Extracting Musical Rhythms from Repetitive Videos

Jeongmin Liu
Department of Electrical Engineering
Pohang University of Science and Technology
Student ID: 20130203

I. Introduction

There are algorithms to make videos that suitable for music. But nobody has researched algorithms to do its inverse process because most videos don't have any musically meaningful regularities. But in this work, I focus on the fact that some abstract and repetitive videos have a rough regularity. That loop videos are created in order to be laid in the background, so situations which require that kind of videos may require appropriate background music also. So I extract a rough temporal regularity from that videos, approximate the regularity to a typical 4/4 rhythm, and insert a simple rhythm instrument pattern.

II. Related Works

There's no related works.

III. Algorithm Overview

The input videos should have abstract illustration elements, and also have some moments that the elements suddenly change. That moments can be classified into two kinds: sudden intensity changes, sudden movements(location change). I call them the sudden events, and both or one of them should appear roughly regularly in the videos.

The block diagram of the algorithm is illustrated in Fig. 1. I explain how to extract the sudden events $E(n)$ from the input video in Section IV. In Section V, the algorithm to create the rhythm pattern vector $\mathbf{B}$ from $E(n)$ is explained. $\mathbf{B}$ is used to create MIDI sequence. The MIDI is the standard protocol which is used to illustrate music score for electronic music instruments. In Section VI, the results obtained using two videos are explained. Finally, the conclusion and future work are given in Section VII.

IV. Extracting a Rough Temporal Regularity

A. Extracting the raw sudden events

The intensity at the point (x, y) of the n-th video frame($0 \leq n \leq N-1$, $n \in \mathbb{N}$) is $f(x, y, n)$. The frame rate of the video is V [frames/sec]. $F(u, v, n)$ is the spatial frequency domain representation of $f(x, y, n)$. The rate of the intensity change is

$$F'_M(u, v, n) = |F(u, v, n)| - |F(u, v, n-1)|, \quad \text{where } n \geq 1. \tag{1}$$

And the acceleration of the intensity change is

$$F''_M(u, v, n) = F'_M(u, v, n) - F'_M(u, v, n-1), \quad \text{where } n \geq 2. \tag{2}$$

The position of a frequency component in the video is approximately represented by the phase of the $F(u, v, n)$. The unwrapped phase with respect to n is $\angle F(u, v, n)$. The rate of the movement is

$$F'_P(u, v, n) = \angle F(u, v, n) - \angle F(u, v, n-1), \quad \text{where } n \geq 1. \tag{3}$$

And the acceleration of the movement is

$$F''_P(u, v, n) = F'_P(u, v, n) - F'_P(u, v, n-1), \quad \text{where } n \geq 2. \tag{4}$$

I need the significant term of $F''_M(u, v, n)$ and $F''_P(u, v, n)$. So the maximum values of them is chosen for every n.

$$E_M(n) = \max_{u,v} \left[K_M(u, v, n)|F''_M(u, v, n)|\right]. \tag{5}$$

Because people think the moment when the intensity increases and then decreases or *vice versa* is more significant than the other moments, $K_M(u, v, n)$ is defined by

$$K_M(u, v, n) = \begin{cases} 1, & \text{for } F'_M(u, v, n)F'_M(u, v, n-1) > 0 \\ \beta, & \text{for } F'_M(u, v, n)F'_M(u, v, n-1) < 0 \end{cases}. \tag{6}$$

β is a constant larger than 1. The similar process is applied to $F''_P(u, v, n)$:

$$E_P(n) = \max_{u,v} \left[K_P(u, v, n)|F''_P(u, v, n)|W(u, v, n)\right] \quad \text{where } W(u, v, n) = \log_{10}\left(1 + \frac{|F(u, v, n)|}{\alpha}\right), \tag{7}$$

$$K_P(u, v, n) = \begin{cases} 1, & \text{for } F'_P(u, v, n)F'_P(u, v, n-1) > 0 \\ \beta, & \text{for } F'_P(u, v, n)F'_P(u, v, n-1) < 0 \end{cases}. \tag{8}$$

$W(u, v, n)$ is the weight function to filter less visible components. To prevent magnitude term becomes too dominant, the logarithm function is used. α is a constant larger than 1.

