VOCAL-INSTRUMENT SEPARATION PROGRAM

Dahyun Chung and Thomas Downey

University of Rochester, Audio and Music Engineering
Rochester, NY 14627

ABSTRACT

In most forms of current popular music, there are two main components: a collection of vocalists singing a melodic line, and a background part that is played by a collection of instruments. There is considerable interest in giving consumers the ability to separate these two parts of the song, creating one vocal melodic track and one instrumental background track. There exists a MATLAB program that is somewhat successful at turning this concept into a reality. However, this code does not achieve perfect separation. For all songs that are separated using this program, there are still some distorted background tracks and low frequencies. This project observes the successes and failures of the MATLAB program, and it attempts to offer improvements to the code through textual modifications and the addition of a Butterworth filter. We focus on the clarity of the vocal audio file, thus our suggested improvements are mainly beneficial for the quality of the vocal separation.

***Index Terms*—** REPET, Butterworth Filter

1. INTRODUCTION

In this paper, we study the code and theory behind the original REPET code designed by Dr. Zafar Rafii. We focus our attention on the quality of the separated vocal track, and we offer some suggestions for improvement in the code that would enhance the quality of the separated vocal melodic track. Specifically, we suggest some parameter adjustments and the addition of a Butterworth filter.

Our motivation has been drawn from a variety of musicians and a combination of our past experiences. Dahyun became interested in the topic of vocal-instrumental separation through her work as an intern. As the musical director of an a cappella group, Thomas became interested in a vocal-instrumental separation tool after realizing that it would allow an a cappella arranger to listen to the background and foreground parts separately, thus easing the arranging and transcribing process. A simple tool to separate foreground and background could be helpful for many other people as well. For example, a karaoke producer would use the separation tool to eliminate the foreground from a song, allowing karaoke singers to create their own melodies.

Through discussions with Dr. Zhiyao Duan and through independent research, we learned about the existing codes and methods used for background and foreground separation. An effective and available method of separation is known as the REpeating Pattern Extraction Technique (REPET). The idea behind REPET is simple: since most background music in pop songs is repetitive, a computer program should be able to identify these repeating patterns and extract them from the original sound file. Once this background is identified, the foreground is created by subtracting it from the original sound file. Because of REPET's effectiveness, and availability, we decided to focus our work on this program. The process of the REPET code is described in more detail in section II of this paper.

The rest of this paper is arranged as follows: In section II, we discuss in detail the theory and code involved with original REPET. In section III, we offer our suggestions for code changes that would enhance the quality of the vocal track. Section IV discusses the results of our code modifications. Lastly, in section V, we offer conclusions and suggestions for further research.

2. REPET FUNCTION

Before we discuss our modifications and results, we must discuss the steps of REPET in greater detail. REPET is a program designed by Zafar Rafii during his time at Northwestern University. The purpose of REPET is to identify the repeating background parts and then isolate it from the song. Because of this, the code works best when the background parts consist of repeating sections (ex: dance music, instead of modern jazz). The three stages of REPET are visualized in the figure below:

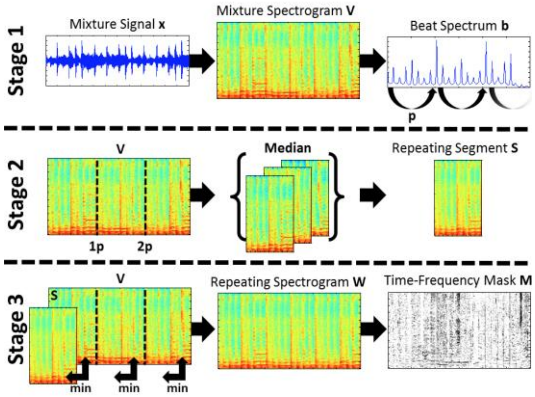

Fig. 1. Graphic explanation of REPET [2]

The first stage of REPET is the identification stage. It will identify the signal of the song, transform into a spectrum and identify the repeating beat periods. The mixture spectrogram V is identified after using STFT and the Hamming window of the original signal.

The second stage determines the repeating pattern, then averages them to subtract them from the original in order to separate the foreground and background. In this process, Wiener Khinchin Theorem is applied in the code. The theorem states, "the autocorrelation function of a wide-sense-stationary random process has a spectral decomposition given by the power spectrum of that process" [1], and therefore, it is used to auto correlate repeating sounds in a given interval.

