GENERATING A NEW PERCUSSIVE SECTION FOR MUSIC

First author
Mojtaba Heydari
mheydari@ur.rochester.edu

ABSTRACT

One of the main weak points of many automatic music generation systems is their over-focus on harmonic parts and paying less attention and in most of cases, omitting rhythmic sections of music. Whereas, non-harmonic (percussive) elements of music are likely to have as important role in forming the general structure of music as harmonic parts. In this paper, a new rhythmic section generator is proposed that first detects music tempo leveraging dynamic programing and then creates multiline dynamic drum parts for the music track by considering some music rules and utilizing random vectors. It also allows the user to select between different genres of music and then provides a new percussive section for music each time. Generated music is corresponding to the selected genre. Three genres including pop, metal, and trance are provided that are extendable. For each genre, different instrument types and different music rules are applied.

INTRODUCTION

Music has an important role in modern human life. Therefore, by growing technology, different tools to generate music, manipulate it, analyzing it, storing and sharing it and etc are developed. Two of the most interesting and popular fields related to music research are music generation and music remixing tools. However, dealing with music is not as easy as it seems and due to many complexities and constraints of music such as necessity of following music grammar and deep relation of it with time factor and also requirements of a decent music such as creativity and transferring meaningful emotions, make music creation and manipulation a demanding subject of research. The flow of this paper is as follows: first, we introduce some of the related works related to music generation and music composition and then we propose the approach that we utilized to generate and drums part and then we talk about the future works, possibilities, advantages, and challenges.

1. RELATED WORKS

There is a great amount of research works dealing with music generation that most of them are corresponding to melodic and harmonic music generation. Trochidis et al [1] uses an N-gram model and a group of short pre-generated and manually annotated percussive music to generate a sequence of percussive music using Markov chains. In their work, the percussive segments are prepared, and their method only attaches them together and the tempo has three modes including fast, normal and slow. Uhle et al [2] propose a comprehensive method including detection and classification of percussive events, segmentation, quantization of detected events, Tatum grade estimation, estimation of time signature and detection of recurrent patterns of the percussive events. However, their work is not focused on generation music parts and more is into detection and estimation of parameters. Ellis et al [3] hired a collection of 100 drum beats and by detecting their tempo, downbeat, and pattern period estimation and utilizing principal component analysis, extracted a set of basis “patterns” that can be combined to give approximations and interpolations of all the examples. Here in this work, we propose a method that in addition to using tempo, onset, and beat detection, it also utilizes an activate de-activate mode for recognizing break-fill parts and finally generates drums line for the music piece corresponding to events and with regards to the chosen genre of music.

2. PROPOSED METHOD

Given that the rhythmic part of music is very delicate in terms of time and deals with time in milliseconds scale and is usually repetitive in very short time intervals with high dynamics, in order to generate an appropriate drums part for a piece of music, at first stage it is essential to create a practical, precise and flexible event detection system including onsets, beats, tempo detection and for aligning it more with music occasions, determining if the beat is activated or inactivated in each time segment. After ensuring that the rhythmic features are extracted precisely, it is time to synthesize drum loops that fit the music features including music genre, emotion, dynamics and etc. So, the first segment of the method is related to music events detection. Then we will go into synthesizing mode. The total block diagram of the proposed method is demonstrated in the figure 1.

2.1 Beat tracking and tempo detection

As mentioned before, one of the most important parts of percussive segment generation for a song is onset and beat detection. There are different methods that are proposed for such a goal. Some of them are intuitive and also easy to implement such as energy threshold base. One the

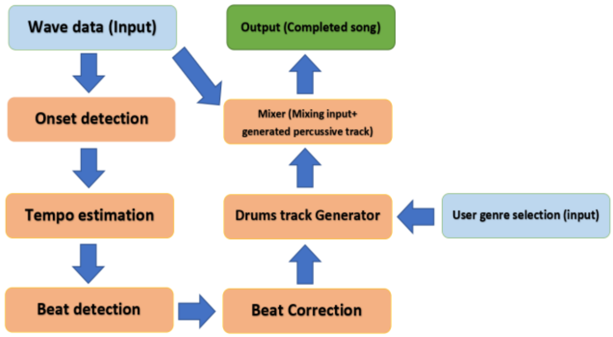

Figure 1. Total block diagram of the proposed method.

