A Beat Tracking System for Acoustic Signals of Music

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 a beat tracking system that processes acoustic signals of music and recognizes temporal positions of beats in real time. Musical beat tracking is needed by various multimedia applications such as video editing, audio editing, and stage lighting control. Previous systems were not able to deal with acoustic signals that contained sounds of various instruments, especially drums. They dealt with either MIDI signals or acoustic signals played on a few instruments, and in the latter case, did not work in real time. Our system deals with popular music in which drums maintain the beat. Because our system examines multiple hypotheses in parallel, it can follow beats without losing track of them, even if some hypotheses become wrong. Our system has been implemented on a parallel computer, the Fujitsu AP1000. In our experiment, the system correctly tracked beats in 27 out of 30 commercially distributed popular songs.

1 Introduction

Musical beat tracking systems are important for various multimedia applications. These systems recognize temporal positions of quarter notes, just as people keep time to music by hand-clapping or foot-tapping. Such a system is needed by applications, for example, video editing systems, in which a visual track can be automatically synchronized with an audio track using beat tracking. In particular, it facilitates the editing of music promotion videos such as those on MTV since visual motions are synchronized with beats. In an audio editing system or hard disk recording system, beat tracking makes automatic indexing of music possible. The users of these systems can deal with acoustic signals as a set of beats instead of raw acoustic wave data. In live performances, furthermore, beat tracking is useful in the control of stage lighting by a computer. For instance, various properties of lighting such as color, brightness, direction, and effect can be changed in time to the music.

Previous beat tracking systems [1]−[7] have dealt with MIDI as their input. Their reliance on MIDI, however, limited the input source to electronic instruments, and moreover limited their application. Those systems generally dealt with classical works, in particular piano solo, and emphasized tracking tempo changes. Although some systems [8][9] dealt with acoustic signals, they concentrated on automatic transcription rather than beat tracking, and did not work in real time. They could not process music played on ensembles of a variety of instruments because of the difficulty of segregating the sound of each instrument.

This paper describes a beat tracking system, called *BTS*, which processes monaural acoustic signals and recognizes temporal positions of beats in real time. BTS deals with popular music in which drums maintain the beat. We concentrate on finding the position of beats in acoustic signals rather than on following tempo changes, because popular songs have less tempo change than do classical works. BTS receives acoustic signals sampled from a compact disc, tracks beats corresponding to quarter notes, and transmits their temporal positions to other computers through a network in real time.

The difficulties of tracking the beats in acoustic signals are: (1) Acoustic signals sampled from a commercial compact disc consist of sounds of various kinds of instruments. The onsets of notes are difficult to obtain precisely (unlike the case of MIDI signals, where there is no such problem). (2) The simple technique of peak-finding with a threshold is not useful for tracking beats since there are many energy peaks that are not directly related to beats. (3) A beat may not directly correspond to a real sound; it is a perceptual concept

that a human feels in music. There is not necessarily a specific sound that directly indicates the position of beats. Several interpretations of signals are therefore possible at any given point. (4) In general, it is difficult to determine which note-value a beat corresponds to, and whether a beat is a strong beat or a weak beat.

To address these issues, BTS examines multiple possibilities of positions of beats in parallel. BTS computes the reliability of each onset time so that subsequent processing might disregard low reliability onset times. BTS uses a collection of programmatic agents, each of which interprets these onset times and predicts the next beat according to its own strategy. The position of the next beat is determined on the basis of the most reliable agent. In addition, BTS leverages the fact that for a large class of popular music, a bass drum and a snare drum usually sound on the strong and weak beats, respectively. By assuming that these drum-sounds are present, BTS can detect them, infer the length of a quarter note, and determine whether a beat is strong or weak.

BTS has been implemented on a distributed memory parallel computer, the Fujitsu AP1000. BTS reports *beat information* (*BI*) that consists of: the temporal position of a beat (*beat time*), whether the beat is strong or weak (*beat type*), and the current tempo. BTS correctly tracked beats in 27 out of 30 popular songs sampled from commercially distributed compact discs.