The last stage of the REPET is when the foreground and background are determined. The calculated repeating sections will be subtracted from the original song (with both vocal and instrumental parts), and the result of that is the foreground, and other element is the background of the song.

3. CODE MODIFICATIONS

This section elaborates on our modifications to the original REPET code, explaining the reasoning behind our changes. Significant results from these changes (specifically the addition of a Butterworth filter) can be found in the following section.

3.1. Cutoff Frequency

In the repet() function at the beginning of the code, there is a high-pass filter that plays a major role in separating the vocal melody from the instrumental background. Ideally, the cut-off frequency for this high pass filter would be set to the lowest frequency vocalized by the singer. In the code, this frequency is set to 100 hertz. However, after using REPET on more than 20 different songs, ranging from many different styles, we did not ever encounter a vocalist that sang below 185 hertz. The only time a singer vocalized a value below 200 hertz was during a rap section of a song.

We decided that, with the varying frequency range of pop singers, it would be best to use a variable "cut" as the cut-off frequency for this high-pass filter. This allows the user to vary its value through user inputs.

3.2. Repeating Time Period

Before the code is executed, the user enters the range of the repeating time period. This range determines the repeating segments that will eventually be averaged and subtracted from the original song to obtain the foreground track. The original code suggested the user to input a range of 0.8 seconds to 8.0 seconds. With this range, we discovered a significant amount of background noise leaking into the vocal track. For this reason, we generated the code with different time period on different songs.

When we lowered the range (ex: 0.8 seconds to 3 seconds), REPET became more efficient at removing drum noises from the vocal track. Thus, we suggest a smaller input range for the code to search for the repeating time period. On the contrary, if we broadened the range of the repeating time period, the overall background sounds were separated. Some of details in the background were still in the vocal track, but most of the background was isolated.

3.3 Butterworth Filter Applied to Vocal Audio File

Despite our previous two code modifications, the generated vocal audio files still contain some heavy bass sound from the drums and bass instruments. In an attempt to dampen this sound, we decided to apply another high-pass filter to the vocal track. After much consideration, we decided to use a Butterworth filter.

Diana suggested the use of the Butterworth filter based on her internship experience in KAIST two years ago when she worked with Adx Trax Pro, a commercial software that separates an original song into vocal and instrument parts. Adx Trax Pro isolates vocal and instrumental tracks more efficiently than the simple version of REPET. However, even after generating Trax Pro, there were low frequency signals that leaked a wide, hollow, bass sound into the vocal track.

We manually deleted the low frequency sounds that were giving unnecessary sound that made both vocal and instrumental unclear through the use of a high-pass Butterworth filter. By facing this issue from using a commercial version, we decided to focus on getting rid of all the low frequencies. We tried different filters but concluded that Butterworth filter fitted the most for our goal in this case that we chose the Butterworth filter instead of other filters. The use of this additional high-pass filter produced a better-isolated vocal track.

4. RESULTS

In this section, we focus on the visualization of a successful implementation of REPET. We briefly mention in section II that REPET works best with pop songs that have repetitive backgrounds. After applying REPET to multiple different genres of music, our theory was confirmed that the code works best with dance music. Below is a series of figures that apply REPET and the Butterworth filter to a dance song. The song is called "Baby Let's Go" by Aziatix, and the representations of REPET implementation can be seen in Figures 2 and 3. Figure 4 shows the implementation of our Butterworth filter.

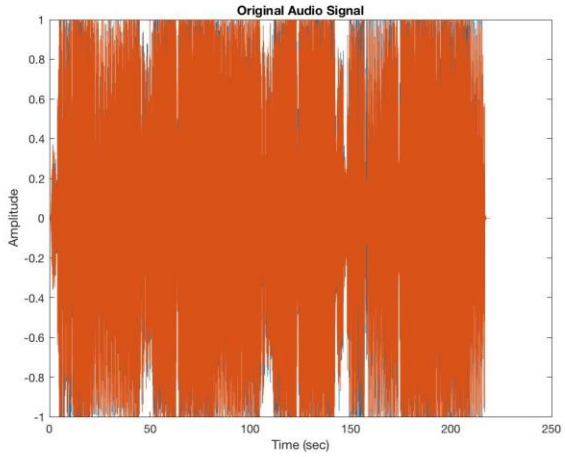

Fig. 2. Original Audio Signal – the background and the foreground are still mixed here.