other hand, some others are more exhaustive and require a great deal of training time and large annotated data such as hiring neural networks. Given that beat tracking through dynamic programming and utilizing onsets that are extracted from frequency spectrum is an unsupervised efficient and fast way of beat tracking, so in this part davies-beat-tracking method [4] is utilized that follows the traditional dynamic programming method that is proposed at [5]. Also, for making the method more reliable and efficient, we added a bunch of optimizations and corrections to it that we describe some of them in this part. In order to utilize dynamic programming, a short-time Fourier transformation of the time domain signal has been taken and then the spectral flux of signal is calculated. Then according to the dynamic programming method, the optimum path and the best sequence of beats are chosen. One of the other modules that could be very useful in drums track generation is obtaining some knowledge about music tempo. In dynamic programming, an approximation of tempo is utilized. However, the tempo itself can be calculated after beat tracking. In most of the music pieces, the tempo is constant but still in many cases, it is challenging to detect the correct tempo. For tempo detection, again different methods have been proposed so far. besides neural networks that require a great deal of training data with annotations, there are many approaches that are proposed. From correlation to applying different comb filters to the detected onsets graph and shifting it through time to find the comb with the maximum number of onsets inside it. The disadvantage of applying comb filter banks is that it is so sensitive to time shifts. Another method is subtracting each onset from the previous one and take the average of them to gain the mean tempo. Due to noise in onset detection and presence of incorrect onsets that are quite close together and the probability of incorrect beat detection in some extremely crowded and muddy or weak parts of music, this method usually leads to unprecise tempo detection. We propose using the mode of detected tempos from different pair of beats, instead of using mean of them. So that we choose some pairs of random beats next together from different parts of music piece and then calculate their interval and corresponding tempo to that interval and after many different repeats, calculate the mode of them. For evaluation of tempo detection, some metrics such as MIREX-6 could be utilized. However, in this project due to presence of different genres and styles, we utilized 12 different music pieces from different music kinds and hired an online tempo detection platform that asks users to tap a key in beat times [6]. the results are provided in the table 1. Also, a part of the code that uses a random pair of samples to detect tempo is provided below:

```
distance=zeros(100,1);
b=round(rand(100,1)*length(a));
 for i=1:100
    if b(i)>1
    distance(i)= round(60/ (a(b(i),1)-
a(b(i)-1,1)));
    end
 end
 Tempo=mode(distance);
```

Where b corresponds to chosen random samples and a is an array that each element of that is related to one of the detected onsets.

Utilizing mean	Utilizing mode
%66.6	%91.6

Table 1. The table of precisions of detected tempo using the average of simple beats and proposed usage of the mode of random choices of detected beats.

Given that the final goal is to create a drums section for music, it is critical to have the correct number of beats in the correct positions. Otherwise, the could be lots of incorrect hits of different drums elements that ruin the music.

$$S(B) := \sum_{\ell=1}^{L} \Delta(b_\ell) + \lambda \sum_{\ell=2}^{L} P_\delta(b_\ell - b_{\ell-1}) \qquad (1)$$

Eqn (1) is comprised by two parts and for each part an error may occur. For the first part, it may generate a redundant beat quite close to a real one (when there is very high spectral flux strength close to a real one, it bypasses the effect of the second part). It does not only add for instance a single kick or snare hit but it also adds a bunch of other hits in downbeats of that and its divisions as the beat's tail. In order to cope with that, we propose utilizing a quarter of the tempo as the criteria. So that in the cases that there are two beats that are closer than one-quarter interval of expected tempo, we check the interval of each of them with the other closest neighbor of them and then keep the one with a closer interval with the neighbor to tempo interval and remove the other one.(check out lines 39-48 of synth_onset6.m). Furthermore, the second sentence may add an additional error in beat estimation. It happens when there is a place that is aligning with estimation places but there should not be a beat there, because there is a silence. In order to cope with that, we calculate energy amount around that beat and if it is less than a threshold, we remove the beat. (check out lines 32-37 of synth_onset6.m)

2.2 Music Generation section

After obtaining correct places of beats and starting points, it is time to generate percussive parts for given music piece. In order to extend the job, we provide the user with three different music genres to choose between. So that by choosing each of them, the system uses different music elements and different algorithms to create a drum part corresponding to chosen music genre. In order to have more diversity of generated drum loops, we stayed away from generating whole drum loop as a segment for each part. Rather, we utilized the generation method for every single line of the drum part. In other words, we hired our method with different sets of rules for kick part as well as hiring it with different sets of rules for snare, hi-hat and so one. The set of roles are as follows:

