Music Understanding At The Beat Level — Real-time Beat Tracking For Audio Signals —

Masataka Goto and **Yoichi Muraoka**
School of Science and Engineering, Waseda University
3-4-1 Ohkubo Shinjuku-ku, Tokyo 169, JAPAN.
{goto, muraoka}@muraoka.info.waseda.ac.jp

Abstract

This paper presents the main issues and our solutions to the problem of understanding musical audio signals at the beat level, issues which are common to more general auditory scene analysis. Previous beat tracking systems were not able to work in realistic acoustic environments. We built a real-time beat tracking system that processes audio signals that contain sounds of various instruments. The main features of our solutions are: (1) To handle ambiguous situations, our system manages multiple agents that maintain multiple hypotheses of beats. (2) Our system makes a context-dependent decision by leveraging musical knowledge represented as drum patterns. (3) All processes are performed based on how reliable detected events and hypotheses are, since it is impossible to handle realistic complex signals without mistakes. (4) Frequency-analysis parameters are dynamically adjusted by interaction between low-level and high-level processing. In our experiment using music on commercially distributed compact discs, our system correctly tracked beats in 40 out of 42 popular songs in which drums maintain the beat.

1 Introduction

Our goal is to build a system that can understand musical audio signals in a human-like fashion. We believe that an important initial step is to build a system which, even in its preliminary implementation, can deal with realistic audio signals, such as ones sampled from commercially distributed compact discs. Therefore our approach is first to build such a robust system which can understand music at a low level, and then to upgrade it to understand music at a higher level.

Beat tracking is an appropriate initial step in computer understanding of Western music, because beats are fundamental to its perception. Even if a person cannot completely segregate and identify every sound component, he can nevertheless track musical beats and keep time to music by hand-clapping or foot-tapping. It is almost impossible to understand music without perceiving beats, since the beat is a fundamental unit of the temporal structure of music. We therefore first build a computational model of beat perception and then extend the model, just as a person recognizes higher-level musical events on the basis of beats.

Following these points of view, we build a beat tracking system, called *BTS*, which processes realistic audio signals and recognizes temporal positions of beats in real time. BTS processes monaural signals that contain sounds of various instruments and deals with popular music, particularly rock and pop music in which drums maintain the beat. Not only does BTS predict the temporal position of the next beat (quarter-note); it also determines whether the beat is strong or weak[1]. In other words, our system can track beats at the half-note level.

To track beats in audio signals, the main issues relevant to auditory scene analysis are: (1) In the interpretation of audio signals, various ambiguous situations arise. Multiple interpretations of beats are possible at any given point, since there is not necessarily a single specific sound that directly indicates the beat position. (2) Decisions in choosing the best interpretation are context-dependent. Musical knowledge is necessary to take a global view of the tracking process. (3) It is almost impossible to detect all events in complex audio signals correctly and completely. Moreover any interpretation of detected events may include mistakes. (4) The optimal set of frequency-analysis parameters depends on the input. It is desirable to adjust those parameters based on a kind of global context.

Our beat tracking system addresses the issues presented above. To handle amgibuous situations, BTS examines multiple hypotheses maintained by multiple agents that track beats according to different strategies. Each agent makes a context-dependent decision by matching pre-registered drum patterns with the currently detected drum pattern. BTS also estimates how reliable detected events and hypotheses are, since they may include both correct and incorrect interpretations. To adjust frequency-analysis parameters dynamically, BTS supports interaction between onset-time finders in the low-level frequency analysis and the higher-level agents that interpret these onset times and predict beats.

To perform this computationally intensive task in real time, BTS has been implemented on a parallel computer, the Fu-

[1] 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 quarter note.

jitsu AP1000. In our experiment with 8 pre-registered drum patterns, BTS correctly tracked beats in 40 out of 42 popular songs sampled from compact discs. This result shows that our beat-tracking model based on multiple-agent architecture is robust enough to handle real-world audio signals.

2 Acoustic Beat-Tracking Issues

The following are the main issues related to tracking beats in audio signals, and they are issues which are common to more general computational auditory frameworks that include speech, music, and other environmental sounds.

2.1 Ambiguity of interpretation

In the interpretation of audio signals, various ambiguous situations arise. At any given point in the analysis, multiple interpretations may appear possible; only later information can determine the correct interpretation. In the case of beat tracking, the position of a beat depends on events that come after it. There are several ambiguous situations, such as ones where several events obtained by frequency analysis may correspond to a beat, and different inter-beat intervals[2]seem to be plausible.

2.2 Context-dependent decision

Decisions in choosing the best interpretation are context-dependent. To decide which interpretation in an ambiguous situation is best, global understanding of the context or situation is desirable. A low-level analysis, such as frequency analysis, cannot by itself provide enough information on this global context. Only higher-level processing using domain knowledge makes it possible to make an appropriate decision. In the case of beat tracking, musical knowledge is needed to determine whether a beat is strong or weak and which note-value it corresponds to.

