

A MULTI-AGENT ALGORITHM FOR REAL-TIME AUTOMATIC BEAT AND TEMPO SYNCHRONIZATION

Joanne C. Santos Edwin M. Umali Ian Dexter Garcia

Department of Electrical and Electronics Engineering
University of the Philippines Diliman, Quezon City

Abstract - The first thing a person notices when he hears music is its rhythm. Foot-tapping, hand-clapping, and dancing comes naturally to humans. So for computers to be able to understand music just like humans, the first thing it must learn to do is track beats. Once the beat is determined, automatic transcription and other applications such as music video editing will naturally follow. Beat tracking involves synchronization and determining the beat of the music as it comes, not after the entire music has been played. To be able to synchronize with the beat, the system should have be able to determine the beat before it arrives. In this paper, we discuss the development of an algorithm for beat and tempo synchronization that performs in real-time and deals with real world audio signals, specifically, compact disc music. The algorithm consists of a time-frequency domain salient onset detection, tempo induction, and multi-agent beat prediction. An agent is chosen based on different musical cues that best identifies the beat and tempo of a musical piece in real-time.

Key Words – beat tracking, tempo synchronization

1 Introduction

The beat, exemplified in foot-tapping or hand-clapping by humans, is fundamental to the understanding of music. So for a computer to possess musical intelligence, beat tracking is the first thing it must learn to do. Once it determined the beat, a higher level of music perception needed for automatic transcription and human-computer improvisation can be acquired. Beat synchronization can be used for automated music video editing, indexing of data, audio editing, automatic accompaniment, real-time animations or graphics, and real-time stage lighting control in live performances.

For a beat tracker to be most useful, it should be able to deal with real world signals, and should be able to perform in real-time. Therefore, the objective of this research is to create a real-time algorithm for beat and tempo synchronization that deals with real world audio signals, specifically, compact disc music.

In this paper, a beat is defined as the pulse perceived when listening to music, which is equally spread out locally. It does not include higher rhythmic levels. Tempo refers to the rate at which notes are played and it is normally expressed as the number of beats per unit time. Its inverse, the inter-beat-interval

(IBI), the difference between two successive beats, is used interchangeably with tempo.

Beat tracking is the task of locating the occurrences of beats in time in a given musical piece. Tempo tracking is determining the tempo at any particular time in the music.

When the term inter-onset-interval (IOI) is used, it refers to the intervals between any pair of onsets. Intervals between successive onsets are qualified as adjacent IOI.

A multiple-agent architecture consists of several agents, each having its own viewpoints, that independently performs a task. Then, it is a matter of choosing which among them performs the task correctly.

This algorithm uses a multiple-agent approach. The systems consist of several agents, each one composed of its own beat prediction and tempo hypothesis which track the beat of the input music. A score, based on some musical cues, will be kept for each of them to determine which agent is the most reliable, and consequently, which prediction will be followed. The evaluation of the agents shall be done with each musical event.

For onset detection, this system used Goto and Muraoka's time-frequency algorithm of finding peaks in summations of components with increasing power. There were some modifications that include adding a threshold to eliminate weak onsets. Tempo induction is done by getting inter-onset intervals in a given window. In beat prediction, the one with the highest score is chosen as the best agent. The score is computed based of the overall weight (degree of increase in power of onset) of the events it has considered as beat locations, number of occurrences of the predicted beat locations, closeness of the prediction, and number of beat hypotheses that are multiples. Figure 1 shows the block diagram of the project. **Error!**

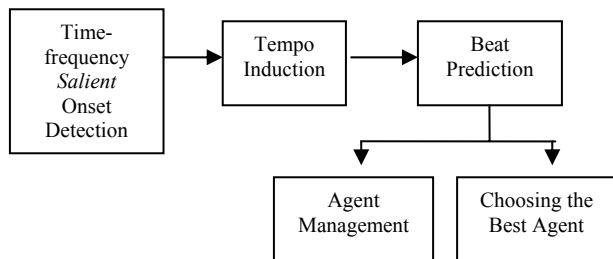


Figure 1 Block Diagram of the Algorithm

2 Review of Related Literature

A beat is a regular pulse perceived when listening to music. There is no specific sound that corresponds to a beat location. Fortunately, some information present in the audio signal can serve as clues to the beat. It is a low-level property of music, and simple estimates of salience, the importance of musical events, are enough to the determination of the beat.

