A SIMPLE MUSIC/VOICE SEPARATION METHOD BASED ON THE EXTRACTION OF THE REPEATING MUSICAL STRUCTURE

Zafar RAFII

Northwestern University
EECS Department
Evanston, IL, USA
zafarrafii@u.northwestern.edu

Bryan Pardo

Northwestern University
EECS Department
Evanston, IL, USA
pardo@northwestern.edu

ABSTRACT

Repetition is a core principle in music. This is especially true for popular songs, generally marked by a noticeable repeating musical structure, over which the singer performs varying lyrics. On this basis, we propose a simple method for separating music and voice, by extraction of the repeating musical structure. First, the period of the repeating structure is found. Then, the spectrogram is segmented at period boundaries and the segments are averaged to create a repeating segment model. Finally, each time-frequency bin in a segment is compared to the model, and the mixture is partitioned using binary time-frequency masking by labeling bins similar to the model as the repeating background. Evaluation on a dataset of 1,000 song clips showed that this method can improve on the performance of an existing music/voice separation method without requiring particular features or complex frameworks.

***Index Terms*—** Music/Voice Separation, Repeating Pattern, Binary Time-Frequency Masking

1. INTRODUCTION

“Repetition [...] is the basis of music as an art.” [1]. A typical piece of popular music has generally an underlying repeating musical structure, with distinguishable patterns periodically repeating at different levels, with possible variations. An important part of music understanding is the identification of those patterns. To visualize repeating patterns, a two-dimensional representation of the musical structure can be calculated by measuring the (dis)similarity between any two instants of the audio. This similarity matrix can be built from the Mel-Frequency Cepstrum Coefficients (MFCC) [2], the spectrogram [3], the chromagram [4], or other features such as the pitch contour (melody) [5] depending on the application, as long as similar sounds yield similarity in the feature space. The similarity matrix can then be used for example to compute a measure of novelty to locate significant changes in the audio [3] or to compute a beat spectrum to characterize the rhythm of the audio [6]. This ability to detect relevant boundaries within the audio can be of great utility for audio segmentation and audio summarization [3], [4], [5].

We propose to apply such a pattern discovery approach for sound separation, by means of extracting the repeating musical structure. The basic idea is to identify in the spectrogram of a song, time-frequency bins that seem to periodically repeat, and extract them using binary time-frequency masking. An immediate application would be music/voice separation.

Music/voice separation systems usually first detect the vocal segments using some features such as MFCCs, and then apply separation techniques such as Non-negative Matrix Factorization [7], pitch-based inference [8],[9], or adaptive Bayesian modeling [10]. Unlike previous approaches, our method does not depend on particular features, does not rely on complex frameworks, and does not require prior training. Because it is only based on self-similarity, this method could potentially work on any audio, as long as there is a repeating structure. It has therefore the advantage of being simple, fast, blind, and also completely automatable.

The rest of the paper is organized as follows. Section 2 presents the method. Evaluation is done in Section 3. Finally, conclusion and perspectives are discussed in Section 4.

2. METHOD

2.1. Repeating Period

To identify the repeating segments in a song, we first need to estimate a period of the repeating musical structure. Periodicities in a signal can be found by using the autocorrelation function, which measures the similarity between a segment and a lagged version of itself over successive time intervals.

We first compute the spectrogram V of the mixture x, calculated from the magnitude Short-Time Fourier Transform (STFT) X with Hamming windowing of length N, with the symmetric part discarded but the DC component kept. We then compute the autocorrelation of each frequency component (row) of V^2 and obtain the autocorrelation matrix B. We use V^2 to emphasize the appearance of peaks in B. If the mixture x is stereo, V^2 is averaged over the channels. By taking the mean over the rows of B, we finally obtain the vector b which estimates the overall acoustic self-similarity of x as a function of the time lag. We normalize b by its first coefficient. The calculation of b is shown in Equation 1.

$$B(i,j) = \frac{1}{m-j+1} \sum_{k=1}^{m-j+1} V(i,k)^2 \, V(i,k+j-1)^2$$
$$b(j) = \frac{1}{n} \sum_{i=1}^{n} B(i,j) \qquad (1)$$
$$\text{for } i = 1 \ldots n \;\text{ and }\; j = 1 \ldots m, \;\text{ where } n = N/2+1$$

The idea is very similar to the beat spectrum proposed in [6], except that no similarity matrix is explicitly calculated and the scalar product is used in lieu of the cosine similarity. Our experiments showed that this method allows for a clearer visualization of the beat structure in x. For simplicity, we will refer to b as the beat spectrum for the remainder of the paper.

