Tempo, Beat and Downbeat estimation for Electronic Dance Music

Somesh Ganesh
Center for Music Technology
Georgia Institute of Technology
Atlanta, Georgia 30332
Email: someshg94@gatech.edu

Alexander Lerch
Center for Music Technology
Georgia Institute of Technology
Atlanta, Georgia 30332
Email: alexander.lerch@gatech.edu

***Abstract*—Electronic Dance Music is a family of genres which has recently become very popular. Still in its early stages, there is a lot of scope for MIR (music information retrieval) research in this type of music. In this paper, we look at some research done in these genres and propose an algorithm which tries to utilize the inherent characteristics of this type of music by using some modified correlation functions. We then come up with a combination of a set of novelty functions which works well with the algorithm to perform tasks like tempo, beat and downbeat estimation. This algorithm is evaluated against other configurations of the algorithm and finally implemented using a GUI (graphical user interface) which is available online for others to try. We thus try to pave the way for more complicated research to build intelligent music softwares and systems for Electronic Dance Music.**

I. Introduction

With the increase in affordable electronic music production and performance softwares and technology, there has been an increase in the interest and awareness of Electronic Dance Music.

Electronic Dance Music is an interesting genre of music with a lot of harmonic, rhythmic and spectral content. There is a lot of scope for research in this genre. Tasks like structural segmentation, instrument identification would be crucial to building intelligent music softwares to accompany and augment the performance of electronic music producers, performers and DJs.

Tasks like tempo, beat and downbeat estimation would provide the base for such research and pave the way for more advanced technical research. This paper explores these three tasks. With a strong focus on tempo estimation, we try to identify beat and downbeat locations based on the estimated BPM (beats per minute) value of the track.

We discuss the background, proposed algorithm, methodology and evaluation in the following sections. We finally sum up our findings and present a conclusion with certain recommendations at the end of the paper.

II. Background

Electronic Dance Music (EDM) is a group of different music genres comprising of music made mostly with computers and electronic instruments [1]. Most tracks in these genres are made with an intention to play in combination with other tracks and to dance to. However, some songs may not be intended for this reason specifically [2].

EDM was mostly an underground culture until now [3]. It has now become extremely popular and dominates most of the top charts in the music industry today [4]. Most of the top and trending music today infuse some elements of EDM in their songs.

Let us look at some research focused on EDM. [5] presents an algorithm for the detection of structural boundaries in EDM. The approach includes downbeat detection along with the implementation of some musically informed rules. These rules are used for tempo estimation and beat tracking before performing segmentation. [6] also use the same tempo estimation algorithm in their approach to modeling rhythm similarity in EDM. The tempo is estimated using an autocorrelation function on the onset detection curve to find periodicities in the track. [7] present a unique model to detect downbeats for three specific sub genres of EDM, namely *hardcore*, *jungle* and *drum and bass*. These genres are less melody driven and involve complex rhythmic and drum patterns. Their approach involves the use of SVR (Support Vector Regression) [8] to infer likely downbeat positions. The attempt is to try to train a model on breakbeat patterns and timbres which can find downbeats in these genres. [9] propose that we look at EDM as loop-based music. As noted by [10], they consier loops as an essential element of electronic music. They attempt to model and decompose electronic music using these loops and their activation and deativation over time in a track.

Most EDM tracks follow a steady tempo and rhythmic pattern. In this paper, we try to use some of these characteristics to create our own model to estimate the tempo and subsequently the beats and downbeats of a track. The idea is to try to develop on pre-existing models used before and build on them.

III. Proposed Algorithm

The block diagram for the proposed algorithm is shown in Figure 1. Let us go through the algorithm.

A. Novelty functions

We first generate a set of novelty functions from the audio input. EDM generally has a steady tempo with low frequency elements like kick drums at regular intervals. To utilize this characteristic of the music, we first pass the audio input to a band pass filter which filters out all other frequency components in the music. For this experiment, we have a band pass frequency range of 50 Hz to 400 Hz. The audio input and the filtered signal are then blocked into frames of 1024 samples. We then generate the following novelty functions from the blocked signals (of the audio input and of the filtered signal).

