AN ALGORITHM FOR CREATING GROOVE TEMPLATES FOR APPLICATION TO MIDI DRUM DATA

Jeremy David Hassett

University of Rochester
Department of Electrical Engineering

ABSTRACT

A method of characterizing the timing and amplitude tendencies of percussive onsets in drum audio tracks is introduced. The method uses a variety of existing high performance algorithms for onset detection, tempo estimation, and source separation along with some original modifications to accomplish the desired task. An emphasis is placed on drum audio track for popular music, in which the source instruments originate from a standard drum kit. The timing and amplitude tendencies of the audio are modelled with normal distribution functions for each beat subdivision and then used to generate timing and velocity shifts for programmed MIDI drum tracks.

1. INTRODUCTION

Drum beats for popular music are often generated programmatically using Digital Audio Workstations (DAWs) in order to avoid the time consuming and costly alternative of the drum recording process. This solution often produces very different results than those provided by a human drummer, as programmed drum patterns typically exhibit ideal timing and invariant amplitude among notes. Human drummers provide much more dynamic timing and note velocity variations, which result in more "natural" sounding drum tracks [1]. These tendencies are often unique to individual drummers as well as different styles of music. By investigating the reoccurring patterns of timing and velocity in human performances, effective groove templates are realized and applied to programmed drum tracks to achieve a more realistic synthesized drum sound.

2. METHOD

Prior to beginning the implementation process, some essential requirements for the groove extraction method were laid out. The first main requirement is that the groove extraction method be able to accurately detect as many percussive musical onsets as possible with few false positives. The utility of the onset detection functionality is that it allows the timing and velocity patterns of a drum pattern to be analyzed. The second requirement is that the tempo of the audio be accurately estimated in order to provide a "reference" grid for the timing variations of actual onsets. This "reference" grid will consist of perfectly evenly spaced markers that correspond to the overall tempo of the drum pattern. An example of the application of a reference grid along with actual note onsets is shown in Fig. 1. It should be noted that this method assumes that the overall tempo is constant over the entire audio signal being analyzed.

2.1. Spectral-based onset detection

The onset detection functionality was implemented using a method inspired by the spectral flux method outlined in [2]. Because this method utilizes the spectral features of an audio signal rather than its time domain properties, it proves more capable of detecting low energy hi-hat onsets (and other low energy onsets) along with the more obvious high-energy onsets of the kick drum and snare drum attacks. In order to test the performance of this onset detection method, a small sample of drum audio tracks was generated using the software program Steinberg Groove Agent 4. One advantage of using this software for initial testing purposes is that the software is capable of generating MIDI information that can be used to verify onset locations, tempo, and velocity. An additional advantage is that the software utilizes recorded drum audio samples and provides drum patterns closer to human performances than many other rigid artificial-sounding drum machines [3].

During the initial testing phase, the spectral flux onset detection method performed very well for high-energy onsets but missed many lower-energy onsets. In order to rectify this issue, the dB spectrogram was used instead of the linear spectrogram. This serves to compress the high-energy transients and allow for more

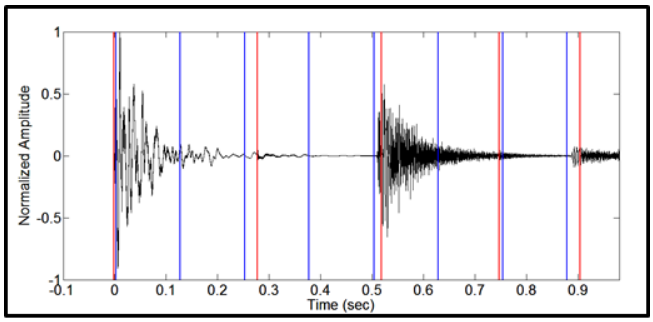

Figure 1 shows the evenly spaced "reference" grid in blue and the actual detected onsets in red.

of the low-energy transients to be selected through peak-picking. The threshold value was set to 0.1 for normalized onset strength curves. This value was found experimentally to provide a minimum number of false negatives, although many false positives were detected. In order to eliminate the high number of a false positives which often occur successively following an actual onset, a minimum onset spacing M was determined. This spacing is used to eliminate the next M frames from being possible onsets once an onset is detected. The value of M was found experimentally to be around 5-6 frames.

