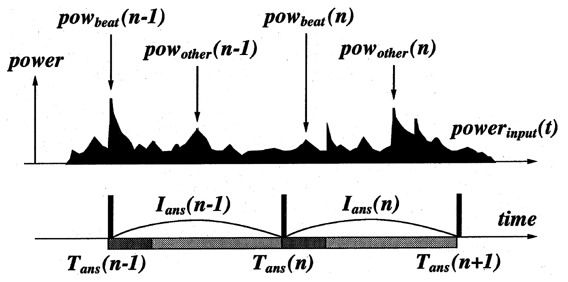


Fig. 12. Finding the local maximum of the power.

where N is the number of the correct beat times. The $\mathrm{AveDiff}_{\mathrm{pow}}$ 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. Practically speaking, the $\mathrm{AveDiff}_{\mathrm{pow}}$ of a song cannot be 1.

Using this power-difference measure, we evaluated the rhythmic difficulty of the 40 songs that we utilized for testing the beat-tracking system. Fig. 13 shows the histogram of $\mathrm{AveDiff}_{\mathrm{pow}}$ for all the songs. Although this measure is not perfect for evaluating the rhythmic difficulty, it represents a meaningful step on the road to measuring the beat-tracking difficulty in an objective way.

5.4. Evaluating the effectiveness of musical decisions

We tested the effectiveness of the musical decisions proposed in Section 3.2 in order to confirm that those decisions based on chord-change possibilities contributed to the performance improvement in determining each level of the hierarchical beat structure. In this experiment, we disabled the system function of making musical decisions based on chord changes by setting 0 to the corresponding chord-change possibility ($C_{\mathrm{Q}}(n)$ and/or $C_{\mathrm{E}}(n)$). The disabled system was tested on the same 40 songs utilized in the evaluation described in Section 5.2.

The results are listed in Table 9. In this time, in order to facilitate the performance comparison within the table, the percentages in parentheses indicate the ratio of the number of the correctly tracked songs to 40 (the number of all the songs). This result shows that musical decision based on eighth-note-level knowledge contributed the quarter-note-level performance and that musical decision based on quarter-note-level knowledge contributed the half-note-level and measure-level performances. This also shows that both of the chord-change possibilities are necessary for determining the hierarchical beat structure.

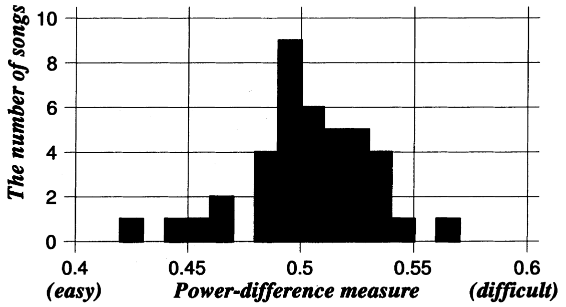


Fig. 13. Histogram of the power-difference measures for all the songs.

Table 9
Results of evaluating musical decision effectiveness

	Condition (Enable: o, Disable: ×)			
Quarter-note chord-change possibility	×	×	o	o
Eighth-note chord-change possibility	×	o	×	o
Measure level	0 songs (0.0%)	0 songs (0.0%)	18 songs (45.0%)	32 songs (80.0%)
Half-note level	0 songs (0.0%)	0 songs (0.0%)	20 songs (50.0%)	34 songs (85.0%)
Quarter-note level	14 songs (35.0%)	29 songs (72.5%)	20 songs (50.0%)	35 songs (87.5%)

5.5. Summary and beat-tracking application

The above-mentioned results show that the beat detection rates obtained in our evaluation with real-world audio signals were more than 87.5 percent and that the method of detecting chord changes and basing musical decisions on chord changes were effective enough to contribute to determining the hierarchical beat structure comprising the three rhythmic levels.

We have also developed an application that displays real-time computer graphics dancers whose motions change in time to musical beats (Fig. 14). This application has shown that our system is useful in multimedia applications in which human-like hearing ability is desirable.

6. Conclusion

We have described the main beat-tracking problem in dealing with drumless audio signals, a solution to that problem, and the configuration and implementation of a real-time beat-tracking system. The experimental results with the proposed tracking-accuracy measurement set show that the system detected, in audio signals sampled from compact discs of popular music, the hierarchical beat structure comprising the quarter-note, half-note, and measure levels at the accuracy more than 87.5%.

