International Journal of Innovative Research in Electronics and Communications (IJIREC)
Volume 1, Issue 2, May 2014, PP 18-24
ISSN 2349-4042 (Print) & ISSN 2349-4050 (Online)
www.arcjournals.org

A Pilot Study of Automatic Tempo Measurement in Rhythmic Music

Vedshree Panchwadkar
Cummins College of Engg
Pune, India
vedshreepd@gmail.com

Vanita Gadakh
Cummins College of Engg
Pune, India
gadakh.vanita@gmail.com

Makarand Velankar
Cummins College of Engg
Pune, India
makarand_v@rediffmail.com

Abstract: *Tempo in music is beats perceived by us in unit time. The tempo is measured as number of beats per minute (BPM) in a music clip. This paper presents a system for automatically calculating tempo of audio music signal. In this system, as an initial step after giving sound clip as an input, a threshold percentage in decibels selected. In the next step, samples having values above the threshold are considered as beats. We have used Audacity as an open source tool for the initial experiments. By using nyquist programming in audacity which is used specially for sound, we have added a nyquist plug-in in the Audacity open source software. This paper talks about the two newly added plugins by us which shows the beat positions of the selected sound clip and the time instances of these beats are retrieved using nyquist programming and thus using this equidistant beats are located, whereas the other plugin shows the actual calculated tempo of the same sound clip. Total number of beats is calculated from selected clip and using this beats per minute are obtained. The tempo of the sound clip is calculated based on the basic formula for beats per minute in which the total length of the sound clip is calculated and with the help of this, number of beats per minutes is obtained. For the sound clips which have more number of beats per minute has more tempo and those sound clips with less number of BPM or medium BPM has less or medium tempo respectively. This paper proposes the algorithm which gives the best results for rhythmic sound clips.*

Index Terms: *Beats Per Minute, threshold, audio samples,*

1. INTRODUCTION

Tempo in music is beats perceived by us in unit time. The tempo is measured as number of beats per minute (BPM) [1] in a music clip. This means that a particular note value (for example, a quarter note) is specified as the beat, and the marking indicates that a certain number of these beats must be played per minute. The greater the tempo, the larger the number of beats perceived in a minute. In electronic dance music, accurate knowledge of a tune's BPM is important to DJs for the purposes of beat matching. So the new functionality for measurement of tempo of music clip will be implemented. Tempo estimation is needed for various purposes by music composers, Djs and regular listeners like us as well. Manual calculation of tempo by keeping track of the number of beats in a music clip is quite a task. Thus, automated tempo measurement can prove helpful for reducing the efforts and time that is required in manual calculation of tempo. It is the basic time unit of music and it determines the temporal structure of an audio signal, making beat tracking a very important task in music information retrieval(MIR) [2].

MIR is the upcoming research field with various methodologies proposed based on content based music IR. At present most of the systems available like YouTube etc. uses meta-data based approach for Music Retrieval. Content based music retrieval is a novel approach and it can give better results over the meta-data based approach which is more suitable for text documents. Our initial work is a small contribution towards content based music search and retrieval.

The automatic extraction of BPM from musical signals is a challenging process due to many reasons such as offbeat rhythms, absence of rhythms in few cases or very fast or complex rhythms etc. Beat tracking in such cases and determining BPM automatically is very difficult. We have worked on simple rhythmic clips in our pilot experiments and have achieved acceptable results for simple rhythmic clips. We need to improvise our algorithm for more complex musical clips and utility for wide variety of musical genres.

2. Related Work

A brief description of some of the existing approaches to beat tracking is presented in this section.

Two algorithms are referred to measure tempo of music file. First one is an online musical beat tracking algorithm based on Kalman filtering(KF) with an enhanced probability data association (EPDA) [3] method is proposed. In this method,the music contents is represented via Mel-scale frequency cepstral coefficients (MFCC) [4].This beat tracking algorithm is built upon a linear dynamic model of beat progression, to which the Kalman filtering technique can be conveniently applied. The beat tracking performance can be seriously degraded by noisy measurements in the Kalman filtering process. Three methods are presented for noisy measurements selection. They are the local maximum (LM) method, the probabilistic data association (PDA) method and the enhanced PDA (EPDA) method. Also another algorithm called Tempo Detection Using a Hybrid Multiband Approach [5] is used for calculating beats per minute. The model tracks the periodicities of different signal property changes that manifest within different frequency bands by using the most appropriate onset/transient detectors [6] for each frequency band.