2   Acoustic Beat-Tracking Issues

This section presents the beat-tracking issues and our solutions to them. These issues are classified into three groups: issues of signal processing, issues of real-time processing, issues of musical judgement.

☐ **Issues of signal processing**

 1. There is not necessarily a specific sound that directly indicates the position of beats. In fact, a musical beat may not directly correspond to a real sound; there may even be no signal on a beat.

    BTS tracks beats by managing multiple evidence such as onset times in several different frequency ranges, onset times of two different kinds of drum-sounds (a bass drum and a snare drum), and their loudness. BTS interprets this evidence by using musical knowledge such as the manner in which the two kinds of drums are used in a large class of popular music.

 2. It is generally impossible to obtain precise onset times from acoustic signals that contain sounds of various instruments.

    BTS employs sophisticated means of estimating the onset time in the frequency analysis stage. First, BTS finds multiple interpretations of onset times corresponding to various time-window widths. Second, the reliability of an onset time is computed 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 higher the reliability of an onset time, the greater its importance in subsequent processing.

☐ **Issues of real-time processing**

 3. To track beats in real time, only acoustic signals obtained previously are available.

    BTS should predict the next beat time in advance from the onset times obtained previously. At the time when BTS finishes recognition of a sound in an acoustic signal, its onset time has already passed.

 4. BTS should be able to recover the correct tracking even if the current hypothesis becomes incorrect.

    BTS examines multiple possibilities of positions of beats in parallel. BTS manages multiple agents that track beats according to different strategies. Even if some agents lose track of beats, BTS will correctly track beats as long as other agents have the correct hypotheses. Each agent interprets onset times and makes its own hypothesis of beats. BTS generates beat information on the basis of the most reliable hypothesis.

☐ **Issues of musical judgement**

 5. It is generally difficult to determine whether a beat is a strong beat or a weak beat − i.e., it is difficult to track beats at the half-note level.

    BTS assumes that the time-signature is 4/4, this being the most frequent time-signature in the repertoire we are considering. To infer beat type, BTS also assumes that a bass drum (*BD*) usually sounds on a strong beat (the first or third quarter notes in a measure) and a snare drum (*SD*) on a weak beat (the second or fourth). This does not mean that all BD and SD must sound on the strong and weak beats, respectively, but rather that that arrangement should be the most frequent. These assumptions fit a large class of popular music.

    BTS, like human listeners, utilizes BD and SD as principle clues to the location of strong and weak beats.

Because the sounds of BD and SD are not known in advance, BTS automatically acquires the characteristic frequencies of these sounds during the beat-tracking process. Note that BTS cannot simply use the detected BD and SD to track the beats, because this detection process is too noisy. The detected BD and SD are only used to determine the beat type (strong or weak) of an already detected beat.

6. It is difficult to judge whether the inter-beat-interval (*IBI*)[1] corresponds to the temporal length of a quarter note.

The tempo of an input song is constrained to be between 70 M.M.[2] and 180 M.M. and almost constant; popular songs have less tempo variation than do classical works. Even if IBI is limited in the length of the quarter note corresponding to the supported tempo, cases occur such that BTS cannot judge whether the IBI is the length of a quarter note.

BTS concludes that the current IBI is the length of a quarter note if SD usually alternates with BD on the beats obtained previously. If BTS can detect neither BD nor SD, the stable interval between onsets with high reliability is regarded as the length of a quarter note.

3 Overview of BTS

Figure 1 shows the overview of BTS. First, *Frequency Analysis* finds notes' onset times in an input acoustic signal digitized by *A/D Conversion* and also detects BD and SD. Second, multiple agents in *Beat Prediction* interpret the onset times found previously and make parallel hypotheses: each agent first calculates the IBI (inter-beat-interval); it then predicts the next beat time, and infers its beat type, and finally evaluates its own reliability. *BI Generation* assembles BI (beat information) on the basis of the most reliable hypothesis. Finally, *BI Transmission* transmits the BI to other application programs via a computer network.