1) RMS from the filtered signal (RMS 1 in Figure 1): The root mean square (RMS) values are computed for each block of the filtered signal. The formula for RMS of a block is given as

$$rms = \sqrt{\frac{\sum_{i=1}^{n} {x_i}^2}{n}} \tag{1}$$

where n = length of the block. The RMS values of the blocked filtered signal would show higher values in the presence of low frequency elements like the kick and bass drums which dominate most of the rhythmic content in this music. Thus, we would mostly get a signal periodic with the duration of a beat. We take a difference of this signal to get more well defined peaks at beat locations.

2) RMS from the audio input (RMS 2 in Figure 1): We also take into account periodicities of elements on the other end of the spectrum. High frequency components of snare drums or melodic components could contribute to the periodicity in the music and hence we also compute the RMS values for the blocked signal of the audio input using equation 1. We compute the difference of this signal too.

3) Spectral centroid from the audio input: EDM has a lot of spectral content which can be made use of for a lot of MIR (music information retrieval) tasks. We compute the spectral centroid of the blocked signal of the audio input. This gives us the an idea of the spectral distribution in a block of the signal. The formula for spectral centroid of a block is given as

$$centroid = \frac{\sum_{k=0}^{N-1} f(k)X(k)}{\sum_{k=0}^{N-1} X(k)} \tag{2}$$

where X(k) = the weighted frequency value of bin number *k* and f(k) = center frequency of that bin. We compute a difference of the spectral centroid values too.

4) Spectral flux from the audio input: Another crucial parameter showing the change in spectral content across the song is the spectral flux. Spectral flux gives us an idea of the spectral difference in the signal between each consecutive frame. The formula for spectral flux of a block is given as

$$flux = \sum_{k=1}^{N/2} (|X_r(k)| - |X_{r-1}(k)|)^2 \tag{3}$$

where X(k) = the weighted frequency value of bin number *k*.

B. Tempo estimation

Feature extraction and computation of novelty functions is followed by tempo estimation. We have three main blocks in this section.

1) Autocorrelation function: Autocorrelation is the correlation of a signal with shifted versions of itself as a function of delay or lag. It represents the similarity between these shifted versions and finds repeating patterns. The formula for autocorrelation is given as

$$r_{xx}(\eta) = \sum_{i=-\infty}^{\infty} x(i).x(i+\eta) \tag{4}$$

where η represents the lag. From the formula, we can deduce that we will obtain higher values of autocorrelation at lag values which correspond to period lengths of the signal. Hence, we take the lag of the maximum autocorrelation value as the period length of the signal.

Now, there can be multiple period lengths for two reasons. The first is can be attributed to the presence of different elements in the song having different period lengths. The second reason is simply because a signal having a period 'P' is also periodic with period length '2P', '3P', etc. To tackle this second problem, we define a tempo range for estimation. Here, we have defined a range of 80 - 160 BPM. If the calculated period length does not correspond to a BPM value in this range, we simply multiply or divide the period length by 2 till it reaches this range. It is important to note that we do not take a lag value that corresponds to a BPM value in this range but take the lag value of the maximum autocorrelation value, whatever BPM value it may represent, and simply multiply and divide it by 2 till it reaches the allowed range.

Each of the four novelty functions computed earlier are passed into this function which gives a period length corresponding to a BPM value within a range of 80 - 160 BPM. Thus, we have four estimates of the tempo after this block.

2) Cross correlation grid function: Cross correlation between two signals is a measure of their similarity which is represented as a function of the displacement of one relative to the other. Its formula is given as

$$r_{xy}(\eta) = \sum_{i=-\infty}^{\infty} x(i).y(i+\eta) \tag{5}$$

where η is the relative displacement.