B. Normalization, Thresholding, & Grouping

To match the scale of $E_M(n)$ and $E_P(n)$, I normalized by their averages. And I removed some values smaller than

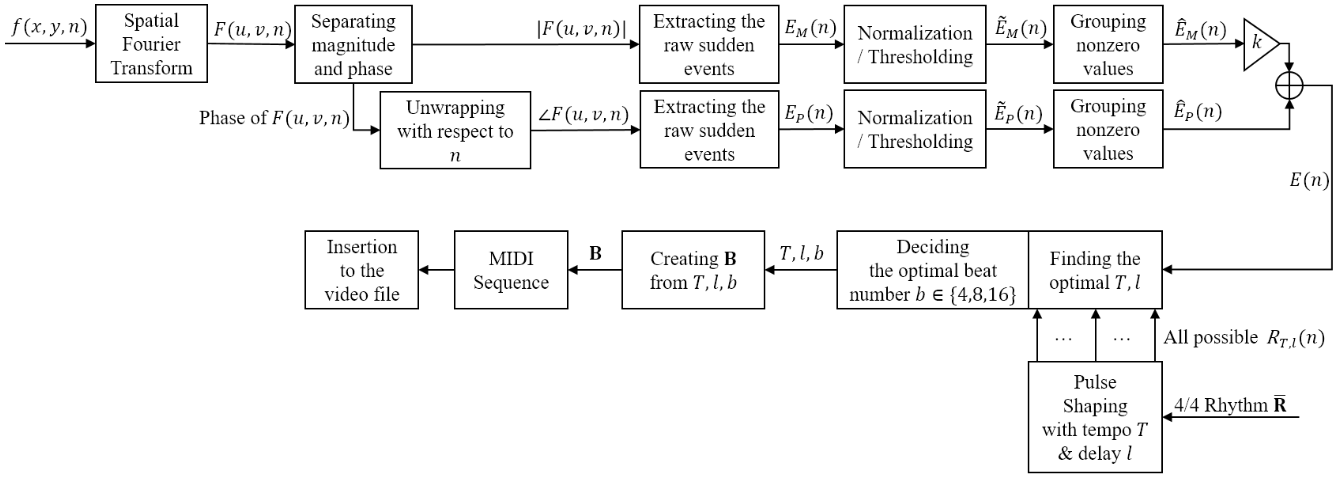
$f(x,y,n)$
Spatial Fourier Transform
$F(u,v,n)$
Separating magnitude and phase
$|F(u,v,n)|$
Phase of $F(u,v,n)$
Unwrapping with respect to n
$\angle F(u,v,n)$
Extracting the raw sudden events
$E_M(n)$
$E_P(n)$
Normalization / Thresholding
$\tilde{E}_M(n)$
$\tilde{E}_P(n)$
Grouping nonzero values
$\hat{E}_M(n)$
$\hat{E}_P(n)$
k
$E(n)$
Finding the optimal T, l
Deciding the optimal beat number $b \in \{4,8,16\}$
T, l, b
Creating **B** from T, l, b
B
MIDI Sequence
Insertion to the video file
All possible $R_{T,l}(n)$
Pulse Shaping with tempo T & delay l
4/4 Rhythm $\bar{\mathbf{R}}$

Fig. 1: The Algorithm Block Diagram

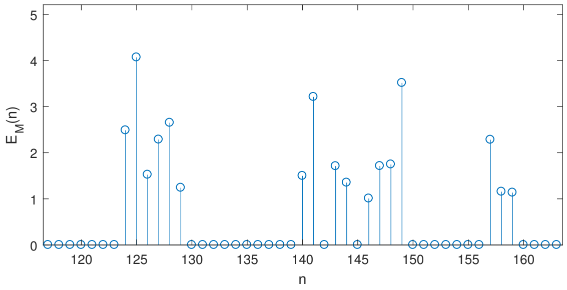

Fig. 2: An example for the grouping process. The values where $n = 124, 126, 127, 128, 129$ are removed during the process.