2.3 Imprecision in event detection

It is almost impossible to detect all events in complex audio signals correctly. In frequency analysis, detected events will generally include both correct and incorrect interpretations. A system dealing with realistic audio should have the ability to decide which events are reliable and useful. Moreover, when the system interprets those events, it is necessary to consider how reliable interpretations and decisions are, since they may include mistakes.

2.4 Adjustment of frequency-analysis parameters

The optimal set of frequency-analysis parameters depends on the input. It is generally difficult, in a sound understanding system, to determine a set of parameters appropriate to all possible inputs. It is therefore desirable to adjust these parameters based on the global context which, in turn, is estimated from the previous events provided by the frequency analysis. In the case of beat tracking, appropriate sets of parameters depend on characteristics of the input song, such as its tempo and the number of instruments used in the song.

3 Our Approach

Our beat tracking system addresses the general issues discussed in the last section. The following are our main solutions to them.

3.1 Multiple hypotheses maintained by multiple agents

Our way of managing the first issue (ambiguity of interpretation) is to maintain multiple hypotheses, each of which corresponds to a provisional or hypothetical interpretation of the input [Rosenthal *et al.*, 1994; Rosenthal, 1992; Allen and Dannenberg, 1990]. A real-time system using only a single hypothesis is subject to garden-path errors. A multiple hypotheses system can pursue several paths simultaneously, and decide at later time which one was correct.

BTS is based on multiple-agent architecture in which multiple hypotheses are maintained by programmatic agents which use different strategies for beat-tracking (Figure 1 shows the processing model of BTS). Because the input signals are examined according to the various viewpoints with which these agents interpret the input, various hypotheses can emerge. For example, agents that pay attention to different frequency ranges may predict different beat positions.

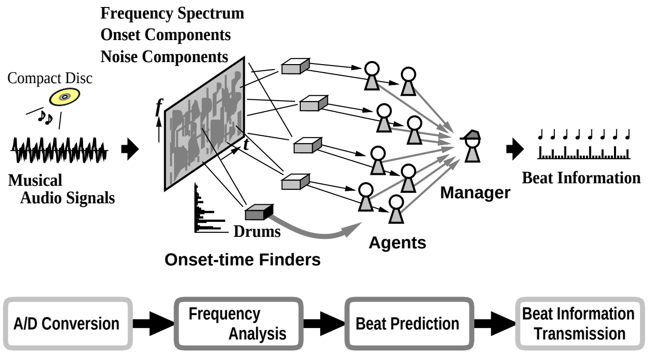

Figure 1: Processing model

The multiple-agent architecture enables BTS to survive difficult beat-tracking situations. Even if some agents lose track of beats, BTS will correctly track beats as long as other agents keep the correct hypothesis. Each agent interprets notes' onset times obtained by frequency analysis, makes a hypothesis, and evaluates its own reliability. The output of the system is then determined on the basis of the most reliable agent.

3.2 Musical knowledge for understanding context

To handle the second issue (context-dependent decision), BTS leverages musical knowledge represented as pre-registered drum patterns. In our current implementation, BTS deals with popular music in which drums maintain the beat. Drum patterns are therefore a suitable source of musical knowledge. A typical example is a pattern where a bass drum and a snare drum sound on the strong and weak beats, respectively; this pattern is an item of domain knowledge on how drum-sounds are frequently used in a large class of popular music. Each agent matches such pre-registered patterns with the currently detected drum pattern; the result provides a more global view

[2] The inter-beat interval is the temporal difference between two successive beats.

of the tracking process. These results enable BTS to determine whether a beat is strong or weak and which inter-beat interval corresponds to a quarter note.

Although pre-registered drum patterns are effective enough to track beats at the half-note level in the case of popular music that includes drums, we feel that they are inadequate as a representation of general musical knowledge. Higher level knowledge is therefore necessary to deal with other musical genres and to understand music at a higher level in future implementations.

3.3 Reliability-based processing

Our way of addressing the third issue (imprecision in event detection) is to estimate reliability of every event and hypothesis. The higher the reliability, the greater its importance in all processing in BTS. The method used for estimating the reliability depends on how the event or hypothesis is obtained. For example, the reliability of an onset time is estimated by a process that takes into account such factors as the rapidity of increase in power, and the power present in nearby time-frequency regions. The reliability of a hypothesis is determined on the basis of how its past-predicted beats coincide with the current onset times obtained by frequency analysis.

3.4 Interaction between low level and high level processing

To manage the fourth issue (adjustment of frequency-analysis parameters), BTS supports interaction between onset-time finders in the low-level frequency analysis and the agents that interpret the results of those finders at a higher level. IPUS [Nawab and Lesser, 1992] also addresses the same issue by structuring the bi-directional interaction between front-end signal processing and signal understanding processes. This interaction enables the system to dynamically adjust parameters so as to fit the current input signals. We implement a simpler scheme – i.e., BTS does not have the sophisticated discrepancy-diagnosis mechanism implemented in IPUS.

