A REAL-TIME BEAT TRACKING SYSTEM WITH CONFIDENCE ANALYSIS ON EMBEDDED SYSTEM

Chun Wei Chen, Bo-Yao Chen, Hong-Yen Huang, Yu-Wei Huang, Tzu-Chun Yeh, Jyh-Shing Roger Jang

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
SPCUP2017 Team: 24680

ABSTRACT

BeatMir, a real-time beat tracking system which first find the last beat position and tempo by a small delay and predict the future positions by calculating the beat period and take the last analyzed beat position as starting position since we assume the tempo in a music will not change frequently in short time. The system will continuously analyze the newest beat position and tempo and update the predicted result. We also design a subsystem to determine if the confidence of the detected tempo is high enough. We use the aubio[1] as our real-time beat-tracking core but replace the spectral flux with SuperFlux[2]. *BeatMir* is run on Raspberry Pi 3 and evaluated on the dataset provided in spcup 2017. The results are compared to the offline beat tracking because we want to make our real-time result work as well as offline result by same beat tracking method.

1. INTRODUCTION

Since the format of music is not as intuitive as the format of image, many works relate to music require several features to represent the signal. Beat is an essential feature of music. Beat tracking is a fundamental research topic in the whole music information retrieval. However, in real-time beat-tracking system, there are many issues need to be concerned. For example, in the offline beat tracking system, we can utilize and manipulate the features from the whole musical excerpt, such as mean, median function, or normalization according to the information extracted from the whole excerpt, but we cannot obtain such information in the online system. Since the feature set is restricted, to keep the performance as good as the offline system is a challenging issue. Another issue is to predict the beat position according to the current data, more difficult than analyzing the beat position from the existed data time. For these purposes, we provided a real-time beat tracking system called BeatMir, dedicating to handle with these issues. BeatMir is a parallel system which predict the beat positions while continuously analyze the newest data and update the features we have.

In this paper, we will introduce our system structure in detail in section 2 and briefly describe the SuperFlux and aubio in our system in section 3. Section 4 provides detailed introduction of our prediction method. Section 5 is the experiment result.

2. THE SYSTEM STRUCTURE

Figure 1 shows an overview of beat tracking system BeatMir. The three main stages of our tracking system are Communicator, Detector and Predictor.

In first stage, Communicator “listen” the music by a microphone. It will collect the signal data, convert analog signal to digital signal and write into a buffer. After a buffer is sent to the beat tracker buffer, communicator will wait the response of Detector (Detection Stage) without stopping listening and writing signal.

In Detection stage, Detector collect the data from buffer and send it to Aubio every 0.5 second. A 5-second-long audio data is sent to Aubio once a time. Aubio does FFT to transform audio data and uses super flux to detect the onset beat position. Detector calculates analyzed data and detects bmp, confidence and the last beat position in this stage.

In Prediction stage, Predictor gets a signal from detector to update the latest bpm and beat position and predict the future beat position by the detected bmp and last beat position. However, the analyzed bmp and beat position might be an error since we only have a small segment of the music. As a result, we need to determine whether the detected result is trustable or not before updating the new predicted beat position in Tapper.

The following sections describe two of the main stages: Detection and Prediction.

3. BEAT TRACKING

Beat tracking is an essential part in the system since we need to analyze the past music to predict the future. There are already many great works about beat tracking. As a result, we just used the well-known aubio as our framework and replaced the spectral flux with concept of SuperFlux.

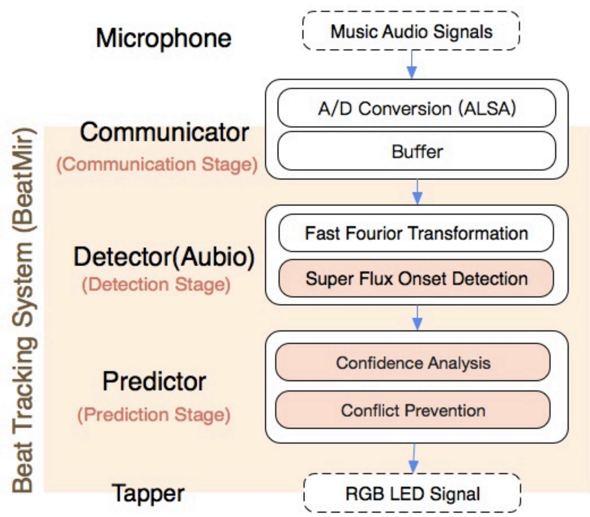

Figure 1: An overview of the structure of BeatMir.