Though there is really no specific sound that corresponds to a beat location, beats are usually found at note onset times. Beats are also more likely to coincide with strong or salient onsets.

Beat tracking systems are differentiated by a number of factors. They could perform processing either online or offline. Offline methods are able to utilize information from the entire musical piece. Online, or real-time, methods are limited in that the only information available are those that came from previous events.

The input data received by beat trackers could either be in the form of raw audio, onset times or MIDI. Audio input would require onset detection. Whereas MIDI input contains, apart from onset times,

information such as the pitch, note density and note duration.

The techniques used in the prediction of beats also vary, from probabilistic models to multiple-agent methods. The following are significant works on the multiple-agent method.

Goto and Muraoka [1],[2], [3] describes two real-time beat trackers processing raw audio input. The first one dealt with performances in which drums maintain the beat, using the drum sounds to infer beat locations by matching it to pre-stored patterns. The second system detects chord changes to make musical decisions for drumless signals and recognizes rhythmic structure at the quarter note, half note and measure levels. Goto [4] integrates the two systems into one beat tracking system for music with or without drum sounds.

Their algorithm used a time-frequency domain onset detection that takes into account the components whose power has been increasing. The peaks in the sum of these components are the onset times. The predicted beat is computed by adding the inter-beat-interval hypothesis to the previous predicted beat time. The inter-beat-interval is the most frequent adjacent inter-onset-interval.

Another author who has a series of works on beat and tempo tracking is Simon Dixon. Dixon [5] processes raw audio recordings offline and did not employ multiple hypotheses of beat locations. Dixon [6] uses the same input format, performs offline, but employs multiple agents to determine the beat. The confidence value for the agent is determined by its proximity to an event and the amplitude of the event. Dixon and Cambouropoulos [7] on the other hand accepts MIDI input, therefore have more information apart from onset times such as pitch and note duration. It uses these information to determine the salience of events which is used as a clue to the locations of beats. Dixon [8] was able to process both raw audio and MIDI input, and uses musical salience in predicting beat locations.

Dixon computes onset detection by using a simple high pass filter to eliminate weak onsets, and finds peaks in the amplitude envelope. A clustering algorithm is used to get possible tempos. This is done by grouping similar inter-onset-intervals.

Meudic [9] formulated a causal version of Dixon's beat tracking algorithm, to be able to use it in real time. It can be used for MIDI-like input containing the onset time, duration and pitch information. This work made

use of markings to identify the more rhythmically important events.

Meudic computes possible tempos by getting intervals from the current onset to all previous onsets within a window. The best beat is that which has the greatest number of multiples and highest total weight.

3 Materials and Methods

3.1 Input Format

The format of the input data shall be in uncompressed linear pulse code modulated (PCM) signals, as that found in compact discs. The file format that will be used is .wav format.

3.2 Onset Detection

Note onsets can best be detected by looking at the time-frequency response of the music. A simple amplitude peak detection sometimes does not work because some notes do not have very pronounced peaks at their onset. Some may also be masked by other notes. However, if we look at a spectrogram of a signal, onset times are marked by very noticeable peaks, especially in the higher frequencies (see Figure 2). The noticeable lines are the onset times.