2.2. Tempo estimation using spectral product

The second step in implementing the groove template extraction method is to accurately estimate the tempo of a drum pattern. This is accomplished using the spectral product method outlined in [4]. The spectral product method takes advantage of the fact that the Fourier transform of a periodic onset strength curve will display peaks at multiples of the fundamental onset frequency, which typically corresponds to the tempo of the pattern. This method generates very strong peaks at tempo locations through the multiplication of a portion of the onset strength spectrum with compressed multiples of this spectrum. The spectral product method can be described by the equation:

$$S\left(e^{j\omega_n}\right) = \prod_{l=1}^{M} \left|X\left(e^{jl\omega_n}\right)\right|^2, \text{ for } \omega_n < \frac{\pi}{M} \quad (1)$$

where S is the spectrum used to search for tempo peaks. As suggested in [4], M was chosen as 6 and the tempo search was conducted in the range of 5/6 to 5 Hz, which corresponds to a beat rate in the range of 50 to 300 bpm. The tempo was found by using the dB spectrum of S to find the maximum peak and then using quadratic interpolation to locate the exact peak location.

Track #	Actual Tempo	Estimated Tempo
1	131	130.53
2	120	120.07
3	131	131.23
4	120	120.04
5	120	60.04
6	120	119.99
7	100	100.01
8	140	138.69
9	120	120.25
10	120	120.06

Figure 2 shows the tempo detection results for several drum audio tracks.

Initial results of the tempo extraction algorithm demonstrated very accurate tempo estimation with the exception of a tendency toward doubling the actual tempo. In order to improve the performance, the onset strength curve described above was computed by summing over only the bottom 1/6th of the frequencies in the spectrogram. Because the erroneous tempo estimations typically result from the high frequency of hi-hat and cymbal onsets, using only the low spectral frequencies to generate the onset strength curve results in fewer cases of tempo doubling during estimation. The results of the tempo tracking algorithm for 10 different drum pattern audio tracks is shown below in Fig. 2.

2.3. Generation of a "reference" grid signal

Estimating the tempo gives the spacing between consecutive beats but not the actual location or phase of these beats. In order to find this phase, a comb signal containing impulses at possible beat locations was cross-correlated with the onset strength curve of the audio track. The lag providing the maximum cross-correlation was used to shift the comb of impulses to the correct beat locations [6]. The comb can then be sub-divided to create a grid of 8th or 16th notes.

The next step is to generate a higher frequency grid signal using the beat signal. This was accomplished by simply inserting extra grid markers between the beat locations, effectively subdividing each beat into smaller portions to achieve an appropriate resolution. 16th note resolution was chosen as the default, which corresponds to four subdivisions for each beat. It should be noted that many other resolutions, including those based on triplet patterns, are equally possible. The result after dividing down the beat signal is a suitable grid of rigidly spaced

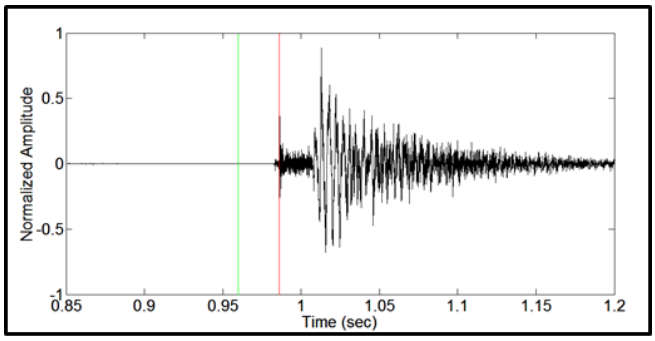

Figure 3 shows the original detected onset in green and the improved onset detection result in red. The shaded area represents the search window for the energy-based onset detection method.

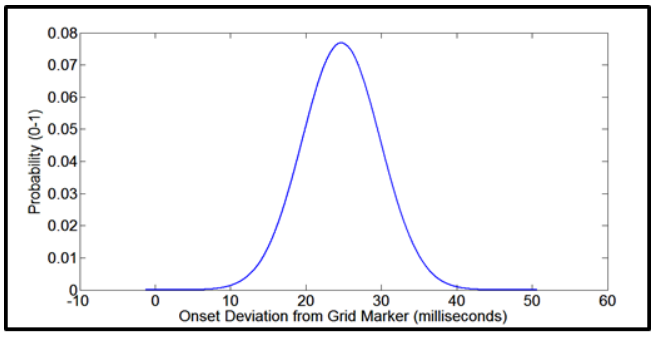

Figure 4 shows the probability of onsets occurring at times relative to their closest grid marker for a particular beat subdivision.

note locations, each of which can serve as reference markers for actual onset locations.