We proposed a method for detecting chord changes by analyzing the frequency spectrum sliced at provisional beat times (top-down information). We think that such an approach, without chord name identification, is meaningful because a person generally does not perceive music as musical symbols. This

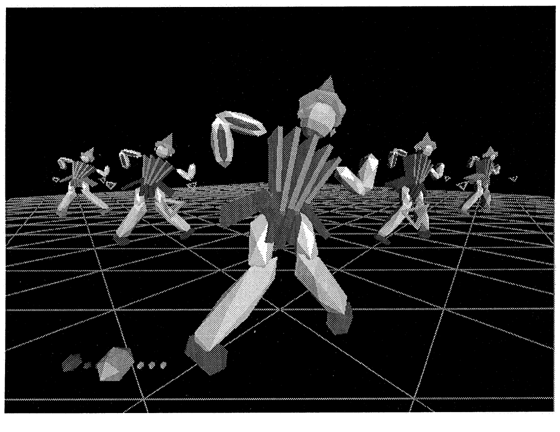


Fig. 14. Virtual dancers synchronized with musical beats.

method enabled our system to determine the beat types in audio signals without drum-sounds and to select the appropriate hypothesis from multiple agent-generated hypotheses.

We plan to upgrade the system by generalizing it to other musical genres and enabling it to follow tempo changes. Future work will include a study of making use of other higher-level musical structure and will include the application to various multimedia systems for which beat tracking is useful, such as systems for video/audio editing, controlling stage lighting, and synchronizing various computer graphics with music.

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.

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.

Desain, P., 1992. Can computer music benefit from cognitive models of rhythm perception?. In: Proceedings of the 1992 International Computer Music Conference. pp. 42–45.

Desain, P., Honing, H., 1989. The quantization of musical time: A connectionist approach. Computer Music J. 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.

Desain, P., Honing, H., 1995. Computational models of beat induction: the rule-based approach. Working Notes of the IJCAI-95 Workshop on Artificial Intelligence and Music. pp. 1–10.

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.

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.

Goto, M., Muraoka, Y., 1995a. Music understanding at the beat level – real-time beat tracking for audio signals –. Working Notes of the IJCAI-95 Workshop on Computational Auditory Scene Analysis. pp. 68–75.

Goto, M., Muraoka, Y., 1995b. A real-time beat tracking system for audio signals. In: Proceedings of the 1995 International Computer Music Conference. pp. 171–174.

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.

Goto, M., Muraoka, Y., 1997. Issues in evaluating beat tracking systems. Working Notes of the IJCAI-97 Workshop on Issues in AI and Music. pp. 9–16.

Ishihata, H., et al., 1991. An architecture of highly parallel computer AP1000. In: IEEE Pacific Rim Conference on Communications, Computers, Signal Processing. pp. 13–16.

Katayose, H., et al., 1989. An approach to an artificial music expert. In: Proceedings of the 1989 International Computer Music Conference. pp. 139–146.

Large, E.W., 1995. Beat tracking with a nonlinear oscillator. Working Notes of the IJCAI-95 Workshop on Artificial Intelligence and Music. pp. 24–31.

Lee, C., 1985. The rhythmic interpretation of simple musical sequences: Towards a perceptual model. In: P. Howell, I. Cross, R. West (Eds.), Musical Structure and Cognition. Academic Press, New York, pp. 53–69.

Rosenthal, D., 1992a. Emulation of human rhythm perception. Computer Music J. 16 (1), 64–76.

Rosenthal, D., 1992b. Machine rhythm: Computer emulation of human rhythm perception. Massachusetts Institute of Technology, MA, Ph.D. thesis.

Rowe, R., 1993. Interactive Music Systems. The MIT Press, Cambridge, MA.

Scheirer, E.D., 1996. Using bandpass and comb filters to beat-track digital audio. Unpublished.

Schloss, W.A., 1985. On the automatic transcription of percussive music – From acoustic signal to high-level analysis. CCRMA, Stanford University, Ph.D. thesis.

Smith, L.M., 1996. Modelling rhythm perception by continuous time-frequency analysis. In: Proceedings of the 1996 International Computer Music Conference. pp. 392–395.

Todd, N.P.M., 1994. The auditory 'primal sketch': A multiscale model of rhythmic grouping. J. New Music Res. 23 (1), 25–70.

Todd, N.P.M., 1995. The kinematics of musical expression. J. Acoust. Soc. Amer. 97 (3), 1940–1949.

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.