BTS employs multiple onset-time finders that have different analytical points of view and are tuned to provide different results. For example, some finders may detect onset times in different frequency ranges, and some may detect with different levels of sensitivity (Figure 1). Each of these finders communicates with two agents called an *agent-pair*. Each agent-pair receives onset times from the corresponding finder, and can, in turn, re-adjust the parameters of the finder based on the reliability estimate of the hypotheses maintained by its agents. If the reliability of a hypothesis remains low for a long time, the agent tunes the corresponding onset-time finder so that parameters of the finder are close to these of the most reliable finder-agent pair. In other words, there is feedback between the (high-level) beat-prediction agents and the (low-level) onset-time finders.

4 System Description

Figure 2 shows the overview of our beat tracking system. BTS assumes that the time-signature of an input song is 4/4, and its tempo is constrained to be between 65 M.M.[3] and 185 M.M. and almost constant; these assumptions fit a large class of popular music. The emphasis in our system is on finding the temporal positions of quarter notes in audio signals rather than on tracking tempo changes; in the repertoire with which we are concerned, tempo variation is not a major factor. In our current implementation, BTS can only deal with music in which drums maintain the beat. BTS transmits *beat information* (*BI*) that is the result of tracking beats to other applications in time to the input music. BI consists of the temporal position of a beat (*beat time*), whether the beat is strong or weak (*beat type*), and the current tempo.

The two main stages of processing are *Frequency Analysis*, in which a variety of cues are detected, and *Beat Prediction*, in which multiple hypotheses of beat positions are examined in parallel (Figure 2). In the *Frequency Analysis* stage, BTS detects events such as onset times in several different frequency ranges, and onset times of two different kinds of drum-sounds: a bass drum (*BD*) and a snare drum (*SD*). In the *Beat Prediction* stage, BTS manages multiple agents that interpret these onset times according to different strategies and make parallel hypotheses. Each agent first calculates the inter-beat interval; it then predicts the next beat time, and infers its beat type, and finally evaluates the reliability of its own hypothesis. BI is then generated on the basis of the most reliable hypothesis. Finally, in the *BI Transmission* stage, BTS transmits BI to other application programs via a computer network.

The following describe the main stages of Frequency Analysis and Beat Prediction.

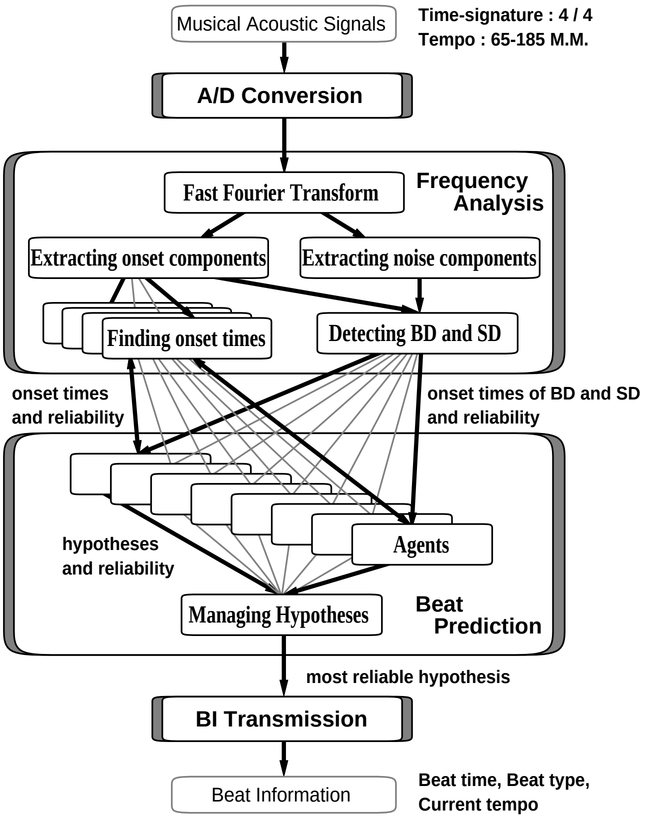

Figure 2: Overview of our beat tracking system

[3] the number of quarter notes per minute

4.1 Frequency Analysis

Multiple onset-time finders detect multiple tracking cues. First, onset components are extracted from the frequency spectrum calculated by the Fast Fourier Transform. Second, onset-time finders detect onset times in different frequency ranges and with different sensitivity levels. In addition, another drum-sound finder detects onset times of drum-sounds by acquiring the characteristic frequency of the bass drum (BD) and extracting noise components for the snare drum (SD). These results are sent to agents in the Beat Prediction stage.

Fast Fourier Transform (FFT)

The frequency spectrum (the power spectrum) is calculated with the FFT using the Hanning window. Each time the FFT is applied to the digitized audio signal, the window is shifted to the next frame. In our current implementation, the input signal is digitized at 16bit/22.05kHz, the size of the FFT window is 1024 samples (46.44msec), and the window is shifted by 256 samples (11.61msec). The frequency resolution is consequently 21.53Hz and, the time resolution is 11.61msec.