2.4. Refinements to onset and grid detection method

Prior to using the reference grid to map the timing tendencies of onset within the audio file, some additional refinements were made to both the onset locations and "reference" grid locations. In practice, it was found that the detected onsets were slightly offset from the actual obvious energy changes in the audio signal. It is suspected that this is due to the onset timing resolution, which is inherently limited by the hop size of the short-time Fourier transform. In order to achieve improved resolution, a short window of the audio signal was selected around each onset. Within this window, energy-based onset detection was performed using a much reduced window and hop size. The result gives a more accurate representation of the onset location. The implementation of this method for a single onset is shown below in Fig. 3. A similar method was used to refine the grid marker locations. The grid markers were allowed to shift a small distance from their original location and the highest cross-correlation value was used to shift the markers. It should be noted that the rigid spacing between the markers was still held constant. The result of this refined onset and grid marker detection method is displayed in Fig. 1.

2.5. Modelling the timing tendencies

In order to classify the timing properties of different onsets in the audio signal, the note onsets were first grouped together based on their closest reference grid marker. It was found experimentally that the timing tendencies of onsets are generally correlated with their grid location. For example, if an audio signal is divided up into a sixteenth note grid there will exist four subdivisions for each beat. It is conventional in musical scores to label these subdivision as "1", "e", "+," and "a," where the "1" represents the downbeat number. The average deviation of a given onset from its closest reference grid marker was found to roughly align with its specific beat subdivision. Therefore, the onset deviations for each of the four beat subdivisions was recorded.

In order to model the timing variations measured above, the mean and standard deviation of an onsets' deviation from its respective grid marker was computed for each beat subdivision. These parameters are then used to generate a normal distribution function characterizing the probability of an onset occurring at a certain relative to its grid marker. This method models the actual timing variations of human drummers and is used to shift the notes of computer programmed MIDI drum tracks [1]. An example of an onset timing variation function is given in Fig. 4.

2.6. Source separation using cepstral characteristics

The next step in implementing the groove template extraction method is to track the amplitude tendencies of the audio file with respect to beat subdivision. The simplest method of achieving this functionality would be to simply compute the energy at each onset location and correlate this energy to its closest beat subdivision. In practice, however, this provides unsatisfactory results. Because multiple instrument attacks will occur at a given beat subdivision, the distributions at these subdivisions will exhibit very large standard deviations. For example, a given beat subdivision may be correlated with many low-energy hi-hat attacks as well as a few very high-energy snare attacks. If these energies were compared directly, it would appear as if very large accents had occurred in the audio file at snare attack locations. In order to rectify this problem, a simple method of source

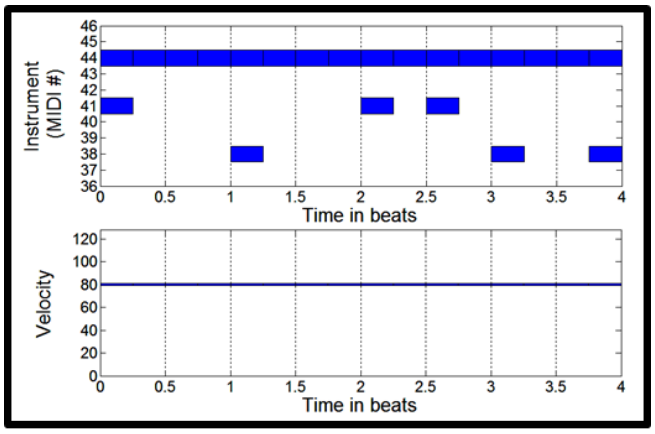

Figure 5 shows the original MIDI track with perfect beat locations and even velocity throughout.

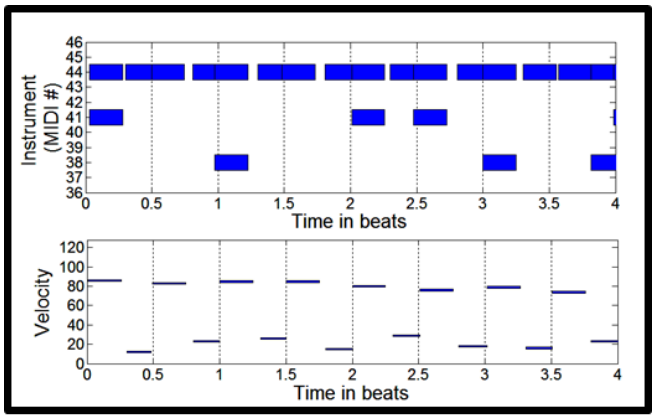

Figure 6 shows the resulting MIDI track after applying a generated groove template.

separation is implemented and the relative energy of onsets with respect to the mean source onset energy are used to generate the energy distributions.