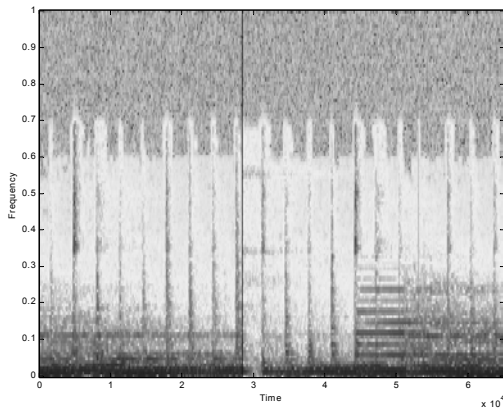


Figure 2. Spectrogram of a Sample Music

The frequency spectrum is calculated using an FFT with window length of 1024 samples and overlap of 256 samples. Onsets are found to be peaks in

$$D(t) = \sum_f d(t,f) \quad (1)$$

where

$$d(t,f) = \max(p(t,f), p(t+1,f)) - \text{PrevPow} \\ \text{when } (\min(p(t,f), p(t+1,f)) > \text{PrevPow}) \quad (2)$$

$$\text{PrevPow} = \max(p(t-1,f), p(t-1, f \pm 1))$$

f = frequency range where the summation is applied

Onset components ($d(t,f)$) are points in the time-frequency domain which have power that is increasing with respect to nearby time and frequency regions (see Figure 3). Figure 4 shows the power spectrum of a sample signal and Figure 5 shows the points in the spectrum which are onset components (the points with nonzero power).

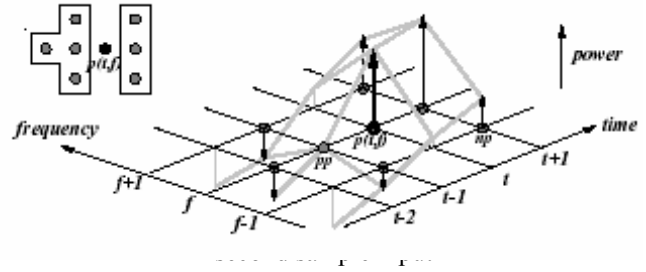


Figure 3. Goto's Onset Detection [1]

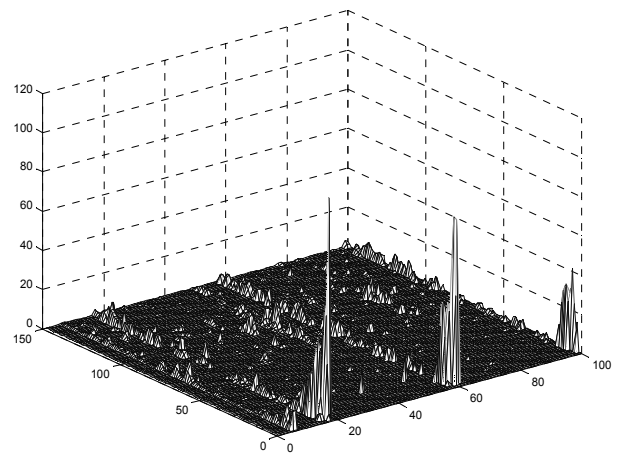


Figure 4. Time-frequency domain graph of the two-second sample input

The algorithm uses Goto's method for onset detection, without the smoothing function. Instead, an absolute threshold is imposed on the peaks in the summation of degree of increase in power. Only those that are above the threshold are considered onsets. This method eliminates the weak onsets. This is implemented by testing the algorithm with beat locations as onsets, such that the algorithm only tracks the beat locations with minimal additional onsets. The frequency ranges used for the summation are the following:

f1=(freq>=20 & freq<470);
 f2=freq>=470 & freq<11025);
 f3=(entire frequency range);

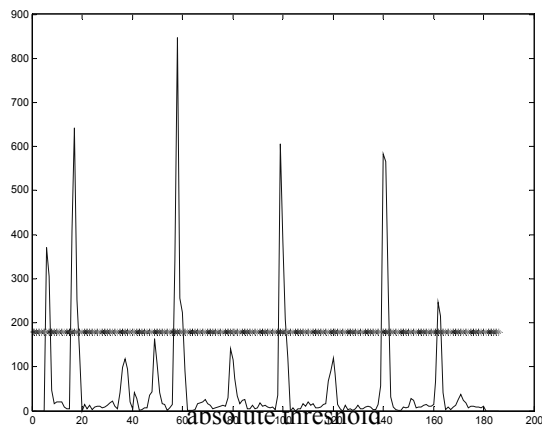


Figure 5. The onset components of the input signal from Equation 2

In this research, three variations of this onset detection is performed. One is getting onsets from the three frequency ranges and using the number of successive IOI's that are multiples as basis for choosing the best set of onsets. The other two is getting onsets only from the second, and third frequency range, respectively.