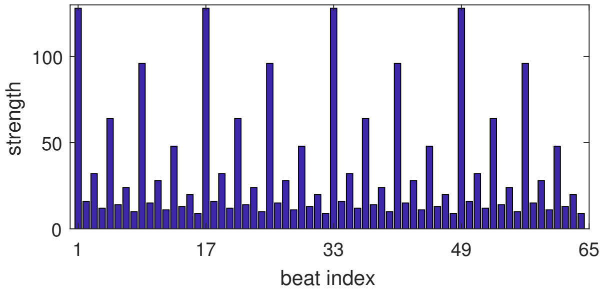

Fig. 3: The vector $[\bar{\mathbf{R}}\ \bar{\mathbf{R}}\ \bar{\mathbf{R}}\ \bar{\mathbf{R}}]$. Each bar means the strength of each beat. Each beat has the length of sixteenth note.

a constant γ because insignificant values don't represent the sudden events.

$$\tilde{E}_M(n) = \begin{cases} E_M(n)/\mu_M, & \text{if } E_M(n)/\mu_M > \gamma \\ 0, & \text{if } E_M(n)/\mu_M \leq \gamma \end{cases}$$

$$\tilde{E}_P(n) = \begin{cases} E_P(n)/\mu_P, & \text{if } E_P(n)/\mu_P > \gamma \\ 0, & \text{if } E_P(n)/\mu_P \leq \gamma \end{cases} \tag{9}$$

$$\text{where } \mu_M = \frac{1}{N}\sum_{n=1}^{N} E_M(n),\ \mu_P = \frac{1}{N}\sum_{n=1}^{N} E_P(n).$$

I assume adjacent nonzero values of $\tilde{E}_M(n)$ and $\tilde{E}_P(n)$ show only one sudden event. Therefore, all values (with proper weights) except the largest value are added to the largest value in each adjacent value group. For example, in Fig. 2, values from $n = 157$ to $n = 159$ are a adjacent value group. Because the largest value is $\tilde{E}_M(157)$, so the value where $n = 157$ becomes $\tilde{E}_M(157)+\tilde{E}_M(158)/|157-158|+\tilde{E}_M(159)/|157-159|$, and the values where $n = 158, 159$ becomes 0. After the process, $\tilde{E}_M(n)$ and $\tilde{E}_P(n)$ must be re-normalized by dividing by the average of their nonzero values. Let $\hat{E}_M(n)$ and $\hat{E}_P(n)$ be the results of that grouping and re-normalizing process from $\tilde{E}_M(n)$ and $\tilde{E}_P(n)$ respectively.

Finally, the sudden events are represented by

$$E(n) = k\hat{E}_M(n) + \hat{E}_P(n) \quad \text{where } k \geq 1. \tag{10}$$

V. Approximating the Rough Regularity to a Typical Musical Rhythm

A. Finding the optimal tempo and delay

In Section IV, the rough regularity consisting of the sudden events $E(n)$ is obtained. Next, I define a bar of typical 4/4 rhythm as a vector

$$\bar{\mathbf{R}} = \begin{bmatrix} 128 & 16 & 32 & 12 & 64 & 14 & 24 & 10 \\ 96 & 15 & 28 & 11 & 48 & 13 & 20 & 9 \end{bmatrix}^T. \tag{11}$$

The i-th element r_i of $\bar{\mathbf{R}}$ means the strength of the i-th beat in a bar when its division level is 16 as illustrated in Fig. 3. I define $T \in \mathbb{N}$ as the length of a quarter note(the unit is the number of frames). Most music have the tempo between 60bpm and 200bpm. The unit bpm means the number of quarter notes per minute. So I restrict the range of T as $T \in \left[\frac{60}{200}V, \frac{60}{60}V\right]$. I used raised cosine filters to interpolate