Once the beat spectrum is calculated, the first coefficient, which measures the similarity of the whole signal with itself (time lag of 0), is discarded. If a repeating structure is present in x, b would form peaks periodically repeating at different levels, revealing the hierarchical underlying repeating structure of x. The period p of the repeating musical structure is then defined as the period of the longest strong repeating pattern in x, represented by the peaks with the largest level and repeating at the longest period in b. The calculation of the beat spectrum b and the identification of the repeating period p are illustrated in the top row of Figure 1.

2.2. Repeating Segment Model

After estimating the period p of the repeating musical structure, we use it to evenly segment the spectrogram V into segments of length p. We then compute a mean repeating segment $\overline{V}$ over the r segments of V, which can be thought of as the repeating segment model. The idea is that time-frequency bins comprising the repeating patterns would have similar values at each period, and would also be similar to the repeating segment model. Our experiments showed that the geometric mean leads to a better extraction of the repeating musical structure than the arithmetic mean. The calculation of $\overline{V}$ is shown in Equation 2. The segmentation of the spectrogram V and the calculation of the mean repeating segment $\overline{V}$ are illustrated in the middle row of Figure 1.

$$\overline{V}(i,l) = \left(\prod_{k=1}^{r} V(i, l+(k-1)\,p) \right)^{\frac{1}{r}} \qquad (2)$$
$$\text{for } \; i = 1 \ldots n \;\text{ and }\; l = 1 \ldots p$$

2.3. Binary Time-Frequency Masking

After computing the mean repeating segment $\overline{V}$, we divide each time-frequency bin in each segment of the spectrogram V by the corresponding bin in $\overline{V}$. We then take the absolute value of the logarithm of each bin to get a modified spectrogram $\widetilde{V}$ where time-frequency bins repeating at period p have values near 0. The calculation of $\overline{V}$ is shown in Equation 3.

$$\widetilde{V}(i, l+(k-1)\,p) = \left| \log \left(\frac{V(i, l+(k-1)\,p)}{\overline{V}(i,l)} \right) \right| \qquad (3)$$
$$\text{for } \; i = 1 \ldots n, \;\; l = 1 \ldots p \;\text{ and }\; k = 1 \ldots r$$

V can then be partitioned by assigning time-frequency bins with values near 0 in $\widetilde{V}$ to the repeating background. This assumes that the repeating structure (the music) and the varying sound (the vocals) have sparse and disjoint time-frequency representations. In practice, time-frequency bins of music and voice can overlap, and furthermore the repeating musical structure generally involves variations. Therefore, we introduce a tolerance t when creating the binary time-frequency mask M. Our experiments show that a tolerance of $t = 1$ gives good separation results, both for music and voice. The calculation of M is shown in Equation 4. The calculation of the modified spectrogram $\widetilde{V}$ and the binary time-frequency mask M are illustrated in the bottom row of Figure 1.

$$M(i,j) = \begin{cases} 1 & \text{if } \widetilde{V}(i,j) \leq t \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$
$$\text{for } \; i = 1 \ldots n \;\text{ and }\; j = 1 \ldots m$$

Once the binary time-frequency mask M is computed, it is symmetrized and applied to the STFT X of the mixture x to get the STFT of the music $\hat{X}_{music}$ and the STFT of the voice $\hat{X}_{voice}$, as shown in Equation 5. The estimated music signal $\hat{x}_{music}$ and voice signal $\hat{x}_{voice}$ are finally obtained by inverting their corresponding STFTs into the time domain.

$$\begin{cases} \hat{X}_{music}(i,j) & = \; M(i,j)\, X(i,j) \\ \hat{X}_{voice}(i,j) & = \; (1 - M(i,j))\, X(i,j) \end{cases} \qquad (5)$$
$$\text{for } \; i = 1 \ldots N \;\text{ and }\; j = 1 \ldots m$$

Figure 1 illustrates the whole separation system.

3. EVALUATION

3.1. Dataset

We evaluated our music/voice separation system using the MIR-1K dataset[1]. The dataset contains 1,000 song clips recorded at a sample rate of 16 kHz, with durations ranging from 4 to 13 sec. The clips were extracted from 110 karaoke Chinese pop songs performed by male and female amateurs. The dataset includes manual annotations of the pitch contours, indices of the vocal/non-vocal frames, indices and types for unvoiced frames, and lyrics.