3. Working of Our Algorithm

As a base tool for implementing our algorithm, open source tool Audacity has been used. A Nyquist plug-in [7] has been added to Audacity for the implementation. The coding has been done using Nyquist supported Lisp language. For calculations some basic nyquist functions [8] are used such as s-length, snd-time, snd-rate, snd-fetch, snd-follow, etc. The basic working is explained in the following block diagram fig. 1.

As shown in fig.1, the initial step is to give sound clip as an input, then select a part or whole of the clip. Select the option Beats per minute then the threshold window will appear. Threshold percentage in decibels is selected. The samples having value above the threshold will get filtered. Out of these filtered samples only the prominent ones are considered as beats. Time instances of these beats are retrieved and thus using this equidistant beats are located. For calculating beats per minute, total length of the clip is measured and thus the average beats per minute are calculated.

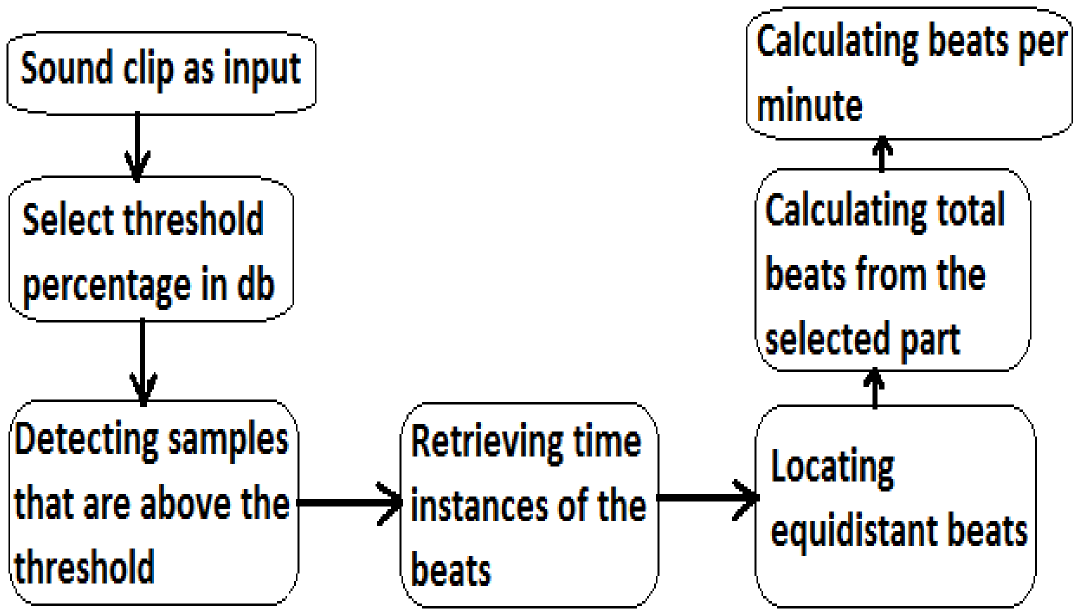

Fig1. *Block diagram of the working*

Following sequential screen shots of our system will explain the process and steps we followed for automatic BPM calculation.

Any rhythmic sound clip is given as an input as shown in fig.2.Click on the newly added plugin named as Beats Per Minute as shown in fig. 4 and select threshold. In fig.2the selected threshold is 65. After selecting threshold, beats are located as shown in fig.3. Then click on the newly added second plugin named as Beats Per Minute for rhythms as shown in fig 4 (Plugin next to Beats Per Minute). This plugin will calculate the BPM. Tempo (BPM) of rhythmic file is 130 for the specific sound clip as shown in fig.5.