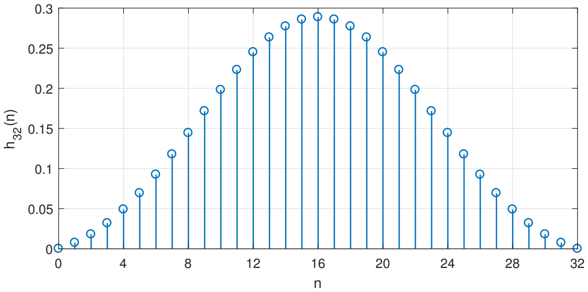

Fig. 4: The impulse response of the discrete-time raised cosine filter $h_{64/2}(n)$.

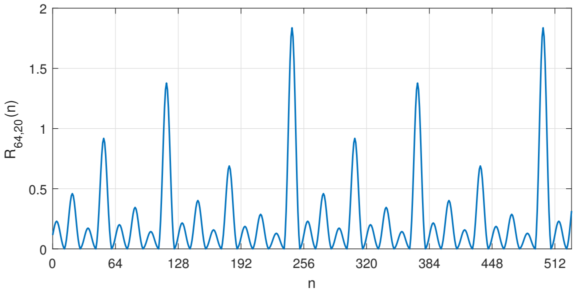

Fig. 5: The rhythm information $R_{T,l}(n)$ with $T = 64$ and $l = 20$, $N = 530$

between each beat. The impulse response of the raised cosine filter $h_{T/2}(n)$ is defined as

$$h_{T/2}(n) = \begin{cases} A\frac{\pi}{4}\mathrm{sinc}\left(\frac{1}{2}\right), & n = \frac{T}{4} \pm \frac{T}{8} \\ A\,\mathrm{sinc}\left(\frac{1}{T/4}(n - T/4)\right) \frac{cos\left(\frac{\pi}{T/4}\left(n-\frac{T}{4}\right)\right)}{1-\left(\frac{2}{T/4}\left(n-\frac{T}{4}\right)\right)^2}, & \\ & n \in \left[0, \frac{T}{2}\right], n \neq \frac{T}{4} \pm \frac{T}{8} \end{cases} \tag{12}$$

as illustrated in Fig. 4. A is a constant to make the energy of $h_{T/2}(n)$ is 1. The rhythm information sequence $R_T(n)$ is defined as

$$R_T(n) = \begin{cases} \sum_{i=1}^{16} r_i h_{T/2}\left(2n - \frac{T}{2}(i-1)\right), & 0 \leq n \leq 4T-1 \\ R_T(n-4T), & 4T \leq n \end{cases}. \tag{13}$$

Let $S_{T,l} = \sum_{n=0}^{N-1} R_T(n+l)$. Then shifted and normalized sequence $R_{T,l}(n)$ is defined as

$$R_{T,l}(n) = \frac{128}{S_{T,l}} R_T(n+l), \qquad \text{where } 0 \leq n \leq N-1,\ 0 \leq l \leq 4T-1. \tag{14}$$

$R_{64,20}(n)$ is illustrated in Fig. 5.

Then I need to find T and l which make $R_{T,l}(n)$ the most similar to $E(n)$. Simply, it is obtained by finding T and l which maximize the correlation between $R_{T,l}(n)$ and $E(n)$.

$$(T, l) = \underset{T\in[0.3V,V],\ l\in[0,4T-1]}{\mathrm{argmax}} \sum_{n=0}^{N-1} R_{T,l}(n)E(n) \tag{15}$$