For the method that takes into account three frequency ranges, the onsets are chosen by counting the number of multiple IOI's. At each onset, the interval with the previous onset is computed and compared with the IOI previous to it. If they are multiples, the score for the onset finder increases. It is increased even more if they are equal. The score increases with every additional onset that is also a multiple of the previous onset's IOI. The score is decreased if the next onset does not have an IOI that is a multiple. The highest scoring onset finder at each time is chosen.

The onset detection method that yielded the best results when tested is using the third frequency range, which is the entire frequency range.

3.3 Tempo Induction

Tempo hypotheses are formed by getting intervals from the current onset to all previous onsets within a window. Each onset has a reliability value, which is the ratio of its degree of onset with the recent maximum. If this value is greater than `rel_thresh` (which currently has a value of 0.7), new possible IBI's are computed, which are IOI's between the current onset and previous onsets in a window.

Different listeners would sometimes track beats at different metrical levels. For example, for a song with a slow tempo, a listener may track the song in half the tempo. To prevent discrepancies in the beats tracked by the algorithm and the beats tracked manually, the tempo is limited to a certain range. The possible tempo for this implementation is between 80-160 beats per minute, that is, an IOI must be between 0.375 and 0.75 seconds.

3.4 Beat Prediction

When the onset's reliability is greater than `rel_thresh`, the new possible IBI's would also start on a new beat location, the current onset. The predicted beat location is computed simply by adding the IBI to the onset time.

For the other older agents, if the current onset coincides with their beat prediction, the new predicted beat is computed the same way. If not, the IBI is added to the previous predicted beat.

The only output that is needed is the current tempo and the next beat location. The tempo can be updated by incorporating a percentage of the error in tempo. The weight, which determines the score, can be accumulated in one variable. The previous beat locations are no longer needed once the next location is established. The agents in this implementation maintain only the current tempo, predicted beat and score.

3.4 Agent Management

In a multiple-agent architecture, the less the number of agents, the better. Reasons include less memory requirement, less number of computations, and less possibilities to choose the best agent from. To limit the number of agents, agents are checked whether they have the same tempo and beat prediction. The higher scoring agent will be retained, the other removed from the list. Also, agents who have not been making

correct predictions for a given period are deleted from the list.

3.5 Agent Scoring

Seven methods of scoring were analyzed:

- Weight (degree of increase in power) normalized by IBI

$$D(t) * IBI \quad (3)$$

- number of occurrences normalized by the IBI
- logarithmic weight normalized by the IBI

$$\text{LOG}_{10}(D(t)) * IBI \quad (4)$$

- logarithmic weight normalized by the IBI and 1-error, where error is the difference between predicted beat location and actual onset

$$\text{LOG}_{10}(D(t)) * IBI * (1-\text{error}) \quad (5)$$

where error = $(1 - (\text{predicted_beat} - \text{onset}) / \text{tolerance window})$

- reliability normalized by the IBI
- total weight
- number of multiples

It is found that the sixth scoring strategy, total weight, yields the best result.

4 Discussion of Results

To test the performance of the algorithm, its output beat locations are compared with manually marked beats (quarter notes). An output beat is considered correct if it is within 70 ms either side of the correct beat times [8].

Manual marking was done using Transcriber, with the beats marked with breakpoints. The time of these breakpoints are the correct beat times.

Goldwave was used to get 30-second excerpts from sample music. There were 45 30-second random samples that are transcribed. However, only 25 of these meet the tempo range required by the algorithm. These are listed in Table 1.

The hit rate(number of tracked correct beat locations), the miss rate(number of correct beat locations not tracked), and the ratio of false alarms over hits are computed.