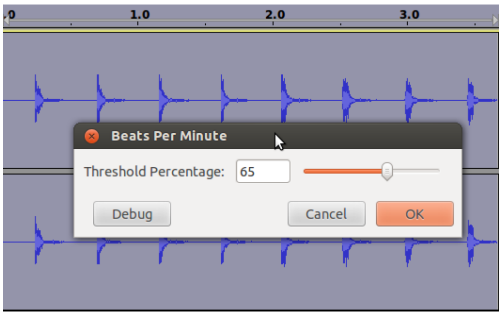

Fig 2. *Selecting threshold*

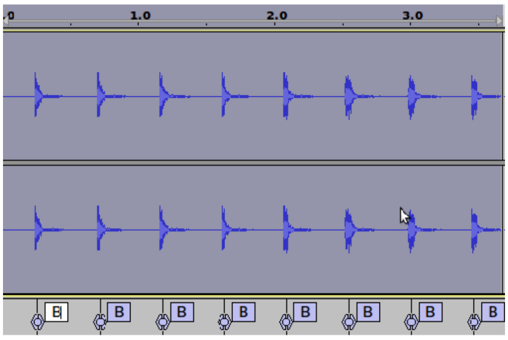

Fig 3. *Locating beats*

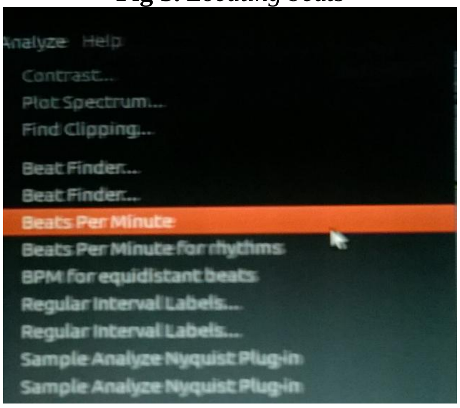

Fig4. *Selecting the plug in*

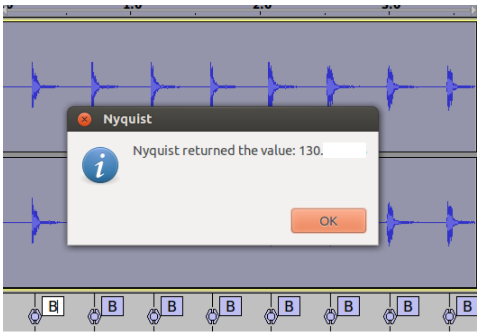

Fig5. *Calculating BPM*

4. ALGORITHM

The flow of the steps that calculates the tempo of selected sound clip goes as shown in figures 2, 3 and 4 respectively.

The algorithm is explained with the help of following flow charts. Fig.6 shows basic flow for calculating time instance. Fig.7 and 8 shows flow for BPM or tempo calculation. These flow charts show the algorithm running behind the steps followed.

Checkval is a variable that is set as 1 when the sample value goes below threshold and set as 0 when the value remains above the threshold. This is used to remove closely located beats that are undesirable and may cause confusion. List is actually a list where the values of samples that are above the threshold are saved, in timeList the time instances of the selected samples are stored in seconds.

length of timeList contains the number of elements in the timeList, var1 is the variable that contains the first element of the timeList that is popped from the list, var2 is the variable that contains the first element of the remaining timeList after popping the first element, diff contains the difference of var1 and var2, dList is the list where all the 'diff' values are stored.

Length of the selected sound clip (in seconds) is saved in a variable named slength, length of dList refers to the number of elements in dList, beats is the variable where number of beats occurring in the selected part of the sound clip is stored.