A simple and effective approach to performing source separation is to use the Mel-frequency cepstral coefficients (MFCCs) of a signal to group signals into various clusters. The idea of this method is that the MFCCs of a signal provide a perception-based description of the timbre of the signal, and thus provide a method of grouping sources with similar timbres. In order to calculate the MFCCs of a signal, a bank of triangular filters equally spaced in the Mel-frequency domain is applied to the power spectrum of the signal and the power is summed within each filter band to generate a new signal. The triangle filters are evenly spaced in the Mel-frequency domain. The conversion from linear to Mel-frequencies is described by the equation:

$$M(f) = \ln(1 + \frac{f}{700}) \quad (2)$$

The discrete cosine transform is then performed on the resultant signal to provide the Mel-frequency cepstral coefficients [7], [8].

The actual grouping of signals is accomplished using a hierarchical clustering algorithm which groups signals based on the Euclidean distance between their MFCCs. This algorithm allows for the specification of a maximum number of clusters to divide the data. For the purposes of tracking the amplitude tendencies of the drum audio track, it is more important that each cluster contain no two onsets from different source instruments than to provide an exact clustering of all the sources. Therefore, the maximum number of clusters is set slightly higher than the estimated number of sources for a drum audio track.

2.7. Modelling the amplitude tendencies

In order to model the amplitude tendencies of the audio track, the log scale root-mean-square (RMS) value for a given onset is compared to the mean log scale RMS of its associated cluster. The distribution of relative amplitudes at each beat subdivision is once again modelled with a normal distribution function. The log scale relative RMS distributions are then converted to MIDI velocity using a linear transformation. Although the interpretation of MIDI velocity can vary drastically amongst artificial synthesizers, the transformation between velocity and log RMS amplitude can be approximated with some type of linear function for most synthesizers [5].

3. RESULTS AND CONCLUSIONS

The results of applying the generated groove templates to MIDI data proved successful in accomplishing the desired functionality. The MIDI notes were shifted in time in a way similar to that of a reference audio track. The MIDI velocities were also altered to realize the on-beat accents and off-beat attenuation observed in reference audio tracks. Examples of original and resultant MIDI data after applying a generated groove template is shown in Fig.5 and Fig.6, respectively. In practical implementations, the only alteration to the outlined method is to allow the user to control a few compression parameters. Adding some compression to the standard deviations allows improved performance as small errors in the onset, beat, and amplitude detection methods result in being artificially high standard deviations. This causes the MIDI tracks to exhibit higher amounts of variation between notes than typically observed in human performed drum tracks. A small adjustment to these parameters results in realistic sounding MIDI performances.

4. REFERENCES

[1] Tidemann, Axel, and Yiannis Demiris. "Imitating the Groove: Making Drum Machines More Human." *Proceeding KI '08 Proceedings of the 31st Annual German Conference on Advances in Artificial Intelligence* (2008): 144-51. Web.

[2] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. "A tutorial on onset detection in music signals." IEEE Trans. on Speech and Audio Processing, vol. 13, no. 5, (2005).

[3] "Why Groove Agent?" Steinberg Groove Agent. Steinberg Media Technologies, n.d., Web. <http://www.steinberg.net/en/products/vst/groove_agent/why_groove_agent.html>.

[4] Alonso, M., B. David, and G. Richard. "A Study of Tempo Tracking Algorithms from Polyphonic Music Signals." Ecole Nationale Superieure Des Telecommunications (ENST), 1 Apr. 2003. Web.

[5] Dannenberg, Roger B. "Interpretation of MIDI Velocity." School of Computer Science, Carnegie Mellon University.

[6] Duan, Zhiyao. "Lecture 6: Rhythm Analysis." ECE 272/472 (AME 272, TEE 272) – Audio Signal Processing (2015).

[7] "Mel Frequency Cepstral Coefficient (MFCC) Tutorial." *Practical Cryptography*. N.p., n.d. Web. 28 Apr. (2015).

[8] Brent, William. "Cepstral Analysis Tools for Percussive Timbre Identification." Department of Music and Center for Research in the Performing Arts, University of California, San Diego.

[9] Toiviainen, Petri, and Tuomas Eerola. "MIDI Toolbox for Matlab." Department of Music, University of Jyväskylä, Finland.