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

3.1 Frequency Analysis

The onset times in several different frequency ranges and the onset times of BD and SD are obtained by the following
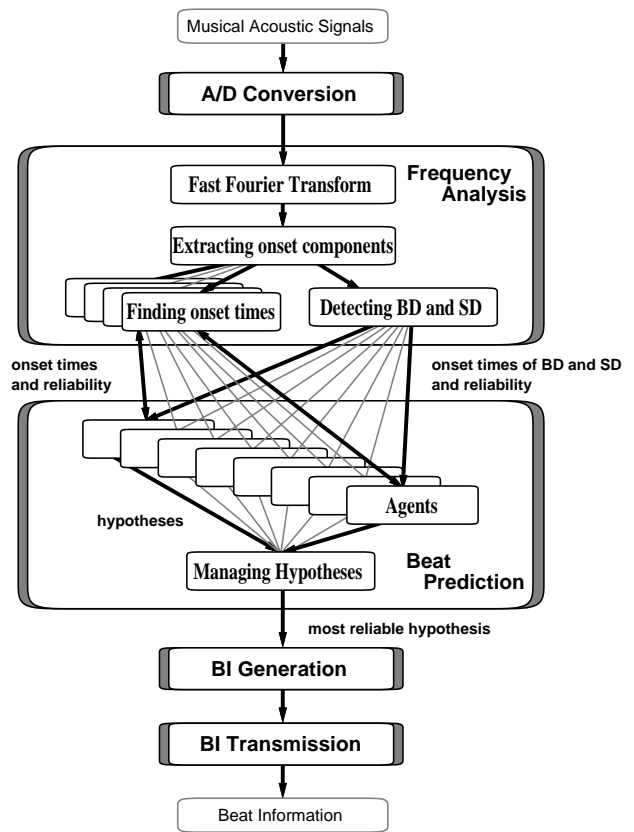
Figure 1: Overview of BTS

processes.

3.1.1 Fast Fourier Transform (FFT)

The frequency spectrum (the power spectrum) is calculated by FFT. Each time FFT is applied to the digitized acoustic signal, the window of FFT is shifted to the next time. The resolution along the frequency axis[3] is determined by the size of the window (WinSIZE). The resolution along the time axis[3] is determined by the shifted size of each window (ShiSIZE).

3.1.2 Extracting onset components

The onset components and their degree of onset (rapidity of increase in power) are extracted from the frequency spectrum. BTS regards the frequency component $p(t, f)$ that fulfills the conditions (1) as the onset component (Figure 2).

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

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

$$pp = \max(p(t-1, f), p(t-1, f \pm 1), p(t-2, f)) \quad (2)$$
$$np = \min(p(t+1, f), p(t+1, f \pm 1)) \quad (3)$$

In other words, these conditions extract frequency components whose power has been increasing.

The degree of onset $d(t, f)$ is given by:

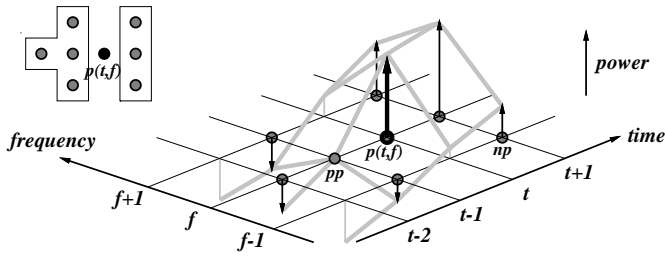$$d(t, f) = p(t, f) - pp + \max(0, p(t+1, f) - p(t, f)) \quad (4)$$

Figure 2: Extracting onset components

3.1.3 Finding onset times

The onset time is found by peak-finding in $D(t)$ along the time axis, where $D(t)$, the sum of the degree of onset, is defined by:

$$D(t) = \sum_f d(t, f) \quad (5)$$

$D(t)$ is linearly smoothed by convolution kernel before the peak time and peak value of $D(t)$ are calculated. The onset time is given by this peak time. The reliability of the onset time is obtained as the ratio of its peak value to a recent maximum peak value.