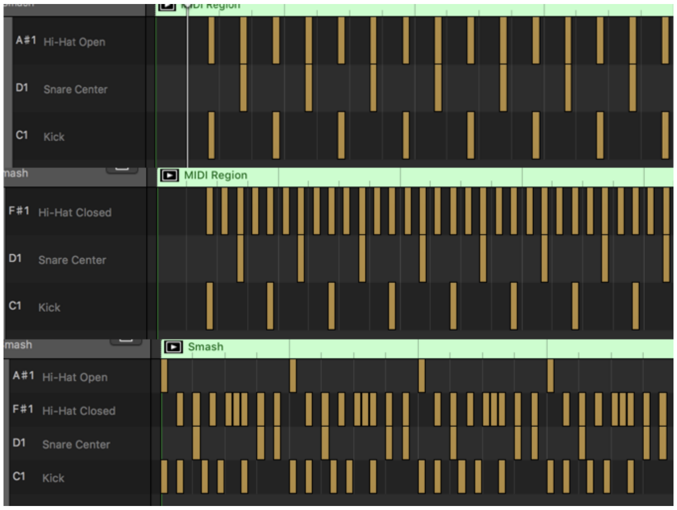

Fig. 6: Simple drum patterns with $b = 4, 8, 16$ from the top to the bottom respectively.

B. Determining the optimal beat number

The optimal beat number b means that rhythm patterns consisting of b-th notes are the most suitable to $E(n)$(or the input video). The algorithm for determining the optimal beat number b is presented in Algorithm 1. Let L be the total number of the location of the 8 weakest beats per musical bar. If the number of noznero values of $E(n)$ which appear at that location is greater than or equal to a half of L, $b = 16$. If not, let L be the total number of the location of the next 4 weakest beats per a musical bar. If the same condition is true, then $b = 8$; if false, $b = 4$.

Algorithm 1 Algorithm for determining the optimal beat number b

```
Input: $E(n)$, $T$, $l$
Output: $b$
for $b := 16, 8$ do
    $\tau := \frac{2T}{b}$
    $c := 0$
    $L := 0$
    for $i := 3\tau - l$ to $N - 1$ step $4\tau$ do
        if $i \geq 0$ then
            $L := L + 1$
            if $E(n)$ has nonzero values for $n \in [i - \tau, i + \tau]$ then
                $c := c + 1$
            end if
        end if
    end for
    if $c \geq \frac{L}{2}$ then
        return $b$
    end if
end for
return 4
```

Fig. 7: The first frame of "Brightness & Pattern.mp4"

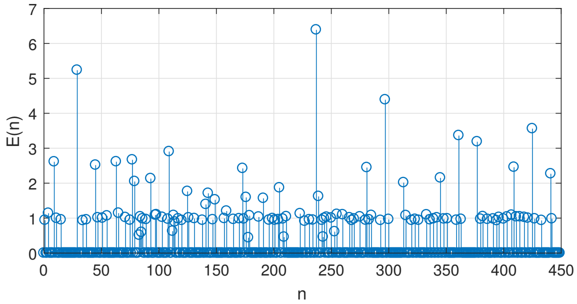

Fig. 8: The result $E(n)$ from "Brightness & Pattern.mp4"

After determining b, T, and l, the simple pattern $\mathbf{B}$ of rhythm instruments(a conventional drum set) is created as Fig. 6. $\mathbf{B}$ is converted to a MIDI sequence by matlab-midi tools [1], and the MIDI sequence is converted to an audio sequence by digital audio workstation(DAW) program. I used Logic Pro X made by Apple Inc..

VI. Experiment Result

A. The Results from "Brightness & Pattern.mp4"

I used "Brightness & Pattern.mp4" by the input video. The first frame of that video is shown in Fig. 7. I set the length $N = 450$. The constants were determined as $\alpha = 3$, $\beta = 2$, $\gamma = 1$, and $k = 1.75$. Because sudden intensity changes are dominant than sudden movements in that video, large values of $E(n)$(illustrated in Fig. 8) were mostly from $\hat{E}_M(n)$. The optimal $R_{T,l}(n)$ was illustrated in Fig. 9. The optimal T,l, and b is presented in Table I.