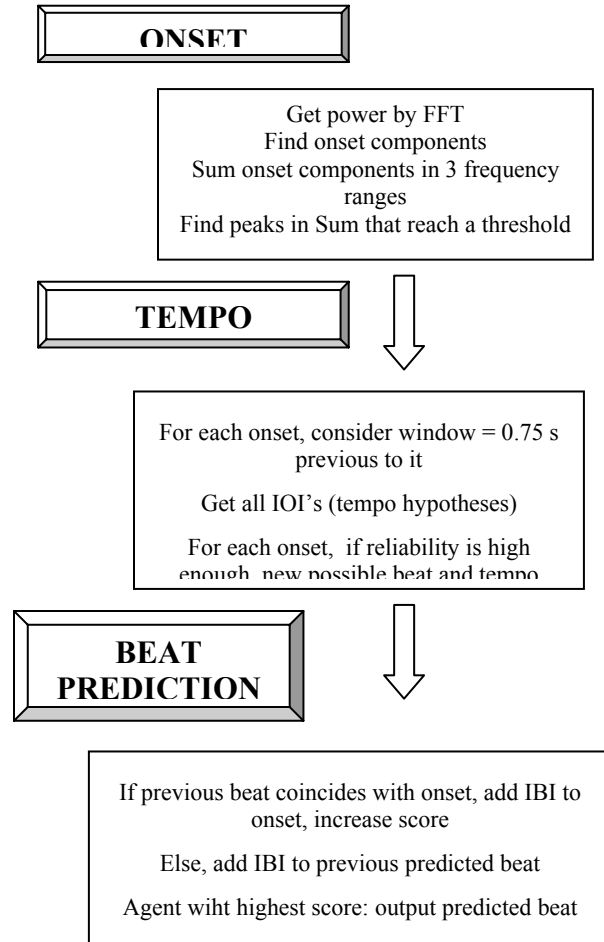


Figure 6 Flow Chart of New Implementation

To combine these three measures of beat tracking accuracy, an evaluation formula was used [8]:

$$\text{Evaluation} = \frac{\text{Hits}}{\text{Hits} + \text{Misses} + \text{False Alarms}} \quad (6)$$

Table 2 shows the performance of the algorithm using the evaluation formula for different types of music. The algorithm successfully tracked the rock, pop, dance and RnB samples. For the ballad excerpts, the quarter notes do not have salient note onsets. At some points, there are actually no onsets at all; the beat is just mainly inferred from the half notes (the intervals of which are not within the tempo range). It is for this reason that the algorithm failed tracking ballad songs. Table 3 and 4 shows the overall accuracy of the output beats. The computation for Table 4 used a less strict tolerance window to show how much are near mistakes.

Table 1 Test Set

Title	Type
Again	Rock
Billie Jean	Dance
Black or White	Dance
Black Velvet	Dance
Don't Turn Around	Dance
Dream Lover	Pop
Like a Virgin	Dance
On the Line	Pop
Roll With It	Rock
The Sign	Dance
What a Feeling	Dance
All for You	Dance
All the Love in the World	Pop
Best In Me	Ballad
Breathless	Pop
Get the Party Started	Dance
How deep is your Love	Ballad
I Knew I Loved You	Ballad
I'm Real	Rnb
Irresistible	Pop
Let Me Blow your Mind	Rnb
On a High	Pop
Say	Pop
Turn the Clock Around	Pop
Walking on Sunshine	Dance

Table 2 Evaluation Values for the Different Types of Music in the Test Set

	Mean	Std Dev
Rock	1	0
Pop	0.9347	0.096464
Dance	0.91955	0.100509
Ballad	0.440467	0.402312
RnB	0.83335	0.16665

Table 3 Accuracy Measures for Beat Detection using a tolerance window of 70 ms

Hit Rate	Miss Rate	False Alarms/ Hits	Evaluation
0.8906	0.0995	0.0505	0.8664

Table 4 Accuracy Measures for Beat Detection using a tolerance window of 150 ms

Hit Rate	Miss Rate	False Alarms/ Hits	Evaluation
0.909	0.080	0.031	0.899

Error!

