Automatic Hit Time Generation for Music Rhythm Games

Tzu-Chun Yeh
Department of Computer Science, National Tsing Hua University
Kenshin.yeh@mirlab.org

Hao-Hsun Li **Jyh-Shing Roger Jang**
Department of Computer Science & Information Engineering, National Taiwan University
{haohsun.li,jang}@mirlab.org

ABSTRACT

This paper proposes a system that can automatically generates the hit times for music rhythm games. In this system, we apply HPSS (harmonic/percussive source separation) to obtain the harmonic and percussive components [1]. A weighted onset detection method based on spectral flux is used to acquire the onset information from both components. The identified onsets are combined and grouped to generate the hit times of the given music. The system is implemented in an Android pad and can be demonstrated directly to show the playability of the automatic generated hit times.

1. INTRODUCTION

A music rhythm game is a game that player can hit or touch the panel to match the hit times (usually the onset times of instruments) in a given music piece. Most of the hit times are designed and annotated by human manually to make them playable with a certain level of difficulty. This annotation process is time consuming and requires a lot of experience. So our goal is to automate the process and let the user be able to play the rhythm game using any piece of music from him/her personal collection.

In this paper, we describe a system that can automatically generate the hit times of a given music piece to be used in a music rhythm game. The system provides an interface for the user to upload a music piece from a mobile device. A server program will analyze the music and return the hit times to the client side for the corresponding rhythm game. Figure 1 shows the gameplay view of the proposed system. Based on the generated hit times, the note tokens on the left and right channels will fall down sequentially alone two timelines respectively, and the user need to hit the screen on the baseline at the right time to eliminate the note tokens in order to achieve a high score.

Figure 2 shows the basic blocks of the proposed system. The hit times of a given music may exist in the form of onsets in both harmonic component (such as string instruments or human voices) and percussive component (such as drums or cymbols). Thus, we need to separate the given music signal into harmonic and percussive sources by HPSS (harmonic/percussive source separation) [3]. After HPSS, onset detection is applied to both sources to find the onsets in both parts. Here, we proposed a weighted version of spectral flux to achieve more precise onsets [1][2][4]. The final hit times are obtained by merging the results of onset detection on both sources, and eliminating hit times that are too close. Moreover, the hit times are grouped into various sets according to the levels of difficulty which is based on the density of the hit times.

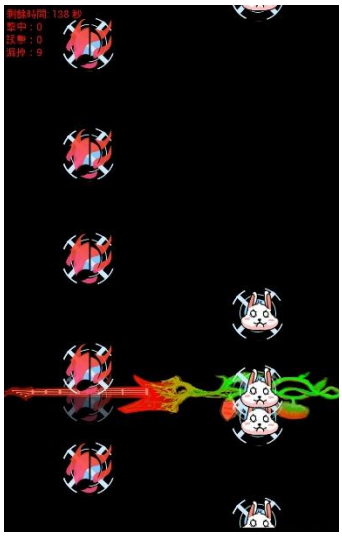

Figure 1. The screenshot of the automatically generated music content and the layout of the proposed system

The format of the output file is also shown in figure 2. More specifically, as shown in Fig 2, The first column of the output is the hit times, the second column is the channel ID of the falling note tokens, and the last column is the levels of difficulty. The channel assignment is based on by k-means clustering on the MFCC at the hit times, such that different channels are likely to host onsets from different instruments.

2. REFERENCES

[1] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies and M. B. Sandler, "A tutorial on onset detection in music signals," IEEE Transactions on Speech and Audio Processing, vol. 13, no. 5, pp. 1035-1047, September 2005.

[2] S. Dixon, "Onset detection revisited," in Proceedings of the International Conference on Digital Audio Effects, 2006.

[3] N. Ono, K. Miyamoto, J. Le Roux, H. Kameoka and S. Sagayama, "Separation of a monaural audio signal

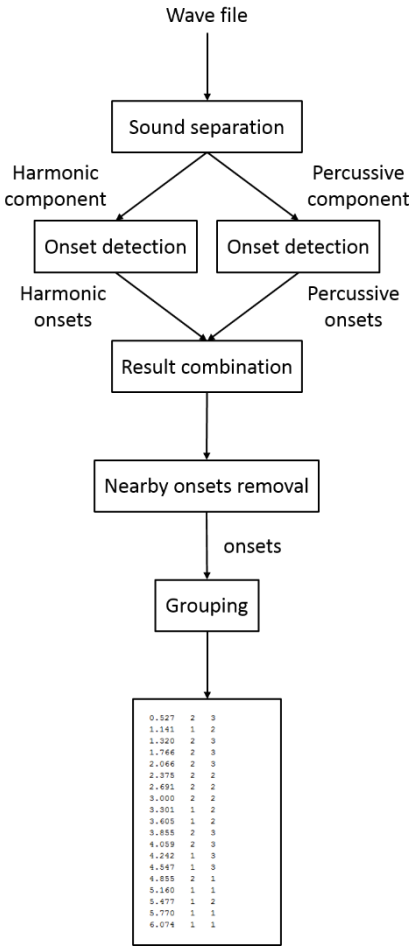

Figure 2. The basic blocks of the proposed system

into harmonic/percussive components by complementary diffusion on spectrogram," in Proceedings of the European Signal Processing Conference, 2008.

[4] J. C. Lagarias, J. A. Reeds, M. H. Wright and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," SIAM Journal on Optimization, vol. 9, no. 1, pp. 112-147, December 1998.