After the autocorrelation block, we pass the period length obtained and the novelty functions into a cross correlation block. The autocorrelation block should work ideally if the period length is constant with an exact value throughout the track. This may not always be the case. Hence, in this block, we define a 'tolerance range' for the tempo which is 0.9 - 1.1 times the period value obtained from the autocorrelation function. A grid is created for BPM values in this range (for BPM values in steps of 0.125). Each grid contains impulses spaced with respect to the period length of the BPM value. The

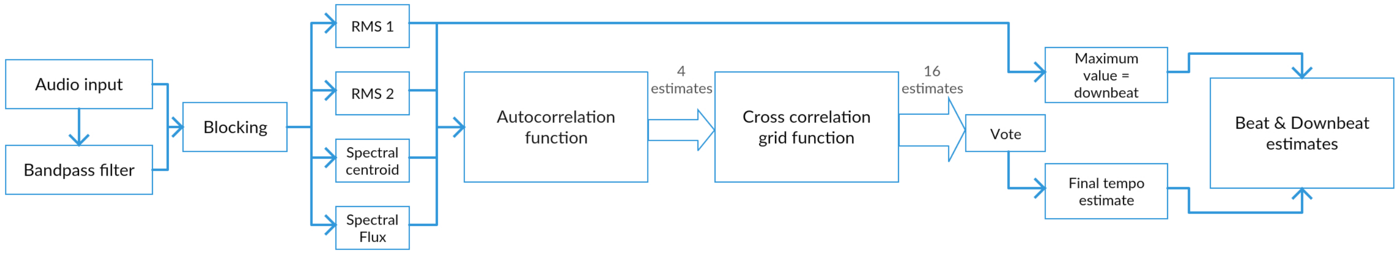

Fig. 1. Block diagram

novelty function passed to this function is cross correlated with all these grids and the BPM with the largest cross correlation value is passed as the output. Note that it is necessary to compensate the amplitude for the number of impulses in the grid. A grid with a higher BPM value will have more impulses and hence its amplitude must be accordingly scaled so that the addition of all the impulses in one grid is a constant value.

After the autocorrelation block, we obtain four period values. All these four period values are passed along with each novelty function in the cross correlation grid function. This will give us a total of sixteen estimates.

3) Voting: We now have sixteen estimates from the cross correlation function which are passed through a voting system. Here, we simply choose the mode of these sixteen values as the final tempo estimate.

C. Beat and Downbeat estimation

Once we have our final tempo estimate, we use this tempo value to estimate the beats and downbeats in the song.

1) Detection of one downbeat: This step involves detecting one downbeat in the song. Our first novelty function, which is the difference of the RMS values of the blocked filtered signal, consists of mainly the low frequency components which would involve the kick drum and bass drum. We can assume these components to have the highest value in the novelty function. Hence, we find the point of highest value in this function and select that as one of the downbeats in the whole track.

2) Estimation of remaining beat locations: Once we find out the location of one downbeat, using the tempo estimate we have (and assuming a 4 beat measure), we can add beats and downbeats on either side of the detected downbeat throughout the track.

Hence, we estimate the tempo, beats and downbeats of the track.

IV. Evaluation and Results

A. Dataset

Two special datasets were made for tempo estimation and key detection specifically for Electronic Dance Music by [11]. A subset of the tempo estimation dataset was used for the evaluation of the proposed algorithm. A total of 106 tracks, each being two minutes long, were used for tempo estimation. Keeping in mind the constraint we mentioned before in the autocorrelation function regarding estimating tempi of multiples of two, the tracks chosen had a BPM value between 80 - 160 BPM. This dataset includes a variety of sub-genres like dubstep, electro house, tech house, trance, hard-dance, electronica, etc.

B. Tempo evaluation

For tempo evaluation, a tempo difference (between the ground truth and estimated tempo) between +1 and -1 is considered as a correct estimate. This might seem a tight constraint for evaluation of tempo, but keeping in mind that the final goal is to estimate the beats and downbeats in a track using this estimated tempo, this restriction would make much more sense. A larger constraint might potentially ruin the beat and downbeat estimation as the error would be cumulative and shift the location of each estimated beat further from the original location.