Figure 2 shows the time-based processing flow in BeatMir system. While the system starts, BeatMir forks 3 main processes to deal with the signal data. As time goes, Detector-Process gets data from buffer and detects previous beat position, bpm and confidences. It will throw a signal to Predictor-Process every time it gets a new result. Predictor-Process predicts and updates next beat position in “tapper” when signal comes in. Tapper will tap to LED-process (throw another signal) when the clock time is about to “beat”. LED-process controls RGB-LED light for the main purpose of giving an output of predicted beat to human through the screen or through the RGB-LED .

Onset detection is an important part in beat tracking process [3]. Recently, SuperFlux is a very powerful onset detection algorithm with vibrato suppression provided by Sebastian Böck and Gerhard Widmer at 2013. With several experiments, which will be introduced in section 5, we found SuperFlux performed well with the dataset provided in spcup2017.

4. BEAT PREDICTION

The prediction is the most critical part in the whole system. There are two major parts in BeatMir, preliminary prediction , confidence analysis and conflict prevention.

4.1. Preliminary prediction

Formula (1) shows how we get the predicted beat positions:

$$P_i = \beta + i \times (60/T) \tag{1}$$

P_i is the i-th predicted beat position. β is the last beat position and T is the BPM. β and T are both come from the analysis of the past 5 second segment of the music.

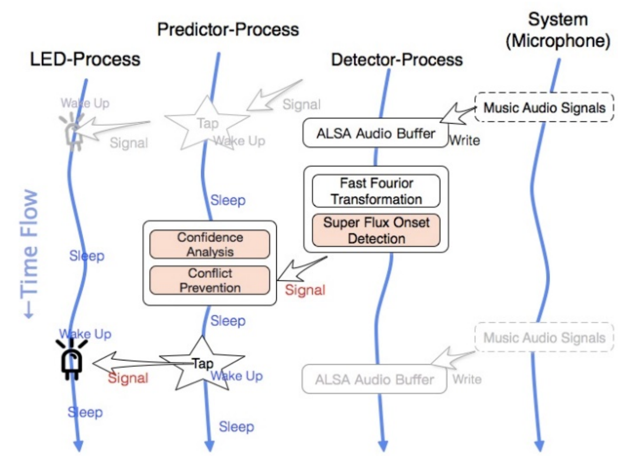

Figure 2: Time-Based Processing Flow

We assume that there is no drastic change of the tempo in music in short term, therefore we can just utilize the BPM of past 5 second segment to be the BPM in nearly future. Figure 3 will provide a detailed description about how these things work. Since the system will update the last beat position and BPM very frequently, the change of the BPM can still be detected with a small delay. Figure 3 shows how we predict the beat positions and Figure 4 shows how BeatMir update the newest result to correct the prediction.

However, frequently updating the BPM will cause several issues, such as conflict. To cope with these issues, we introduced a subsystem to tackle with the conflict. The details will be discussed in section 4.2 and 4.3.

4.2. Confidence analysis

We introduce a thresholding method to deal with the issues mentioned in section 4.1. It determines that an analyzed result is trustable or may be an error.

We use the “confidence” provided by aubio to perform this work. The follows are what this “confidence” is and how we use this variable to improve our system.
The key idea of confidence is how salient the beat positions are. Formula (2) shows how aubio calculate the confidence.

$$C_i = P_i \Big/ \sum_{S_t \in F_i} S_t \tag{2}$$

C_i is the confidence of i-th analysis of beat. P_i is the onset of the last beat position in the i-th analysis. F_i is the union of all the onsets in the i-th analysis. S_t is the onset in time t.

To use confidence in our system, firstly, we make the confidence of an analysis decrease over time since the result of analysis nearly is more trustable. Function (3) shows how this work.