Extracting onset components

Frequency components whose power has been rapidly increasing are extracted as onset components. The onset components and their degree of onset (rapidity of increase in power) are obtained from the frequency spectrum. The frequency component $p(t,f)$ that fulfills the conditions in (1) is regarded as the onset component (Figure 3).

$$\begin{cases} p(t,f) & > & pp \\ np & > & pp \end{cases} \tag{1}$$

Where $p(t,f)$ is the power of the spectrum of frequency f at time t, pp and np are given by:

$$pp = \max(p(t-1,f), p(t-1,f\pm 1), p(t-2,f)) \tag{2}$$

$$np = \min(p(t+1,f), p(t+1,f\pm 1)) \tag{3}$$

If $p(t,f)$ is an onset component, its degree of onset $d(t,f)$ is given by:

$$d(t,f) = \max(p(t,f), p(t+1,f)) - pp \tag{4}$$

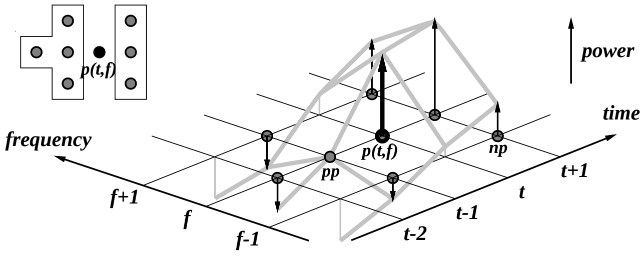

Figure 3: Extracting an onset component

Finding onset times

Multiple onset-time finders[4] use different sets of frequency-analysis parameters. Each finder corresponds to an agent-pair and sends its onset information to the two agents that form the agent-pair (Figure 1, Figure 6).

Each onset time and its reliability are obtained as follows: The onset time is given by the peak time found by peak-picking in $D(t)$ along the time axis, where $D(t)$, the sum of the degree of onset, is defined as:

$$D(t) = \sum_f d(t,f) \tag{5}$$

$D(t)$ is linearly smoothed with a convolution kernel before its peak time and peak value are calculated. The reliability of the onset time is obtained as the ratio of its peak value to the recent local-maximal peak value.

Each finder has two parameters: The first parameter, *sensitivity*, is the size of the convolution kernel used for smoothing. The smaller the size of the convolution kernel, the higher its sensitivity. The second parameter, *frequency range*, is the range of frequency for the summation of $D(t)$ (in Equation (5)). Limiting the range makes it possible to find onset times in several different frequency ranges. The settings of these parameters vary from finder to finder.

Extracting noise components

BTS extracts noise components as a preliminary step to detecting SD. Because non-noise sounds typically have harmonic structures and peak components along the frequency axis, frequency components whose power is roughly uniform locally are extracted and considered to be potential SD sounds.

The frequency component $p(t,f)$ that fulfills the conditions in (6) is regarded as a potential SD component $n(t,f)$ (Figure 4).

$$\begin{cases} hp & > & p(t,f)/2 \\ lp & > & p(t,f)/2 \end{cases} \tag{6}$$

$$hp = (p(t\pm 1, f+1) + p(t,f+1) + p(t,f+2))/4 \tag{7}$$

$$lp = (p(t\pm 1, f-1) + p(t,f-1) + p(t,f-2))/4 \tag{8}$$

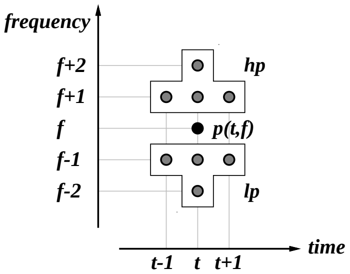

Figure 4: Extracting a noise component

Detecting BD and SD

The bass drum (BD) is detected from the onset components and the snare drum (SD) is detected from the noise components. These results are sent to all agents in the Beat Prediction stage.

[Detecting onset times of BD]

Because the sound of BD is not known in advance, BTS learns the characteristic frequency of BD that depends on

[4] In the current BTS, the number of onset-time finders is 15.

the current song by examining the extracted onset components. For times at which onset components are found, BTS finds peaks along the frequency axis and histograms them (Figure 5). The histogram is weighted by the degree of onset $d(t, f)$. The characteristic frequency of BD is given by the lowest-frequency peak of the histogram.

BTS judges that BD has sounded at times when (1) an onset is detected and (2) the onset's peak frequency coincides with the characteristic frequency of BD. The reliability of the onset times of BD is obtained as the ratio of $d(t, f)$ currently under consideration to the recent local-maximal peak value.

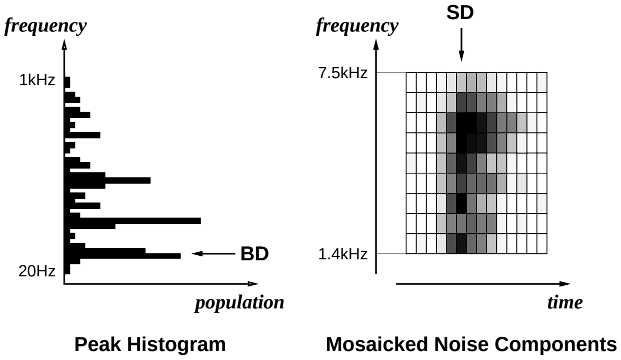

Figure 5: Detecting BD and SD

[Detecting onset times of SD]

Since the sound of SD typically has noise components widely distributed along the frequency axis, BTS needs to detect such components. First, the noise components $n(t, f)$ are mosaicked (Figure 5): the frequency axis of the noise components is divided into sub-bands[5], and the mean of the noise components in each sub-band is calculated.

