MUSIC/VOICE SEPARATION USING THE 2D FOURIER TRANSFORM

Prem Seetharaman, Fatemeh Pishdadian, Bryan Pardo*

Northwestern University
Electrical Engineering and Computer Science
Evanston, IL

ABSTRACT

Audio source separation is the act of isolating sound sources in an audio scene. One application of source separation is singing voice extraction. In this work, we present a novel approach for music/voice separation that uses the 2D Fourier Transform (2DFT). Our approach leverages how periodic patterns manifest in the 2D Fourier Transform and is connected to research in biological auditory systems as well as image processing. We find that our system is very simple to describe and implement and competitive with existing unsupervised source separation approaches that leverage similar assumptions.

Index Terms— Audio source separation, singing voice extraction, 2DFT, auditory scene analysis, automatic karaoke, foreground/background separation, image processing

1. INTRODUCTION

Audio source separation is the act of isolating sound sources in an audio scene. Examples of source separation include isolating the bass line in a musical mixture, isolating a single voice in a loud crowd, and extracting the lead vocal melody from a song. Automatic separation of auditory scenes into meaningful sources (e.g. vocals, drums, accompaniment) would have many useful applications. These include melody transcription [1], audio remixing [2], karaoke [3], and instrument identification [4].

One application of source separation is singing voice extraction. A variety of approaches have been used for singing voice extraction, the vast majority of which use the spectrogram as the input representation. Examples include Non-negative matrix factorization [5], deep learning-based approaches [6], a source filter model with melodic smoothness constraints [7] and a multi-kernel framework [8].

One of the simplest and most robust approaches for singing voice extraction is to leverage repetition. REPET-SIM [9] uses repetition in the spectrogram by using the similarity matrix to find similar frames. Huang et al. [10] separate a low-rank background (the accompaniment) from a sparse foreground (the singing voice) using robust principal component analysis. The most closely related work to ours is REPET [3], which finds periodic repetition in a magnitude spectrogram, separating a periodic repeating background (accompaniment) from a non-periodic foreground (vocals). In this work we describe a novel, simple method to separate the periodic from the non-periodic audio that leverages the two dimensional Fourier transform (2DFT) of the spectrogram. The properties of the 2DFT let us separate the periodic from the non-periodic without the need to create an explicit model of the periodic audio and without the need to find the period of repetition, both of which are required in REPET.

The 2DFT has been used in music information retrieval for cover song identification [11] [12] and music segmentation [13]. There is also some prior work in audio source separation that uses the 2DFT as the input representation. Stöter et al. [14] apply the 2DFT to small 2D patches of the spectrogram. Pishdadian et al. [15] further refined this representation by using a multi-resolution 2D filter bank instead of fixed-size 2D patches. Both approaches use the 2DFT to differentiate modulation characteristics (e.g. vibrato, trills) of distinct sources and separate them from one another. These works both focus on separation of harmonic sources with the same fundamental frequencies (unisons) in very short excerpts of audio. Neither focuses on separating periodic from non-periodic patterns in long audio segments and both required the creation of a more complicated, tiled representation using the 2DFT. We present a novel singing voice extraction technique to separate periodic from non-periodic audio via a single 2DFT of the spectrogram, with no need to create a more complex multi-resolution filter bank.

2. PROPOSED METHOD

Our approach leverages the fact that musical accompaniment will typically have some amount of periodic repetition, while the vocals will be relatively aperiodic. Given this insight, we use the 2DFT to analyze the audio spectrogram and borrow a technique from image processing to perform singing voice extraction.

*This work was supported by NSF Grant 1420971.
2.1. The 2D Fourier Transform

The 2DFT is an essential tool for image processing, just as the 1DFT is essential to audio signal processing. The 2DFT decomposes images into a summation of weighted and phase-shifted 2D sinusoids [16]. We apply a 2DFT to the magnitude spectrogram of audio mixtures to detect and extract particular patterns such as temporal repetitions. We refer to the vertical and horizontal dimensions of the 2D transform domain as scale and rate. These terms are borrowed from studies of the auditory system in mammals [17][18][19], which have shown that the primary auditory cortex uses representations capturing the spectro-temporal modulation patterns of audio signals. In this context, scale corresponds to the spread of spectral energy (e.g. frequency modulation depth) as well as frequency-domain repetitions (e.g. overtones) and rate corresponds to temporal repetitions (e.g. repeating percussive patterns).