BTS employs multiple onset-time finders, each of which sends its onset information to a corresponding agent-pair mentioned later in **3.2**. Each finder has two parameters: The first parameter, *sensitivity*, is the size of 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 in the summation of $D(t)$ (in Equation (5)). Limiting the range makes it possible to find the onset times in several different frequency ranges. The settings of these parameters vary from finder to finder.

3.1.4 Detecting BD and SD

BTS learns the characteristic frequencies of BD and SD for the current song by examining the extracted onset components. These characteristic frequencies are then used for subsequent BD and SD detection. For times at which onset components are found, BTS finds peaks along the frequency axis and histograms them (Figure 3). The histogram is weighted by the degree of onset $d(t, f)$. The characteristic frequency of BD is given by the lowest peak of the histogram, and that of SD is given by the maximum peak of the histogram over the frequency of BD.
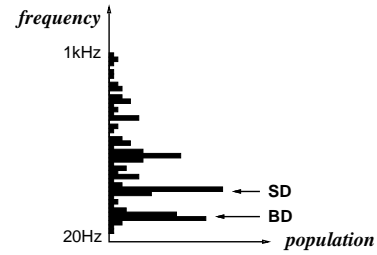

Figure 3: The characteristic frequencies of BD and SD

BTS judges that BD or SD has sounded at times when (1) an onset is detected and (2) the onset's peak frequencies coincide with the characteristic frequencies of BD or SD. The reliability of the onset times corresponding to BD and SD is obtained as the ratio of $d(t, f)$ currently under consideration to a recent maximum peak value.

3.2 Beat Prediction

Multiple agents interpret the results of the Frequency Analysis stage and maintain their own hypotheses, each of which consists of next beat time predicted, its beat type, its reliability, and current IBI. These agents, the number of which is NumAGENTS, 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 of 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. Agent-pairs differ (from other agent-pairs) in that they receive onset information from different onset-time finders. The agent-pairs can, in turn, re-adjust the parameters of the onset-time finders based on estimates of their own reliability − in other words, there is feedback between the (high-level) beat-prediction agents and the (low-level) onset-time finders.

Each agent has three parameters that determine its strategy for making the hypothesis. Both agents in the same pair have the same setting of these parameters. The settings of these parameters vary from pair to pair. Because the input signals are examined by these various viewpoints, various hypotheses can emerge.

The first and second parameters are *sensitivity* and *fre-*

quency range. These two parameters control the parameters of an onset-time finder, and influence the onset-finding strategy. The third parameter, *histogramming strategy*, takes the value *successive* or *alternate*. When the value is *successive*, successive onsets are used in forming the inter-onset-interval (*IOI*)[4] histogram; when the value is *alternate*, alternate values are used. If the reliability of an agent remains low for a long time, the agent adjusts these parameters so that they are close to these of the most reliable agent.

The following sections discuss the formation and management of hypotheses. First, each agent calculates the IBI, and then predicts the next beat time and evaluates its own reliability (**3.2.1**). Second, the agent infers its beat type and modifies its reliability (**3.2.2**). Finally, the most reliable hypothesis is selected from hypotheses of all agents (**3.2.3**).

3.2.1 Predicting next beat

Each agent predicts the next beat time by adding the current IBI to the previous beat time (Figure 4). The IBI is calculated as the most frequent interval between onsets with high reliability. The IBI is given by the interval with the maximum value in the inter-onset-interval (IOI) histogram (Figure 5). Before the agent adds the IBI to the previous beat time, the previous beat time is adjusted to the nearest onset time if they almost coincide.
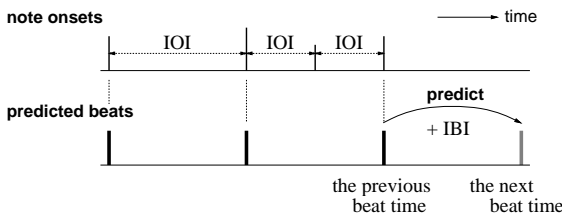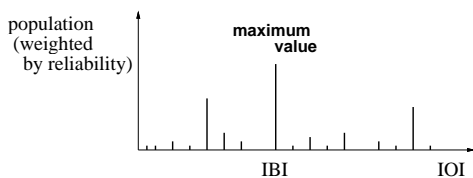
Figure 4: Beat prediction