Second, BTS calculates how widely noise components are distributed along the frequency axis ($c(t)$) in the mosaicked noise components: $c(t)$ is calculated as the product of all mosaicked components within middle-frequency range[6] after they are clipped with a dynamic threshold.

Finally, the onset time of SD and its reliability are obtained by peak-picking of $c(t)$ in the same way as in the onset-time finder.

4.2 Beat Prediction

To track beats in real time, it is necessary to predict future beat times from the onset times obtained previously. By the time the system finishes processing a sound in an acoustic signal, its onset time has already passed.

Multiple agents interpret the results of the Frequency Analysis stage according to different strategies, and maintain their own hypotheses, each of which consists of a predicted next-beat time, its beat type, and the current inter-beat interval (*IBI*) (Figure 6). These hypotheses are gathered by the manager (Figure 1), and the most reliable one is selected as the output.

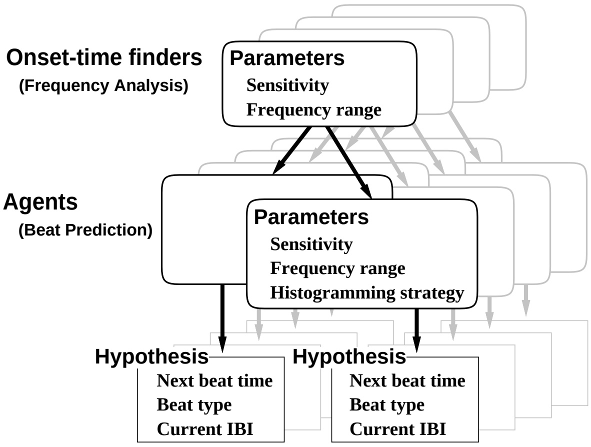

Figure 6: Onset-time finders and agents

All agents[7] are grouped into pairs. Two agents in the same pair use the same IBI, and cooperatively predict the next beat times, the difference of which is half the IBI. This enables one agent to track the correct beats even if the other agent tracks the middle of the two successive correct beats (which covers for one of the typical tracking errors). Each agent-pair is different in that it receives onset information from a different onset-time finder (Figure 6).

Each agent has three parameters that determine its strategy for making the hypothesis. Both agents in an agent-pair have the same setting of these parameters. The settings of these parameters vary from pair to pair. The first two parameters are *sensitivity* and *frequency range*. These two control the corresponding parameters of the onset-time finder, and adjust the quality of the onset information that the agent receives. An agent-pair with high sensitivity tends to have a short IBI and be relatively unstable, and one with low sensitivity tends to have a long IBI and be stable. The third parameter, *histogramming strategy*, takes a value of either *successive* or *alternate*. When the value is *successive*, successive onsets are used in forming the inter-onset interval (*IOI*)[8] histogram; likewise, when the value is *alternate*, alternate values are used.

The following describe the formation and management of hypotheses. First, each agent calculates the IBI and predicts the next beat time, and then evaluates its own reliability (*Predicting next beat*). Second, the agent infers its beat type and modifies its reliability (*Inferring beat type*). Third, an agent whose reliability remains low for a long time changes its own parameters (*Adjusting parameters*). Finally, the most reliable hypothesis is selected from the hypotheses of all agents (*Managing hypotheses*).

Predicting next beat

Each agent predicts the next beat time by adding the current IBI to the previous beat time (Figure 7). The IBI is given by the interval with the maximum value in the inter-onset interval (IOI) histogram that is weighted by the reliability of

[5] In the current BTS, the number of sub-bands is 16.

[6] The current BTS multiplies mosaicked components that are approximately ranged from 1.4kHz to 7.5kHz.

[7] In the current BTS, the number of agents is 30.

[8] The inter-onset interval is the temporal difference between two successive onsets.

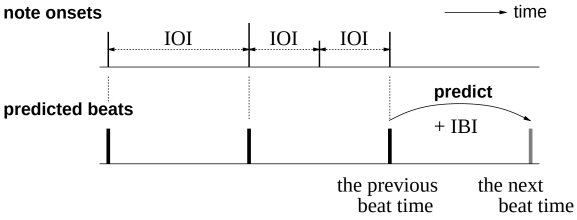

Figure 7: Beat prediction

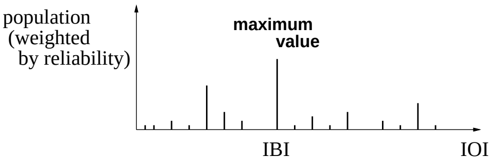

Figure 8: IOI histogram

onset times (Figure 8). In other words, the IBI is calculated as the most frequent interval between onsets that have high reliability. Before the agent adds the IBI to the previous beat time, the previous beat time is adjusted to its nearest onset time if they almost coincide.