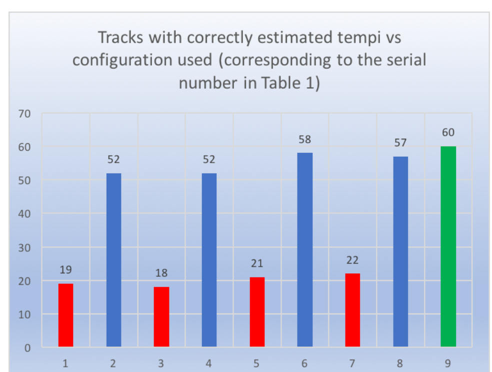

Fig. 2. Tempo evaluation

Nine separate configurations have been compared here. The first four are the outputs of the autocorrelation function corresponding to each novelty function. The next four configurations are the outputs of the cross correlation grid function where the same novelty function is used for autocorrelation and cross correlation. Finally, the last configuration is the proposed algorithm where we have a total of sixteen estimates which are passed through the voting system to obtain a final estimate.

Sr. No.	Novelty function	Method	Percentage of correctly estimated tempi
1	Difference of RMS (of filtered signal)	Autocorrelation	17.9
2	Difference of RMS (of filtered signal)	Autocorrelation + Grid Cross correlation	49
3	Difference of RMS (of audio input)	Autocorrelation	16.9
4	Difference of RMS (of audio input)	Autocorrelation + Grid Cross correlation	49
5	Spectral flux (of audio input)	Autocorrelation	19.8
6	Spectral flux (of audio input)	Autocorrelation + Grid Cross correlation	54.7
7	Difference of spectral centroid (of audio input)	Autocorrelation	20.7
8	Difference of spectral centroid (of audio input)	Autocorrelation + Grid Cross correlation	53.7
9	Combination of all 4 novelty functions	Autocorrelation + Grid Cross correlation	56.6

TABLE I
RESULTS FOR TEMPO ESTIMATION

Figure 2 shows the number of correctly estimated tempi for the dataset used in a bar graph. Here, the red configurations show the autocorrelation results, blue show cross correlation results and the green one shows the result of the combination of all 4 novelty functions after cross correlation and voting. The accuracy (in percentage) has been tabulated in Table I.

C. Beat and Downbeat evaluation

For the evaluation of the estimated beats and downbeats, a simple user interface was designed. This GUI (graphical user interface) lets the user pick any audio file which is displayed in blue in the plot in Figure 3. Tempo, beat and downbeat estimation is performed on the track and all the beat locations are shown in red. An auto looper is implemented for the user to understand the application of the algorithm in real world systems. The user can choose the downbeat he/she wants to loop, the number of beats he/she wants to loop it by and the number of times the loop should run. This GUI is available online[1]. This GUI helps one evaluate the beat and downbeat locations on any track when one uses the auto looper.

V. DISCUSSION

From the results shown in Figure 2 and Table I, one can deduce that the autocorrelation function by itself is not extremely effective at producing a good estimate of the tempo of a track in this genre. This may be because all the periodic events in the track may not take place at exactly each beat location. Very few correct estimates are obtained for this

[1]https://github.com/SomeshGanesh94/Structural-Segmentation-of-Electronic-Dance-Music/tree/master/GUI%20Demo