Figure 5: IOI Histogram

Each agent evaluates its own reliability. The reliability is increased if an onset time coincides with the beat time predicted previously. In addition, if an onset time coincides

---

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

with the time that corresponds to the position of an eighth note or a sixteenth note, the reliability is slightly increased. Otherwise, the reliability is decreased.

3.2.2 Inferring beat type

Each predicted beat time is labeled with its beat type in this process. Although it would be better to infer the beat type on the basis of a variety of clues, in our current implementation, the beat type is inferred only from the onset times of BD and SD.
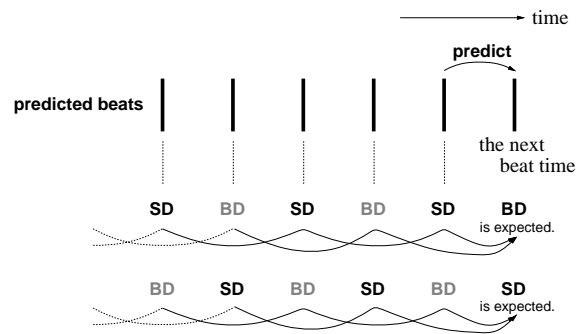
Figure 6: Expectation of BD and SD

The detection of an individual occurrence of BD or SD is likely to be corrupted by noise or other sounds. Furthermore, some BD or SD sounds may be missing, as in the case of a syncopation. BTS compensates for these difficulties by taking advantage of the fact that the BD and SD sounds generally alternate (Figure 6); so for example, following a sequence BD SD ? SD BD ? ? SD BD SD, BTS would expect BD. In this calculation, SD carries more weight than BD, because BD sometimes does not sound on beats in a syncopated rhythm and the possibility of SD sounding on beats is higher than that of BD.

If BD (SD) is expected, BTS concludes that the next beat type is a strong (weak) beat. In both cases, the reliability is increased so that the hypothesis with the IBI corresponding to a quarter note is likely to be selected.

3.2.3 Managing hypotheses

All agent-generated hypotheses are classified into groups according to beat time and IBI. Each group has the total reliability given by the sum of the reliability of hypotheses in its group. The most reliable hypothesis in the most reliable group is selected and sent to the BI Generation stage, which performs some post-processing operations and preparations for transmission of BI to the network.

4 Implementation

Figure 7 shows the implementation on a distributed memory parallel computer, the Fujitsu AP1000. The use of a parallel computer enables us to handle this computationally-intensive task in real time. The AP1000 consists of 64 cells[5]. We divide the 64 cells into 8 groups to which the different processes are assigned directly. The number of cells used in each group is indicated at the lower right of the rectangles in Figure 7. The arrows indicate the global flow of data between groups.
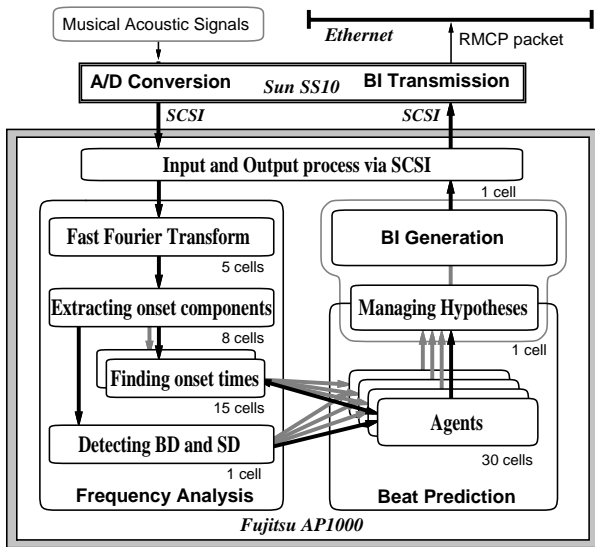
Figure 7: Implementation on the AP1000

In our current implementation, NumAGENTS (the number of agents) is 30. Both of the number of onset-time finders and the number of agent-pairs are therefore 15. Initial parameters of these agent-pairs are shown in Table 1. The unit of sensitivity is the frame-time defined below.