The work in [9] performed music/voice separation on the MIR-1K dataset using a pitch-based inference separation

[1] http://sites.google.com/site/unvoicedsoundseparation/mir-1k

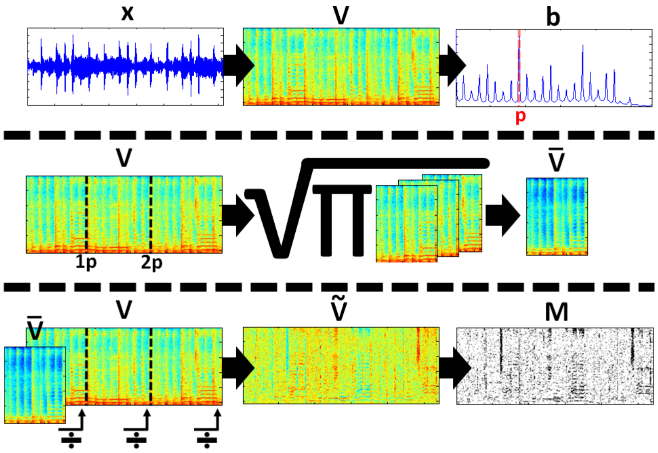

Fig. 1. Overview of our separation system. **1st row**: repeating period p from the beat spectrum b; **2nd row**: segmentation of V to get the mean repeating segment $\overline{V}$; **3rd row**: bin-wise division of V by $\overline{V}$ to get the binary time-frequency mask M.

system. The method combined the singing voice separation method presented in [8] with the separation of the unvoiced singing voice frames and a spectral subtraction technique to enhance music/voice separation.

Following the evaluation framework adopted in [9], we used the 1,000 song clips of the MIR-1K dataset to create 3 sets of mixtures. For each clip, we mixed the music accompaniment and the singing voice into a monaural mixture using 3 different "voice-to-music" ratios: -5 dB (music is louder), 0 dB (same level), and 5 dB (voice is louder).

3.2. Process

In the separation process, the STFT of each mixture x was calculated using a half-overlapping Hamming window of $N = 1024$ samples, equivalent to an analysis length of 0.064 sec at sampling rate of 16 kHz. The repeating period p was automatically estimated from the beat spectrum b simply by computing the local maxima in b and identifying the one that periodically repeats the most often, with the highest accumulated energy over its periods. When building the binary time-frequency mask, we fixed the tolerance t to 1. Our music/voice separation system is thus completely automatic.

For simplicity, x_{voice} is here denoted v. To measure the separation quality between the estimated voice $\hat{v}$ and the original voice v, we used the Signal-to-Distortion Ratio (SDR). As done in [9], we evaluated the separation performance for each mixture by computing the Normalized SDR (NSDR), shown in Equation 6, which measures the improvement of the SDR between the mixture x and the estimated voice $\hat{v}$.

$$NSDR(\hat{x}, v, x) = SDR(\hat{v}, v) - SDR(x, v) \qquad (6)$$

For overall separation performance, the Global NSDR (GNSDR) was calculated by taking the mean of the NSDRs over all the mixtures of each set, weighted by their length. Higher values of GNSDR mean better separation.

3.3. Results

Figure 2 shows the comparison of the overall separation performance for "voice-to-music" ratios of -5, 0, and 5 dB. Black bars (Hsu) show the best automatic version of the pitch-based inference music/voice separation system proposed in [9], with estimated pitch, computer-detected unvoiced frames, and voice enhancement. Gray bars (Rafii) show our automatic music/voice separation system, with estimated period and fixed tolerance. White bars (Ideal) show the ideal binary mask, which serves as the upper-bound on the separation performance. As we can see, our automatic music/voice separation system gave higher GNSDRs than the best automatic version of the pitch-based inference system proposed in [9].

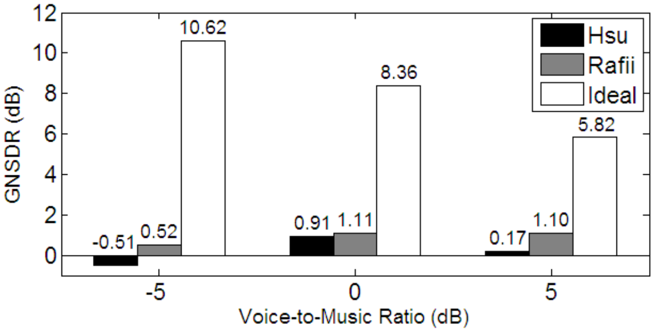

Fig. 2. Comparison of the overall separation performance for "voice-to-music" ratios of -5, 0, and 5 dB, between the best automatic version of the music/voice separation method proposed in [9] *(black)*, our automatic method *(gray)* and the ideal binary mask *(white)*. Higher values are better.

The average computation time for our automatic music/voice separation system over all the mixtures and all the sets was 26 μsec for 1 sec of mixture, when implemented in Matlab on a PC with Intel Core2 Quad CPU of 2.66 GHz and 6 GB of RAM. This shows that our method is also very fast.