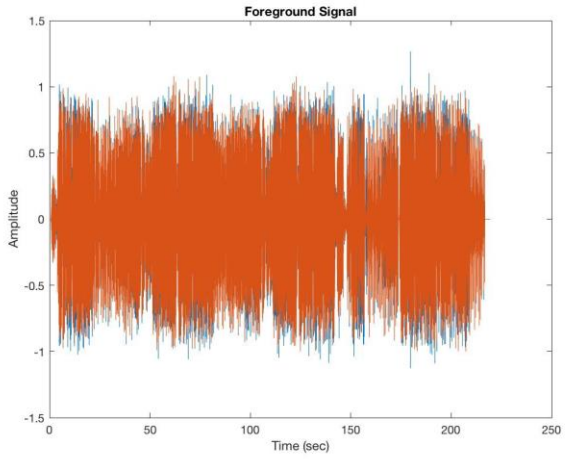

Fig. 3. Foreground track after original REPET code

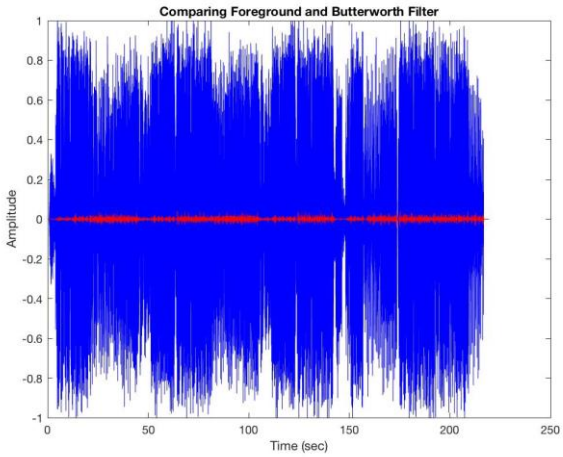

Fig. 4. Implementation of the Butterworth filter on the isolated vocal track. The isolated vocal track is shown in blue, and the low-frequency removed audio is shown in red. Cut-off frequency is at 100 hertz.

As stated before, the audio removed at the Butterworth filter stage is mainly low frequency reverb. This produces an isolated vocal track that has less reverb from the basses and drums, especially the bass drum. A cut-off frequency of 150 hertz was chosen for this Butterworth filter. A higher frequency would have affected the rapper's low voice in the vocal track.

5. CONCLUSIONS

Currently, a program for perfect vocal-instrument separation does not exist. There are other ways to separate the vocal melody from the instrumental background in a singular audio file of a pop songs, and in comparison to other methods of separation, REPET produces results of similar quality. Throughout this paper, we offer some potential improvements to the original REPET process that maximize the clarity of its resulted vocal audio file.

To further improve the clarity of the vocal audio file, additional filters could be applied. While our modifications were especially good at reducing the overall bass sound in the isolated vocal track, there are still some hi-hat sounds that could be removed for further isolation. Potentially, one could remove these sounds with a successful implementation of a Gaussian filter. Further research should focus on the removal of these high-frequency hi-hat sounds.

6. REFERENCES

[1] "Wiener–Khinchin theorem." *Wikipedia*. Wikimedia Foundation, 28 Jan. 2017. Web. 18 Apr. 2017.

<https://en.wikipedia.org/wiki/Wiener%E2%80%93Khinchin_theorem>.
[2] Rafii, Zafar. *Zafar Rafii*. N.p., n.d. Web. 10 Apr. 2017. <http://zafarrafii.com/repet.html>.
[3] Rafii, Z., and B. Pardo. "REpeating Pattern Extraction Technique (REPET): A Simple Method for Music/Voice Separation." *IEEE Transactions on Audio, Speech, and Language Processing* 21.1 (2013): 73-84. Web. 19 Apr. 2017.
[4] Rafii, Zafar, and Bryan Pardo. "A Simple Music/voice Separation Method Based on the Extraction of the Repeating Musical Structure." *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2011): n. pag. Web. 18 Apr. 2017.
[5] Rafii, Zafar, Zhiyao Duan, and Bryan Pardo. "Combining Rhythm-Based and Pitch-Based Methods for Background and Melody Separation." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.12 (2014): 1884-893. Web. 18 Apr. 2017.