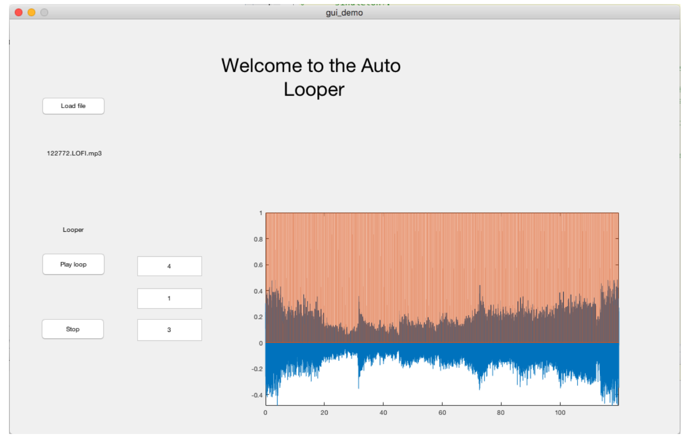

Fig. 3. Auto looper GUI

method. However, it is clear from the results that the cross correlation grid function improves the accuracy significantly. The number of correct estimates increases by more than twice the amount when compared to only autocorrelation. This is because it solves the problem faced by the autocorrelation function by considering a range of tempo values around the period computed by the autocorrelation function. We finally observe that the combination of all 4 novelty functions and a final estimate obtained by voting from the 16 estimates produces the best result amongst the tested configurations. Hence, we can maybe say that this combination of features works well along with this algorithm due to the contribution of certain characteristics of each of the novelty functions to predict the periodicity in the music.

The accuracy of the estimation of the beats and downbeats is dependent on the accuracy of the tempo estimation. Hence, the auto looper GUI will provide a good platform to test the accuracy of the estimated beats and downbeats.

VI. RECOMMENDATIONS AND FUTURE WORK

The following are some recommendations that can be worked upon for improvement in the algorithm.

- Different sets of features at different frequency ranges can be explored for improvement of the tempo estimation and downbeat detection algorithm, for example, difference in the harmonic content in a signal
- Different voting systems can be explored instead of simply taking the mode of the obtained estimates
- Once a downbeat in a track is estimated, the subsequent beats and downbeats can be updated using an adaptive approach where the beat/downbeat is assumed to be located within a range of time and a feature such as the highest RMS value or correlation value can decide the beat location
- A model using a supervised learning algorithm can be designed which learns and predicts locations of beats and downbeats

VII. CONCLUSION

In this paper, we go through some of the research carried out on Electronic Dance Music and propose an algorithm to estimate the tempo, beats and downbeats in a track belonging to this group of genres. We tried to take advantage of certain characteristics of this style of music and tested 9 configurations of an algorithm using autocorrelation and cross correlation functions and finally came up with a combination that outperforms any of the other novelty functions separately. We can thus carry on to build this algorithm and further do research and study topics like structural segmentation, instrument detection, etc. for Electronic Dance Music.

ACKNOWLEDGMENT

I would like to thank my advisor, Dr. Alexander Lerch for his constant guidance and support throughout this semester long project. I also want to thank my colleagues in the music informatics lab at the Georgia Tech Center for Music Technology for their invaluable support and input.

REFERENCES

[1] K. McLeod, "Genres, Subgenres, Sub-Subgenres and More: Musical and Social Differentiation Within Electronic/Dance Music Communities," *Journal of Popular Music Studies*, vol. 13, no. 1, pp. 59–75, Jan. 2001.

[2] M. J. Butler, *Unlocking the groove: Rhythm, meter, and musical design in electronic dance music*. Indiana University Press, 2006.

[3] ""You Better Work!" Underground Dance Music in New York City."

[4] Z. O. Greenburg, "The World's Highest-Paid DJs 2012."

[5] B. Rocha, A. K Honingh, and N. Bogaards, "Segmentation and Timbre Similarity in Electronic Dance Music," in *ResearchGate*.

[6] M. Panteli, N. Bogaards, and A. K Honingh, "Modeling Rhythm Similarity for Electronic Dance Music," 2014, pp. 537–542.

[7] J. Hockman, M. E. Davies, and I. Fujinaga, "One in the Jungle: Downbeat Detection in Hardcore, Jungle, and Drum and Bass." in *ISMIR*, 2012, pp. 169–174.

[8] T. Jehan, "Creating music by listening," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.

[9] C. Dittmar, M. Mller, P. Lpez-Serrano, and J. Driedger, "Towards Modeling and Decomposing Loop-Based Electronic Music," Jan. 2016, pp. 502–508.

[10] N. Collins, M. Schedel, and S. Wilson, *Electronic music*. Cambridge University Press, 2013.

[11] P. Knees, A. Faraldo, P. Herrera, R. Vogl, S. Bock, F. Horschlager, and M. Le Goff, "Two Data Sets for Tempo Estimation and Key Detection in Electronic Dance Music Annotated from User Corrections," 2015, pp. 364–370.