Tempo is calculated by using basic mathematical logic as,

Tempo= (60*no. of beats) / length of the sound clip

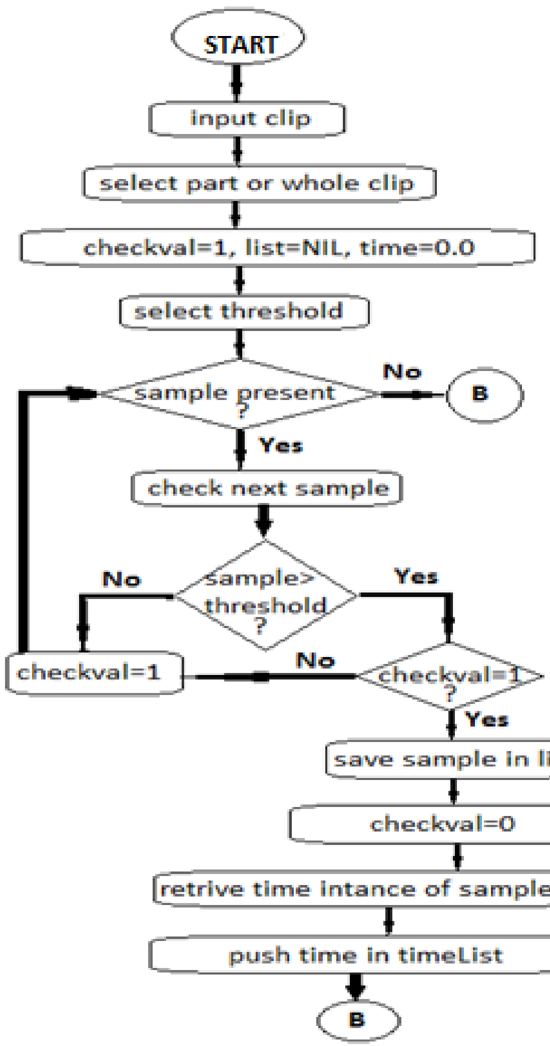

Fig 6. *Time instances in a list*

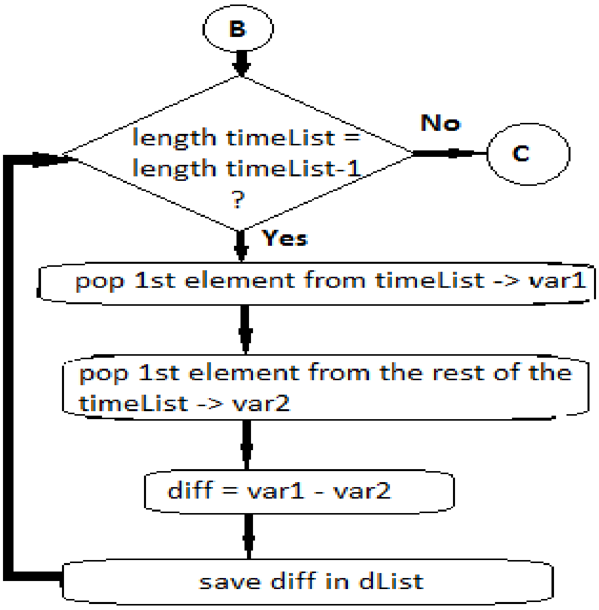

Fig 7. *For calculating number of beats*

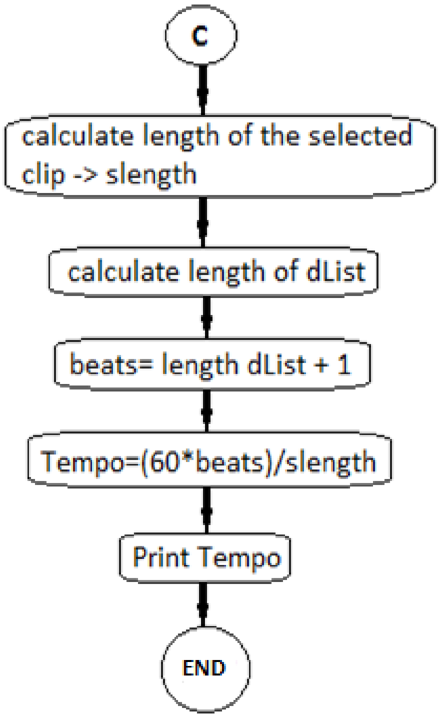

Fig 7. *Calculating Tempo i.e. BPM*

5. Experimental Results

Table 1.

Sound clip	Threshold	Manual BPM	Automatic BPM
Electro analogue bass.wav	65	128	128
Electro-analogue space lead	65	224	224
Clap.wav	65	60	60
Hi-hats.wav	65	16	16
Dub-step lead	55 onwards	52	53
Eight bit ataribassline.wav	35-40	120	72

We have tested our algorithm for range of BPM as slowest as16 BPM to as fastest as 224 BPM as shown in table 1. It has given us desired results for majority of clips. We have tested about 50 rhythmic clips in these pilot experiments. As shown by few samples in Table.1, the proposed algorithm works for most of the rhythmic clips.