Each agent evaluates the reliability of its own hypothesis. This is determined on the basis of how the past-predicted beats coincide with onset times. The reliability is increased if an onset time coincides with the beat time predicted previously. If an onset time coincides with a time that corresponds to the position of an eighth note or a sixteenth note, the reliability is also slightly increased. Otherwise, the reliability is decreased.

Inferring beat type

Our system, like human listeners, utilizes BD and SD as principle clues to the location of strong and weak beats. Note that BTS cannot simply use the detected BD and SD to track the beats, because the drum detection process is too noisy. The detected BD and SD are used only to label each predicted beat time with the beat type (strong or weak).

Each agent determines the beat type by matching the pre-registered drum patterns of BD and SD with the currently detected drum pattern. The beginning of the best-matched pattern indicates the position of the strong beat.

Figure 9 shows two examples of the pre-registered patterns. These patterns represent how BD and SD are typically played in rock and pop music. The beginning of a pattern should be the strong beat, 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.

The beat type and its reliability are obtained as follows: (1) The onset times of drums are formed into the currently detected pattern, with one sixteenth-note resolution that is obtained by interpolating between successive beat times (Figure 10). (2) The *matching score* of each pre-registered pattern is calculated by matching the pattern with the currently detected pattern: The score is weighted by the product of the weight in the pre-registered pattern and the reliability of the detected onset. (3) The beat type is inferred from the position of the strong beat obtained by the best-matched pattern (Figure 11): The reliability of the beat type is obtained from the highest matching score.

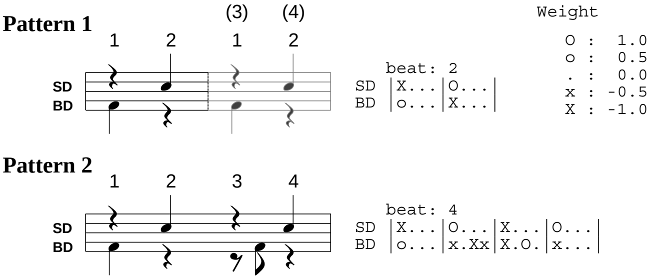

Figure 9: Examples of pre-registered drum patterns

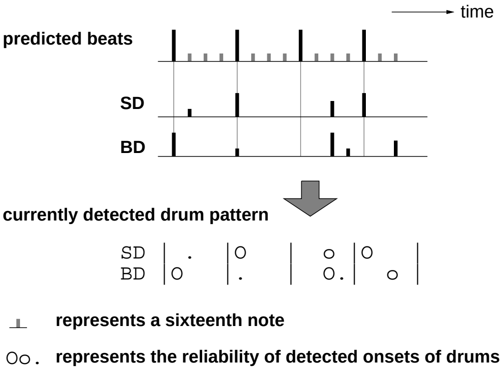
time
predicted beats
SD
BD
currently detected drum pattern
SD | . |O | o |O |
BD |O |. | O.| o |
represents a sixteenth note
Oo. represents the reliability of detected onsets of drums

Figure 10: A drum pattern detected from an input

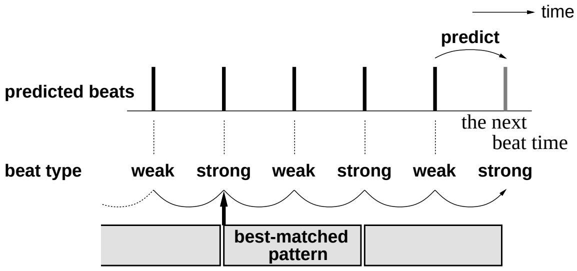

Figure 11: Inferring beat type

The reliability of each hypothesis is modified on the basis of the reliability of its beat type. If the reliability of the beat type is high, the IBI in the hypothesis can be considered to correspond to a quarter note. In that case, the reliability of the hypothesis is increased so that a hypothesis with an IBI corresponding to a quarter note is likely to be selected.

Adjusting parameters

When the reliability of a hypothesis remains low for a long time, the agent suspects that its parameter set is not suitable for the current input. In that case, the agent adjusts its parameters cooperatively, i.e., considering the states of other agents.

The adjustment is made as follows: (1) If the reliability remains low for a long time, the agent requests permission from the manager to change the parameters. (2) If the reliability of the other agent in the same agent-pair is not low, the manager refuses to let the agent change its parameters. (3) The manager permits the agent to change if it has the lowest sum of the reliability in its agent-pair. The manager then inhibits other agents from changing for a certain period. (4) The agent, having received permission, selects a new set of the three parameters that determine its strategy. If we think of the three parameters forming a three-dimensional parameter space, the agent selects a point that is not occupied by other agents and is close to the point corresponding to the parameters of the most reliable agent. The parameter change then affects the corresponding onset-time finder.

Managing hypotheses

The manager classifies all agent-generated hypotheses into groups, according to beat time and IBI. Each group has an overall reliability, given by the sum of the reliability of the group's hypotheses. The most reliable hypothesis in the most reliable group is selected as the output and sent to the BI Transmission stage.