1- **Music genre related roles:** The rules that are hired for each of the elements e.g. drums, snares, and etc, are related to music genre. For instance, the probability of presence of kick for metal genre is high in each beat and down beat and it is much higher that that of pop music in which basically kick appears in the starting point of even beats. Furthermore, the method choses different music instruments regarding different genres that a user defines. (check out generator.m)

2- **Duration of each pattern:** As mentioned before, in order to have more diversity of created drum section for each music piece, we don't create different drum loops utilizing random functions and applied music roles and concatenate them together. Rather, we defined a function that utilizes some weighted random variables to give different standard lengths e.g. 4,6,8 as the number of the repetition for each section of drums(conditioner). So that once the previous duration of an element is finished, it runs the function to generate new standard repetition number for new created pattern of the element. Different duration generated by our system is exhibited in the figure 2.

3- **Music flow roles:** In order to reach to a comprehensive and natural drums track, considering the other content of music is important. We do it partially by estimating the onsets, beats and tempo. However, it is not the all thing that could be done. So here we also consider the song's energy density. So that for the parts with very low energy density, we shut down the rhythm generation. Moreover, for the parts with very high energy, the probability of the presences of the elements could be higher.

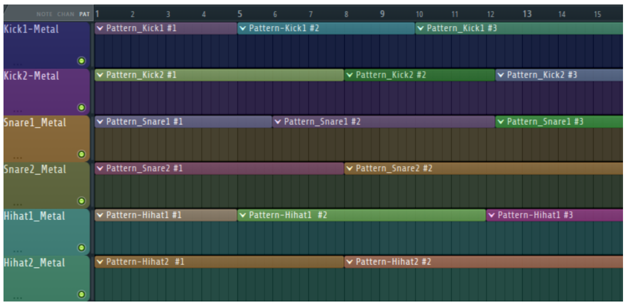

Figure 2. different random durations of each line of the drums track. There are 10 lines hired in total but only 6 of them is shown here.

3. CONCLUSIONS AND FUTURE WORKS

In this work we have proposed a method that first detects music rhythmic characteristics such as tempo, beats and etc then utilizes some random vectors plus music generation roles to create drums part for music. It also could be utilized as a part of music remix generation by using it along with a sound source separation system. So that at first music original drum part could be separated and removed and then this method could be used for creating new drum loops with different genre settings to add the flavor of that genre to the original music.

For a wide range of music genres, preliminary subjective tests show promising music improvisation outcome and generation of creative drums section.

Next, we will extend our research on mentioned fields:

- Adding more music information retrieval methods such as music meter detection for better music generation
- Utilizing other approaches for detection of inactive beats.
- Extracting more cues such as music events could be so helpful in deciding loops' length
- Considering energy level in the neighborhood of each beat as a key factor for deciding the amount of density of generated hits (peak chorus-bridges-verses)
- Considering velocities and dynamics of the song in adding each element

4. REFERENCES

[1] K.Trochidis, C.Guedes, A. Anantapadmanabhan, A.Klaric:"CAMEL: CARNATIC PERCUSSION MUSIC GENERATION USING N-GRAM MODELS," *Sound and Music Computing conference*, Hamburg, August , 2016.

[2] C. Uhle, C. Dittmar: "Generation of Musical Scores of Percussive Un-Pitched Instruments from Automatically Detected Events," *116th Audio Engineering society Convention*, Berlin. 2004.

[3] D. Ellis, J. Arroyo: "EIGENRHYTHMS: DRUM PATTERN BASIS SETS FOR CLASSIFICATION AND GENERATION" *ISMIR 2004: 5th International Conference on Music Information Retrieval: Proceedings:* Universitat Pompeu Fabra, October 10-14, 2004

[4] M. E. P. Davies and M. D. Plumbley, "Context-Dependent Beat Tracking of Musical Audio," *IEEE Transactions on Audio, Speech and Language Processing, vol. 15, no. 3, pp.* 1009–1020, 2007.

[5] Ellis, Daniel P.W. "Beat Tracking by Dynamic Programming," Retrieved: 5/2/2017 'https:// www.ee.columbia.edu/~dpwe/pubs/Ellis07-beattrack.pdf (2007)

[6] http://www.beatsperminuteonline.com/