The separation performance of our automatic music/voice separation system can be potentially improved if using an optimal period, an optimal tolerance, and the index information of the vocal frames. Therefore, we also evaluated our system by successively adding those enhancements. An optimal period was estimated by identifying the local maxima of the beat spectrum b which led to the highest NSDR. An optimal tolerance was estimated by trying successive values of t, ranging from 0.5 to 2.0 with a step of 0.1, and keeping the one which led to the highest NSDR. The index information of the vocal frames was provided by the MIR-1K dataset and was used to filter out the non-vocal frames of the estimated voice signal $\hat{v}$ at the end.

Figure 3 shows the distributions of the separation performance for “voice-to-music” ratio of 0 dB of our automatic music/voice separation system with estimated period and fixed tolerance, and its enhanced versions obtained by successively adding the use of an optimal period, an optimal tolerance, and the index information of the vocal frames. As we can see, our music/voice separation system can be improved using optimal parameters and extra information. A multivariate analysis of variance (MANOVA) showed that those results were statistically different between cases.

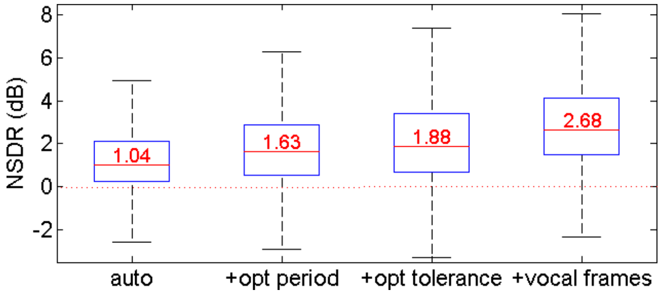

Fig. 3. Separation performance for “voice-to-music” ratio of 0 dB of our automatic music/voice separation method and its successive enhancements. The line in the middle of each box represents the median. Outliers are not shown.

4. CONCLUSION

We have proposed a novel method for music/voice separation, by extraction of the underlying musical repeating structure. Evaluation on a dataset of 1,000 song clips showed that this method can achieve better separation performance than an existing automatic approach, without requiring particular features or complex frameworks. This method also has the advantage of being simple, fast and completely automatable.

There are several directions in which we want to take this work. First, we would like to improve our automatic music/voice separation system by (1) implementing a better repeating period finder, (2) building better time-frequency masks, for example by using a measure of repetitiveness when assigning time-frequency bins, and (3) taking into account the pitch, timbre, or multichannel information. We could also combine our method with other existing music/voice separation systems to improve separation performance. Then, we would like to extend this separation approach for the extraction of multiple hierarchical repeating structures, by using repeating periods at different levels. Finally, we would like to apply this separation approach to the extraction of individual repeating patterns by using a similarity matrix. This could be used for the separation of structural elements in music.

The authors would like to thank Chao-Ling Hsu for providing his numerical results. This work was supported by NSF grant numbers IIS-0643752 and IIS-0757544.

5. REFERENCES

[1] Heinrich Schenker. *Harmony*. University of Chicago Press, 1954.

[2] Jonathan Foote. Visualizing music and audio using self-similarity. In *ACM Multimedia*, volume 1, pages 77–80, Orlando, FL, USA, 30 October-05 November 1999.

[3] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *International Conference on Multimedia and Expo*, volume 1, pages 452–455, New York, NY, USA, 30 July-02 August 2000.

[4] Mark A. Bartsch. To catch a chorus using chroma-based representations for audio thumbnailing. In *Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, 21-24 October 2001.

[5] Roger B. Dannenberg. Listening to “Naima”: An automated structural analysis of music from recorded audio. In *International Computer Music Conference*, pages 28–34, Gothenburg, Sweden, 17-21 September 2002.

[6] Jonathan Foote and Shingo Uchihashi. The beat spectrum: A new approach to rhythm analysis. In *International Conference on Multimedia and Expo*, pages 881–884, Tokyo, Japan, 22-25 August 2001.

[7] Shankar Vembu and Stephan Baumann. Separation of vocals from polyphonic audio recordings. In *International Conference on Music Information Retrieval*, pages 337–344, London, UK, 11-15 September 2005.

[8] Yipeng Li and DeLiang Wang. Separation of singing voice from music accompaniment for monaural recordings. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 15(4):1475–1487, May 2007.

[9] Chao-Ling Hsu and Jyh-Shing Roger Jang. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):310–319, February 2010.

[10] Alexey Ozerov, Pierrick Philippe, Frédéric Bimbot, and Rémi Gribonval. Adaptation of bayesian models for single-channel source separation and its application to voice/music separation in popular songs. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1564–1578, July 2007.