The beat type in the output is updated only using the beat type that has the high reliability. When the reliability of a beat type is low, its beat type is determined from the previous reliable beat type based on the alternation of strong and weak beats. This enables BTS to disregard an incorrect beat type that is caused by some local irregularity of rhythm.

5 Implementation

To perform a computationally-intensive task such as processing and understanding complex audio signals in real time, parallel processing provides a practical and realizable solution. BTS has been implemented on a distributed-memory parallel computer, the Fujitsu AP1000 that consists of 64 cells[9][Ishihata *et al.*, 1991]. We apply four kinds of parallelizing techniques to simultaneously execute the heterogeneous processes described in the last section [Goto and Muraoka, 1995].

6 Experiments and Results

We tested BTS on 42 popular songs in the rock and pop music genres. The input was a monaural audio signal sampled from a commercial compact disc, in which drums maintained the beats. Their tempi ranged from 78 M.M. to 184 M.M. and were almost constant.

In our experiment with 8 pre-registered drum patterns, BTS correctly tracked beats in 40 out of 42 songs in real time. At the beginning of each song, beat type was not correctly determined even if the beat time was obtained. This is because BTS had not yet acquired the characteristic frequency of BD. After the BD and SD had sounded stably for a few measures, the beat type was obtained correctly.

We discuss the reason why BTS made mistakes in two of the songs. In both of them, BTS tracked only the weak beat, in other words, the output IBI was double the correct IBI. In one song, the number of agents that held the incorrect IBI was greater than that for the correct one. Since the characteristic frequency of BD was not acquired correctly, drum patterns were not correctly matched and the hypothesis with the correct IBI was not selected. In the other song, there was no agent that held the correct IBI. The peak corresponding to the correct IBI in the IOI histogram was not the maximum peak, since onset times on strong beats were often not detected, and an agent was therefore liable to histogram the interval between SDs.

These results show that BTS can deal with realistic musical signals. Moreover, we have developed an application with BTS that displays a computer graphics dancer whose motion changes with musical beats in real time [Goto and Muraoka, 1994]. This application has shown that our system is also useful in various multimedia applications in which human-like hearing ability is desirable.

7 Discussion

Various beat-tracking related systems have been undertaken in recent years. Most beat tracking systems have great difficulty to work in realistic acoustic environments, however. Most of these systems [Dannenberg and Mont-Reynaud, 1987; Desain and Honing, 1989; Allen and Dannenberg, 1990; Rosenthal, 1992] have dealt with MIDI as their input. Since it is almost impossible to obtain complete MIDI-like representations of audio signals that include various sounds, MIDI-based systems cannot immediately be applied to complex audio signals. Although some systems [Schloss, 1985; Katayose *et al.*, 1989] dealt with audio signals, they were not able to process music played on ensembles of a variety of instruments, especially drums, and did not work in real time.

Our strategy of first building a system that works in realistic complex environments, and then upgrading the ability of the system, is related to the scaling up problem [Kitano, 1993] in the domain of artificial intelligence (Figure 12). As Hiroaki Kitano stated:

> experiences in expert systems, machine translation systems, and other knowledge-based systems indicate that scaling up is extremely difficult for many of the prototypes. [Kitano, 1993]

In other words, it is hard to scale up the system whose preliminary implementation works in not real environments but only laboratory environments. We can expect that computational auditory scene analysis would have similar scaling up problems. We believe that our strategy addresses this issue.

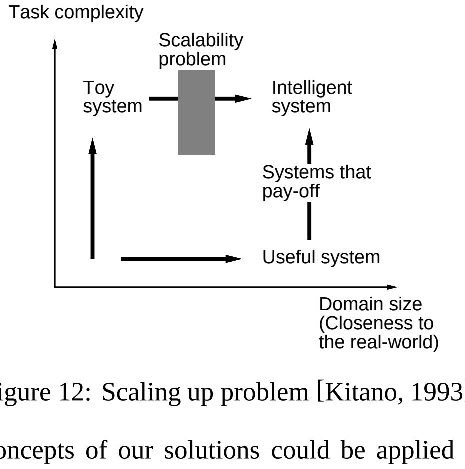

Figure 12: Scaling up problem [Kitano, 1993]

The concepts of our solutions could be applied to other perceptual problems, such as more general auditory scene

[9] A *cell* means a processing element, which has a 25MHz SPARC with an FPU and 16Mbytes DRAM.

analysis and vision understanding. The concept of multiple hypotheses maintained by multiple agents is one possible solution in dealing with ambiguous situations in real time. Context-dependent decision making using domain knowledge is necessary for all higher-level processing in perceptual problems. We think reliability-based processing is essential, not only to various processing dealing with realistic complex signals, but to hypothetical processing of interpretations or symbols. As Nawab and Lesser [1992] describe, the mechanism of bi-directional interaction between low-level signal processing and higher-level interpretation has the advantage of adjusting parameter values of the system dynamically to fit a current situation. We plan to apply our solutions to other real-world perceptual domains.