4.1 A/D Conversion

The input acoustic signal is digitized at 16bit/22.05kHz and then divided into blocks of 256 samples on Sun SPARC Station 10 (SS10). These blocks are transferred from SS10 to AP1000 via SCSI. A block consisting of 256 samples is a fundamental unit in all processes of BTS. The time corresponding to 1 block is called 1 frame-time (11.61msec). The frame-time is the unit of time used in BTS, and the term "time" in this paper is defined as the time measured with the frame-time.

---

[5]Cell means a processing element in the AP1000. Each cell consists of a 25MHz SPARC with FPU, 16Mbytes DRAM and 128kbytes direct-mapped cache memory.

Table 1: Initial parameters of agent-pairs

pair No.	sensitivity	frequency range	histogramming strategy
1	11	0-11kHz	successive
2	13	0-11kHz	successive
3	15	0-11kHz	successive
4	17	0-11kHz	successive
5	19	0-11kHz	successive
6	21	0-11kHz	successive
7	23	0-11kHz	successive
8	25	0-11kHz	successive
9	13	0-11kHz	alternate
10	19	0-11kHz	alternate
11	15	0-430Hz	successive
12	15	430-1.3kHz	successive
13	15	1.3k-3.0kHz	successive
14	15	3.0k-6.5kHz	successive
15	15	6.5k-11kHz	successive

In our current implementation, WinSIZE (the window size of FFT) is 4 frame-times (1024 samples). ShiSIZE (The shifted size of each window) is 1 frame-time (256 samples).

4.2 BI Transmission, RMCP system

BI is transmitted to the Ethernet as the RMCP packet at the predicted time. RMCP stands for remote music control protocol, which is a communication protocol on the UDP/IP between servers and clients in the RMCP system [10]. The RMCP system is a distributed cooperative system that integrates MIDI and LAN. This enables computers on the Ethernet to use the BI in various ways. For example, a workstation connected to the MIDI instrument may create drum-sounds or clapping-sounds in time to the input music. A workstation with a graphics engine may also create computer graphics synchronized with music.

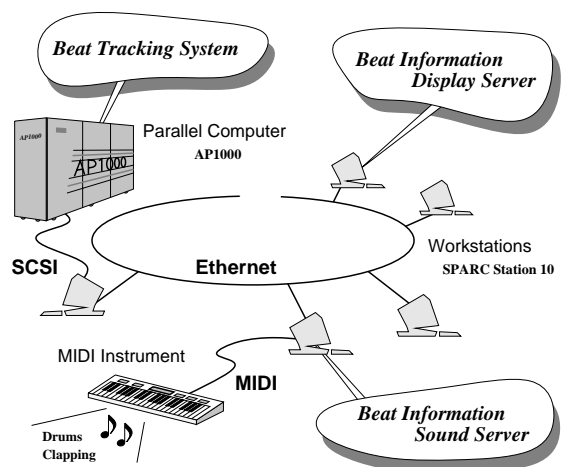We can both see and hear the result of BTS on the RMCP

Figure 8: BTS, BIDS, and BISS on RMCP system

system (Figure 8). BI is graphically represented with another program called *Beat Information Display Server (BIDS)*. BIDS provides a color display of BI received from BTS via the Ethernet, so that we may see beat time and beat type in time to music. BI is also aurally represented with another program called *Beat Information Sound Server (BISS)*. BISS plays drum-sounds or clapping-sounds by controlling a MIDI instrument. In the case of drum-sounds, BISS plays a bass drum sound on strong beats and a snare drum sound on weak beats. High hat cymbals can be played on the eighth or sixteenth notes if BTS transmits BI corresponding to the sixteenth notes[6].