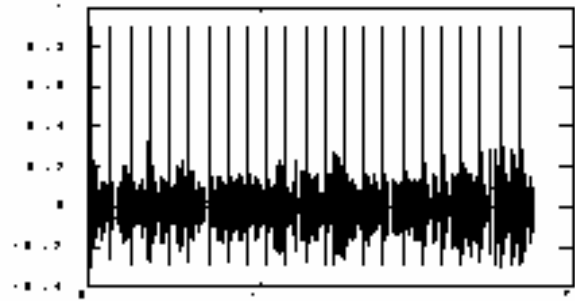


Figure 8 The input music with the beats tracked by the algorithm (vertical lines)

4. Conclusions

An algorithm was developed for beat and tempo synchronization that is suitable for a real-time implementation that deals with compact disc music. The algorithm consists of a time-frequency domain salient onset detection, tempo induction, and multi-agent beat prediction. Choosing the best agent is studied using different musical cues. The total weight, or degree on increase in power, is the most reliable scoring method. The best frequency range for getting onsets is the entire frequency range.

The memory requirement is small; the algorithm only needs a 0.75 second segment of the song to make a prediction. Then, it only needs to remember only the tempo, beat prediction and score for each agent.

The algorithm was tested on 25 30-second samples of pop, rock, dance, RnB and ballad songs.

The algorithm has an average of 86.64 % evaluation score. That is, excluding the first eight seconds of the sample, which is approximately the average time it takes for the algorithm to learn the beat.

The tempo of the input music is restricted to be within 80 to 160 beats per minute. Without limiting the tempo range, for some samples, the algorithm tracks twice the inter-beat-interval as determined manually. This is a reasonable mistake because different listeners may track beats at different metrical levels. To avoid having discrepancies in the metrical level of beat tracking, the tempo restriction was imposed.

An obvious future work after this project is an actual real-time implementation. The algorithm is tailored for real-time. It is completely causal, computationally simple, and a small amount of information is needed to be stored. An online implementation has more opportunities for application,

computer synchronization with live performances and possibly embedded systems.

To improve the algorithm even more, it should be further tested for a variety of musical styles, including more samples of classical and expressive music and maybe some traditional or ethnic music.

Another possibility is to extend the algorithm to include detection of weak beats; music sometimes has beats that have uneven intensity. This algorithm focuses on detecting that one regular strong pulse that is locally even. Going even further, this project could be extended into a rhythm detection, not only determining beat locations, but also the beats' hierarchy in the piece. This could be used as part of music transcription.

References

- [1] M. Goto, and Y. Muraoka. A beat tracking system for acoustic signals of music. In Proc. of the Second ACM Intl. Conf. on Multimedia, 365-372, 1994.
- [2] M. Goto and Y. Muraoka. 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, 1996.
- [3] M. Goto and Y. Muraoka. Real-time rhythm tracking for drumless audio signals { chord change detection for musical decisions. In Proceedings of the IJCAI'97 Workshop on Computational Auditory Scene Analysis. International Joint Conference on Artificial Intelligence, 1997.
- [4] M. Goto. An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds. Journal of New Music Research, vol. 30, No. 2, pp. 159-171, June 2001.
- [5] S. Dixon. A beat tracking system for audio signals, in Proceedings of the Diderot Forum on Mathematics and Music, pp. 101--110. Austrian Computer Society, 1999.
- [6] S. Dixon. A lightweight multi-agent musical beat tracking system, in Proceedings of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis. AAAI Press, 2000.
- [7] S. Dixon and E. Cambouropoulos. Beat tracking with musical knowledge. In ECAI 2000: Proceedings of the 14th European Conference on Artificial Intelligence. IOS Press, 2000.
- [8] S. Dixon. Automatic extraction of tempo and beat from expressive performances. J. New Music Research, 30(1), 2001.
- [9] B. Meudic. A Causal Algorithm for Beat Tracking. Submitted to the International Conference on Music Information Retrieval. 2002.