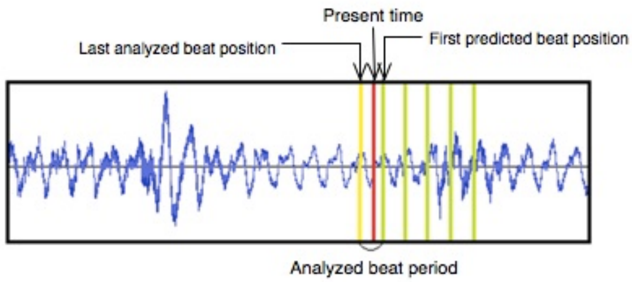

Figure 3: This figure show how we predict the beat positions. The red line is the present time and the yellow line is last beat position we get by trustable analysis (Trustability is discussed in section 4.2). The green lines is the predicted beat positions we get by assuming BPM (or beat period) will not change nearly. We can see a little error in the fifth green line, but the last analyzed beat position and BPM may already replace with more trustable beat position and BPM at that time.

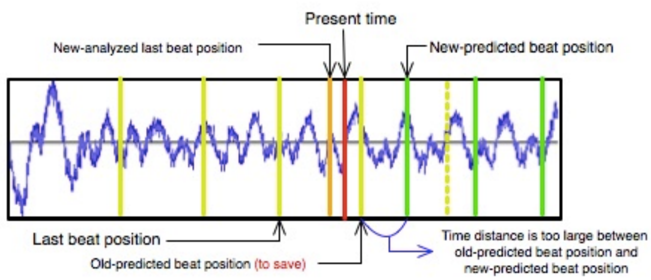

Figure 5: The new-predicted beat position is too far away from the old-predicted beat position. The system will discard the old-predicted beat position which makes it miss a beat around the present period of time.

$$C_i(j) = C_i(i) \times d^{j-i} \tag{3}$$

$C_i(j)$ means the confidence of the i-th analysis when making j-th analysis. d is a coefficient which is used to determine how much the confidence will decrease after next analysis. (d is set to 0.9 empirically in our system.)

The analyzed result is accepted when the new confidence is larger than all the past decreased confidence.

4.3. Conflict prevention

There are several cases that a beat will be skipped or repeated twice in real-time system. For example, Figure 5 shows that the period between the old predicted beat position and the first new-predicted position is too large, which makes the system lose a beat around the present time. To tackle with this issue, we propose a method, and the detail is described as follows.

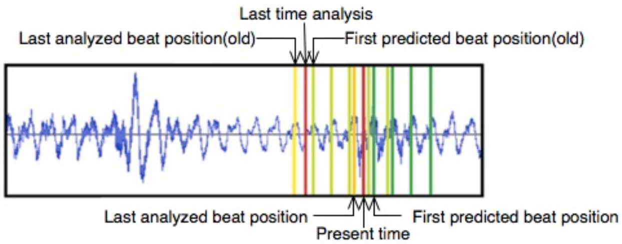

Figure 4: As we mentioned in Figure 3, there is a little error in the fifth light green line. BeatMir will analyze and update frequently therefore the prediction result is already corrected before time really come to that position.

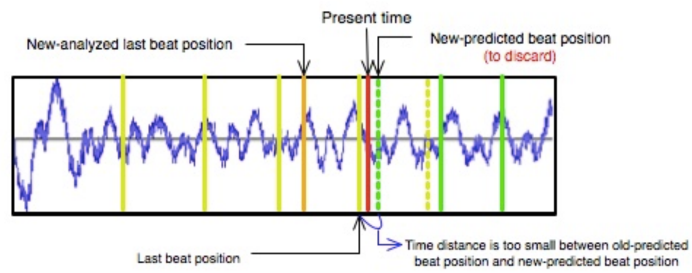

Figure 6: The new-predicted beat position is too close to the last beat position. The system will generate two consecutive beats in a short period of time.

In Figure 5, the system will discard the old-predicted beat position when the system updates the new-predicted beat position at present time. It will make us feel the system miss a beat if the new-predicted beat position is too far away from the old-predicted beat position. So we provide a rule as formula (4):

$$\alpha > p_i \times \Theta_s \tag{4}$$

α is the time distance between the old-predicted beat position in previous analysis and the new-predicted beat position in present analysis. p_i is the beat period in i-th analysis. θ_s is the threshold which is set to 0.8 empirically in our system.
The system would save the old-predicted beat position if the system satisfies this condition. BeatMir will not change to new pattern until the old-predicted beat is tapped.