We have compared the results of automatic BPM given by our algorithm with manual perception of BPM. Default threshold is set to 65 Db. If the difference between actual BPM and Experimental BPM is less than 5 % then it can be ignored at this initial pilot experiment as human perception can very a little bit as well and can have small errors in calculating BPM manually.

We have also observed few exceptions as shown in the last entry. We noticed that our algorithm does not work for complex rhythm where two or more instruments are played (example is last clip shown in above table.1).

We have to test our algorithm for wide variety of rhythmic clips available across different genres of music. The initial results are encouraging for the sample musical clips.

We are working on improvisation of our algorithm which suits to complex rhythms as well and supports different genre of music.

6. Applications

• **Creating playlists -** We humans, while listening to the songs generally tend to listen to the songs according to our mood swings. We can make a playlist by manually choosing our favourite songs. The tempo estimation technique can help us sort the list of our songs according to the speed. We can thus segregate the songs according to our wish with less effort.

• **Disc Jockeys -** Tempo measurement is of great use to the Disc Jockeys commonly known as DJs. They need to sort out the songs according to the party mood. Merging of the two songs can also be done only if the tempo is known.

• **Regular listeners -** Classify songs according to their mood, like party songs, silent songs etc. It becomes easier to choose the songs according to their mood if the tempo is known.

• **Composers and musicians -** Use tempo measurement for the editing of songs and also for their references.

• **Music Information Retrieval -** Music IR community can use the automatic BPM for efficient content based music retrieval.

7. Conclusions and Future Scope

In the project we have developed two plugins in the open source software named Audacity. One plugin locates the beats in the sound clip. It displays the actual position of human perceived beat within a sound on the waveform.

The other plugin calculates the average Beats Per Minute in a sound clip. The output is given as the measure of Beats Per Minute. When the functionality of the plugin reaches upto the mark, it will be a great contribution to Audacity.

The tempo estimation of complex sound clips with beats having irregular time differences between them, or the sound clips with varying tempo has not yet been successfully done. These kinds of sound clips may probably require more powerful algorithms, which may include frequency band separation.

As an application of this technique, a playlist of songs can be created with songs sorted according to the increasing order of their tempo. With the use of tempo estimation, songs can also be segregated into low tempo, medium tempo and high tempo songs. The creation of such a type of playlist may be included in future scope.

Acknowledgment

We take this opportunity to thank our Head of the Department .Mrs.Madhura Tokekar and our principal Dr.Madhuri Khambete for their valuable support and the necessary facilities provided for the novel research work along with the continuous encouragement.

Our initial work was very well supported by the audacity development team and forum for their active help in understanding the required functions and building the required code. We would like to give thanks to Mr.Steve who helped us in understanding the nyquist coding. We would also like to thank our project partners Sneha Khude and Shravani Pande for their active support.

REFERENCES

[1] Tempo (Beats per minute) http://en.wikipedia.org/wiki/Tempo
[2] Music Information Retrieval (MIR) http://en.wikipedia.org/wiki/Music_information_retrieval
[3] Yu Shiu and C.-C. Jay Kuo Musical Beat Tracking via Kalman Filtering and Noisy Measurements Selection, Ming Hsieh Department of Electrical Engineering and Signal and Image Processing Institute University of Southern California, Los Angeles, CA 90089-2564.
[4] MFCC https://projects.developer.nokia.com/ DSP/wiki/Mel_frequency_cepstral_coefficients
[5] Mikel Gainza and Eugene Coyle,Tempo Detection Using a Hybrid Multiband Approach, IEEE Transactions on audio speech and language processing VOL 19 NO 1 January 2011.
[6] Juan Pablo Bello, Laurent Daudet, SamerAbdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler, Senior Member, IEEEA Tutorial on Onset Detection in Music Signals.
[7] Nyquist Programming http://www.audacity-forum.de/download/edgar/nyquist/nyquist-doc/manual/home.html
[8] Nyquist functions http://www.cs.cmu.edu/~rbd /doc/nyquist/part8.htm
[9] Vedshree Panchwadkar, Shravani Pande and Makarand Velankar A survey on paper on beat tracking, International Journal of Research in Computer and Communication Technology, Vol 2, Issue 10, October- 2013.