5 Experiments and Results

We tested BTS on 30 popular songs in the rock and pop music genre. The initial one or two minutes of those songs were used as the input. The input was a monaural acoustic signal sampled from a commercial compact disc, in which BD and SD usually sounded on the strong and weak beats, respectively. Their tempi ranged from 78 M.M. to 168 M.M. and were almost constant.

Table 2: Experimental results*

No.	title (artist)	result	tempo
1	Julian (Princess Princess)	○	78
2	Dancing queen (ABBA)	○	99
3	Something about you (LEVEL 42)	○	106
4	Pride (U2)	×	106
5	Need you tonight (INXS)	○	109
6	Satisfied (Richard Marx)	○	109
7	Love story ha totuzen ni (Kazumasa Oda)	×	114
8	Open your heart (Madonna)	○	115
9	Black or white (Michael Jackson)	○	115
10	Sussudio (Phil Collins)	○	121
11	All that jazz (Breathe)	○	123
12	Seeing is believing (Mike + the Mechanics)	○	133
13	Any way you want it (Journey)	○	139
14	Be good to yourself (Journey)	○	151
15	Arcadia (T-Square)	×	156
16	Danger zone (Kenny Loggins)	○	157
17	Mighty wings (Cheap Trick)	○	159
18	Surfing with the alien (Joe Satriani)	○	168

* There were 12 other Japanese songs in which BTS correctly tracked beats, besides those listed above.

The test results are shown in Table 2. BTS correctly tracked beats in 27 out of 30 songs in real time. At the beginning of each song, the beat type was not obtained correctly even if the beats were found. This is because BTS had not yet acquired the characteristic frequencies of BD and SD.

---

[6]BTS has an optional function to generate BI corresponding to sixteenth notes by interpolating between beat times corresponding to quarter notes. This function is useful if short-interval synchronization is required.

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 the songs No.4, No.7 and No.15. In the song No.4, BTS tracked beat times correctly; however, the beat type was sometimes wrong. The power of BD was sometimes too low to detect the peak frequency corresponding to BD. Because the lowest detected peak corresponded to SD, BTS regarded SD as BD. In the song No.7, although BTS tracked beat times correctly, the beat type was reversed as if BD were SD. BTS did not acquire the characteristic frequencies of BD and SD correctly, because both of the lowest peak and the maximum peak of the histogram were the common frequency peaks in both BD and SD. In the song No.15, BTS tracked beats correctly, for the most part, but during about three measures in the middle, the beat type was reversed due to some irregular rhythm in the drums.

6 Applications of BTS

Musical beat trackers such as BTS are useful in various multimedia applications. This section discusses some typical examples.

☐ **Video editing**

Computer synchronization of visual tracks and audio tracks was discussed in a number of works such as [11]. If these tracks are already prepared or recorded at the same time, simple synchronization is enough. However, in video editing systems, it is necessary that motions or events in a visual track are synchronized with music in an audio track by hand, if the users would like to create MTV-style videos in which visual motions are synchronized with beats. BTS can be used to synchronize events in a visual track with musical beats in an audio track automatically.

☐ **Audio editing (Hard disk recording system)**

Current audio editing systems work with raw acoustic wave data. BTS enables editors to handle acoustic signals as a set of beats instead of raw data. Future versions of BTS will enable handling of acoustic signals at higher levels of musical abstraction.

☐ **Stage lighting control**

BTS is useful in automatically synchronizing computer controlled stage lights with beats of a musical performance. For instance, various properties of lighting such as color, brightness, direction, and effect can be changed in time to the music.