Another issue is illustrated in Figure 6. The system will tap at the new-predicted beat time position after the system updates new analysis. It will make us hear quick two consecutive beats around the present period if the new-predicted beat position is too close to the last beat position. This situation will make the accuracy worse. The system would discard the new-predicted next beat position if it satisfies the condition in function (5),

$$\beta < p_i \times \Theta_d \quad (5)$$

β is the time distance between the last tapped beat position and the first new-predicted beat position in present analysis. p_i is the beat period in i-th analysis. θ_d is the threshold which is set to 0.3 empirically in our system.

After an analysis is acceptance and the conflict is avoided, the predictor will update the predicted beat position.

5. EXPERIMENT

It is important to do some experiment to find the best way we use for beat tracking and evaluate our final result. There are three major experiments we have done. Compare SuperFlux with normal spectral flux in our system, compare the offline result with real-time result and find the best distance to put the microphone.

We first compare aubio using SuperFlux with original aubio. Table 1 shows some major improvements on open dataset.

Table 1. Accuracy of musical excerpts using different method in open dataset.

	Open 003	Open 004	Open 015	Open 025
Original	0.183673	0.392857	0.056604	0.279070
Super Flux	0.534884	0.839286	0.905660	0.431818

As Table 1 shows, there is a huge improvement on some excerpts.

Table 2. Accuracy musical excerpts of offline and real-time system.

	Open 001	Open 002	Open 010	Open 011	Open 015	Open 018
Offline	1	0.913043	1	0.981481	0.90566	0.86
Real-time	0.952381	0.934783	0.977778	0.90566	0.113208	1

As Table 2 shows, accuracy of real-time system is usually a bit worse than accuracy of offline system. There are two special open 015 and open 018. In the case open 015, the result of real-time system is much worse than the result of offline system. There are too many instruments in different parts of this music, but the main theme of the music never changed. It make system focus on different source at different time. In the case open 018, real-time result is even better. We think this is because the real-time system focus on the music near the beat position and the other part of Open 018 may be noise for this beat position in overall offline analysis. Finally, we think the recording quality will affect the performance of our system, so we compare the performance in different distance between the speaker and the recorder. Table 3 shows the accuracy of first ten excerpts in open dataset with different speaker-recorder distance.

Table 3. The average accuracy of first ten musical excerpts in open dataset with different speaker-recorder distance.

	25cm	50cm	75cm	100cm
Accuracy	0.466543	0.620888	0.634034	0.629564

As Table 3 shows, 75 cm is the best distance for our system. Distance is one of the major factor which affect the recording quality and the recording quality will, of course, affect the prediction result. There still other factor that will affect the recording quality such as the maximum and minimum volume tolerance of the device (Microphone, computer, system, etc.), sample rate of the music and recording device (We use 44100 Hz to be our recording sample rate because it is the most common quality of CD) and the environment.

6. CONCLUSIONS

BeatMir is a real-time beat tracking system. There are three major parts in BeatMir, communicator (Communicate between all processes and listen to music.), detector (Beat tracking) and predictor (Beat predicting).

The major contribution of this paper is a method to analyze a result of beat tracking in real-time system is trustable or not and a method to deal with conflict in the real-time. Also, we focus on the efficiency of algorithm because this system run on an embedded system.

The weakness of this system is that it only focus on the past 5 second music. If there are too many accompaniment instruments in the different parts of the music, BeatMir will recognize accompaniment sound as the main melody and predict wrongly if the accompaniment instruments are too loud. Some history feature recording can help in the future work.

Here is our Demo video: https://youtu.be/gDWKiiMk-ZU

7. REFERENCES

[1] Brossier, Paul M. "The aubio library at mirex 2006." MIREX 2006 (2006): 1.

[2] Böck, Sebastian, and Gerhard Widmer. "Maximum filter vibrato suppression for onset detection." Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13), Maynooth, Ireland. 2013.

[3] Dixon, Simon. "Onset detection revisited." Proceedings of the 9th International Conference on Digital Audio Effects. Vol. 120. 2006.