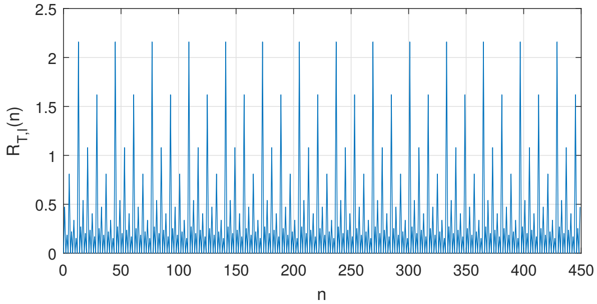

Fig. 9: The result $R_{T,l}(n)$ from "Brightness & Pattern.mp4"

TABLE I: The results from "Brightness & Pattern.mp4"

V	23.9760
T	8
l	20
b	8

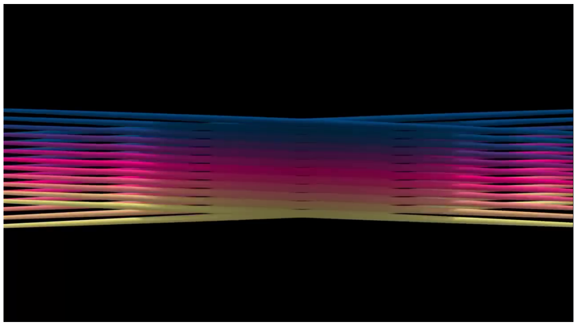

Fig. 10: The first frame of "crosslines.mp4"

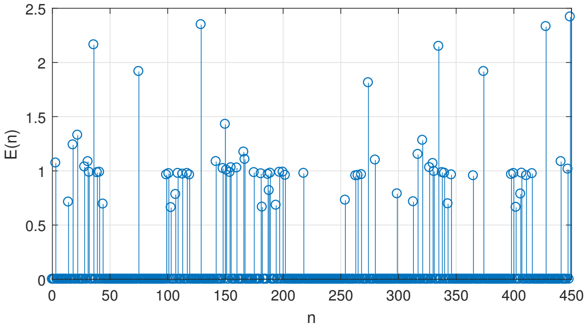

Fig. 11: The result $E(n)$ from "crosslines.mp4"

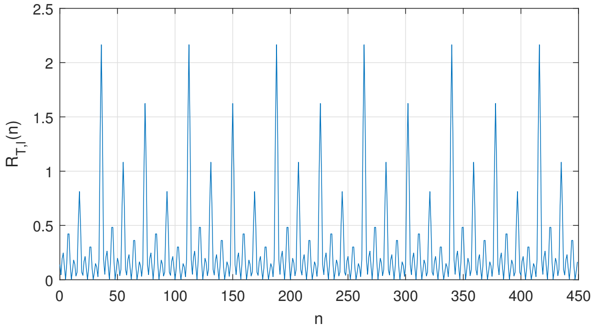

Fig. 12: The result $R_{T,l}(n)$ from "crosslines.mp4"

TABLE II: The results from "crosslines.mp4"

V	29.9700
T	19
l	42
b	8

B. The Results from "crosslines.mp4"

Next, I used "crosslines.mp4" by the input video. The first frame of that video is shown in Fig. 10. I set the length $N = 450$. The constants were determined as $\alpha = 2$, $\beta = 2$, $\gamma = 1$, and $k = 1$. Because sudden movements are quite dominant than sudden intensity changes in that video, values of $E(n)$(illustrated in Fig. 11) are mostly from $\hat{E}_P(n)$. The optimal $R_{T,l}(n)$ is illustrated in Fig. 12. The optimal T,l, and b is presented in Table II.

VII. Conclusion & Future Work

The result rhythm matched to its input video well. In this work, I used only the simple drum patterns that I made, but there are many programs that make natural drum patterns so I can use them. There are three directions that can be studied in the future. First, I need to generalize this algorithm to work with any videos. Second, I can modify the existing algorithms that create natural drum patterns. By using the sudden event data from this work, the created pattern can match to videos better. Finally, I can additionally make an algorithm that recommends conventional music that is suitable for video.

References

[1] K. Schutte, "Matlab and midi." [Online]. Available: http://kenschutte.com/midi