☐ Real-time computer graphics

BTS makes it easy to create real-time computer graphics synchronized with music. For example, a virtual dancer on a computer graphics display may dance in time to music using BI. The motion, step and position of the dancer can be changed with the beats. We have developed a server called *RMCP Animation Server* on the RMCP system, which displays a virtual dancer whose motion is changed with beats in real time (Figure 9).
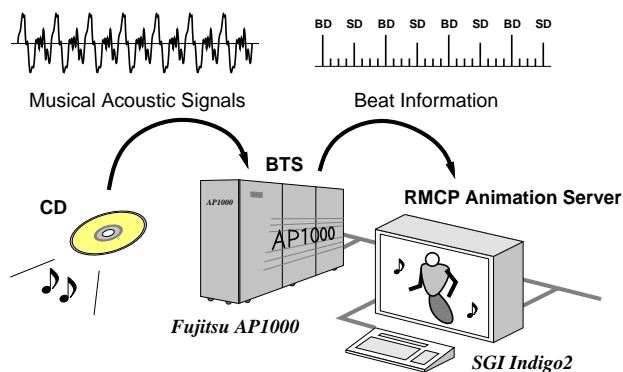

Figure 9: Computer graphics synchronized with music

☐ Live performances

Beat tracking can also provide computers the ability to participate intelligently in live performances. The computer may follow a human performance in real time and join the ensemble.

7 Conclusion

We described the configuration and implementation of a real-time beat tracking system (*BTS*). BTS tracks beats in acoustic signals that contain sounds of various instruments that include drums, and reports beat information corresponding to quarter notes in time to input music. BTS has been implemented on a parallel computer, the Fujitsu AP1000. In our experiment, the system correctly tracked beats in 27 out of 30 popular songs in the rock and pop music genre.

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 wrong. BTS automatically acquires the characteristic frequencies of bass drum and snare drum, and then detects them. This makes it possible to determine whether a beat is a strong beat or a weak beat.

We plan to improve the algorithm of the Beat Prediction stage in order to reduce the limitations on the input. Future work will include improvement of the method of detecting drums, a study on recognition of higher level musical events such as the beginning of measure, and application to a multimedia system.

Acknowledgments

References

[1] Roger B. Dannenberg and Bernard Mont-Reynaud: *Following an Improvisation in Real Time*, Proceedings of the 1987 International Computer Music Conference, pp.241-248 (1987).

[2] Peter Desain and Henkjan Honing: *The Quantization of Musical Time: A Connectionist Approach*, Computer Music Journal, Vol.13, No.3, pp.56-66 (1989).

[3] Paul E. Allen and Roger B. Dannenberg: *Tracking Musical Beats in Real Time*, Proceedings of the 1990 International Computer Music Conference, pp.140-143 (1990).

[4] Anthonie Driesse: *Real-Time Tempo Tracking Using Rules to Analyze Rhythmic Qualities*, Proceedings of the 1991 International Computer Music Conference, pp.578-581 (1991).

[5] David Rosenthal: *Emulation of Human Rhythm Perception*, Computer Music Journal, Vol.16, No.1, pp.64-76 (1992).

[6] David Rosenthal: *Intelligent Rhythm Tracking*, Proceedings of the 1992 International Computer Music Conference, pp.227-230 (1992).

[7] David Rosenthal: *Machine Rhythm: Computer Emulation of Human Rhythm Perception*, Ph.D. Thesis, Massachusetts Institute of Technology (1992).

[8] W. Andrew Schloss: *On The Automatic Transcription of Percussive Music − From Acoustic Signal to High-Level Analysis*, Ph.D. Thesis, CCRMA, Stanford University (1985).

[9] H. Katayose, H. Kato, M. Imai, and S. Inokuchi: *An Approach to an Artificial Music Expert*, Proceedings of the 1989 International Computer Music Conference, pp.139-146 (1989).

[10] Masataka Goto and Yuji Hashimoto: *A Distributed Cooperative System to Play MIDI Instruments − Toward a Remote Session −*, Information Processing Society of Japan SIG Notes, Vol.93, No.109, 93-MUS-4-1 (1993) (in Japanese).

[11] David P. Anderson and George Homsy: *A Continuous Media I/O Server and Its Synchronization Mechanism*, Computer, Vol.24, No.10, pp.51-57 (1991).