Our beat-tracking model is based on multiple-agent architecture (Figure 1) where multiple agents with different strategies interact through competition and cooperation to examine multiple hypotheses in parallel. Although several concepts of the term *agents* have been proposed [Minsky, 1986; Maes, 1990; Nakatani *et al.*, 1994], in our terminology, the term *agent* means a software component that satisfies the following requirements:

- the agent has ability to evaluate its own behavior (in our case, hypotheses of beats) on the basis of a situation of real-world input (in our case, the input song).
- the agent cooperates with other agents to perform a given task (in our case, beat tracking).
- the agent adapts to the real-world input by dynamically adjusting its own behavior (in our case, parameters).

8 Conclusion

We have described the main acoustic beat-tracking issues and solutions implemented on our real-time beat tracking system (*BTS*). BTS tracks beats in audio signals that contain sounds of various instruments that include drums, and reports beat information corresponding to quarter notes in time to input music. The experimental results show that BTS can track beats in complex audio signals sampled from compact discs of popular music.

BTS manages multiple agents that track beats according to different strategies in order to examine multiple hypotheses in parallel. This enables BTS to follow beats without losing track of them, even if some hypotheses become incorrect. The use of drum patterns pre-registered as musical knowledge makes it possible to determine whether a beat is strong or weak and which note-value a beat corresponds to.

We plan to upgrade our beat-tracking model to understand music at a higher level and to deal with other musical genres. Future work will include a study on appropriate musical knowledge for dealing with musical audio signals, improvement of interaction among agents and between low-level and high-level processing, and application to other multimedia systems.

Acknowledgments

We thank David Rosenthal and anonymous reviewers for their helpful comments on earlier drafts of this paper. We also thank Fujitsu Laboratories Ltd. for use of the AP1000.

References

[Allen and Dannenberg, 1990] Paul E. Allen and Roger B. Dannenberg. Tracking musical beats in real time. In *Proc. of the 1990 Intl. Computer Music Conf.*, pages 140–143, 1990.

[Dannenberg and Mont-Reynaud, 1987] Roger B. Dannenberg and Bernard Mont-Reynaud. Following an improvisation in real time. In *Proc. of the 1987 Intl. Computer Music Conf.*, pages 241–248, 1987.

[Desain and Honing, 1989] Peter Desain and Henkjan Honing. The quantization of musical time: A connectionist approach. *Computer Music Journal*, 13(3):56–66, 1989.

[Goto and Muraoka, 1994] Masataka Goto and Yoichi Muraoka. A beat tracking system for acoustic signals of music. In *Proc. of the Second ACM Intl. Conf. on Multimedia*, pages 365–372, 1994.

[Goto and Muraoka, 1995] Masataka Goto and Yoichi Muraoka. Parallel implementation of a real-time beat tracking system – real-time musical information processing on AP1000 – (*in Japanese*). In *Proc. of the 1995 Joint Symposium on Parallel Processing*, 1995.

[Ishihata *et al.*, 1991] H. Ishihata, T. Horie, S. Inano, T. Shimizu, and S. Kato. An architecture of highly parallel computer AP1000. In *IEEE Pacific Rim Conf. on Communications, Computers, Signal Processing*, pages 13–16, 1991.

[Katayose *et al.*, 1989] H. Katayose, H. Kato, M. Imai, and S. Inokuchi. An approach to an artificial music expert. In *Proc. of the 1989 Intl. Computer Music Conf.*, pages 139–146, 1989.

[Kitano, 1993] Hiroaki Kitano. Challenges of massive parallelism. In *Proc. of IJCAI-93*, pages 813–834, 1993.

[Maes, 1990] Pattie Maes, editor. *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*. The MIT Press, 1990.

[Minsky, 1986] Marvin Minsky. *The Society of Mind*. Simon & Schuster, Inc., 1986.

[Nakatani *et al.*, 1994] Tomohiro Nakatani, Hiroshi G. Okuno, and Takeshi Kawabata. Auditory stream segregation in auditory scene analysis. In *Proc. of AAAI-94*, pages 100–107, 1994.

[Nawab and Lesser, 1992] S. Hamid Nawab and Victor Lesser. Integrated processing and understanding of signals. In Alan V. Oppenheim and S. Hamid Nawab, editors, *Symbolic and Knowledge-Based Signal Processing*, pages 251–285. Prentice Hall, 1992.

[Rosenthal *et al.*, 1994] David Rosenthal, Masataka Goto, and Yoichi Muraoka. Rhythm tracking using multiple hypotheses. In *Proc. of the 1994 Intl. Computer Music Conf.*, pages 85–87, 1994.

[Rosenthal, 1992] David Rosenthal. *Machine Rhythm: Computer Emulation of Human Rhythm Perception*. PhD thesis, Massachusetts Institute of Technology, 1992.

[Schloss, 1985] W. Andrew Schloss. *On The Automatic Transcription of Percussive Music – From Acoustic Signal to High-Level Analysis*. PhD thesis, CCRMA, Stanford University, 1985.