In Figure 1, the left and middle columns show illustrative examples of 2D (time-frequency domain) sinusoids and their 2DFTs (scale-rate domain). A 2D sinusoid is represented by a pair of peaks in the transform domain, where the orientation of the peaks with respect to axes (upward or downward) is the opposite of the orientation of the sinusoid. The rate of repetitions across the frequency and time axes are reflected by the absolute value of scale and rate respectively. The right column shows a more complex pattern which can be decomposed into a number of 2D sinusoids using the 2DFT.

A common task in image processing is to remove noise from images. One particular denoising application is the removal of periodic noise, which can be the result of artifacts in the image capture instrument. A straightforward technique to removing periodic noise from an image is by recognizing that periodic noise will appear as a set of peaks in the 2DFT domain (see Figure 2). When 2DFT-domin peaks are masked out, one can invert the resulting representation to produce an image without the periodic noise.

In many audio signals (e.g. music), a non-periodic foreground source (e.g. a singing voice) is often accompanied by a periodic background source (e.g. a repetitive musical accompaniment). Our work adapts the idea of periodic noise removal in images to the audio realm by applying it to the magnitude spectrogram. By masking peaks in the 2DFT of the spectrogram, we can separate the periodic background from the non-periodic foreground. We now describe this algorithm for music/voice separation in more detail.

2.2. Music/voice separation

Let \( x(t) \) denote a single-channel time-domain audio signal and \( X(\omega, \tau) \) its complex Short-time Fourier Transform (STFT), where \( \omega \) is frequency and \( \tau \) is time. Our goal is to model the background music based on a repeating pattern in the magnitude plot of \( X(\omega, \tau) \), also called the spectrogram. To this end, all the processing in our algorithm will be performed on \( |X(\omega, \tau)| \), where \(|.|\) denotes the magnitude operator. Periodically repetitive patterns in the magnitude spectrogram will appear as peaks in the 2DFT of the spectrogram, which reduces a general pattern recognition approach in the time-frequency domain into peak picking in the scale-rate domain.
The scale-rate representation of the spectrogram will be denoted by $\tilde{X}(s, r)$, where $s$ and $r$, stand for scale and rate respectively. The relationship between the spectrogram and its scale-rate transform can then be formulated as

$$\tilde{X}(s, r) = \mathcal{FT}_{2D}\{|X(\omega, \tau)|\},$$

where $\mathcal{FT}_{2D}\{\cdot\}$ denotes the two-dimensional Fourier transform. $\tilde{X}(s, r)$ contains complex values. The magnitude of $\tilde{X}(s, r)$ contains peaks corresponding to periodically repeating elements in the time-frequency domain. Therefore, the core of our algorithm is to locate peaks in the magnitude of the scale-rate transform (2DFT) and mask the peaks to separate the repeating accompaniment from the singing voice. We pick peaks by comparing the difference between the maximum and minimum magnitude values over a neighborhood surrounding each point in the scale-rate domain to some threshold. In this work, the threshold, denoted by $\gamma$, is set to the standard deviation of all $|\tilde{X}(s, r)|$ values.

The neighborhood for peak-picking can be of an arbitrary shape. For this work, we restrict our neighborhood shape to be a simple rectangle in the 2DFT domain. We denote the center of an arbitrary rectangular neighborhood by $c = (s_c, r_c)$, and the neighborhood surrounding this point by $N(c)$. The dimensions of the neighborhood along the scale and rate axes are tunable parameters in our algorithm.

The repeating accompaniment manifests as a series of peaks along the rate axis. Because of this, our neighborhood is shaped to find peaks along the rate axis. In this work, the size of the neighborhood along the scale axis is 1. In our experiments, we vary the size of this neighborhood along the rate axis between 15 and 100 frames in the 2DFT domain. Smaller values for the shape result in leakage from the singing voice into the accompaniment, while larger values result in leakage from accompaniment into singing voice.

Let $\alpha_c$ denote the range of $|\tilde{X}(s, r)|$ values over the neighborhood, that is

$$\alpha_c = \max_{N(c)} |\tilde{X}(s, r)| - \min_{N(c)} |\tilde{X}(s, r)|.$$

The value of the peak-picking mask, which we will refer to as the scale-rate domain background mask can thus be computed at $c$ as follows

$$M_{bg}(s_c, r_c) = \begin{cases} 1 & \alpha_c > \gamma, |\tilde{X}(s_c, r_c)| = \max_{N(c)} |\tilde{X}(s, r)| \\ 0 & \text{otherwise} \end{cases}$$

Intuitively, this is simply a way to discover local maxima in $|\tilde{X}(s, r)|$ that are above a threshold $\gamma$. It should be noted that neighborhood selection and mask value computation is performed for every single point in the scale-rate domain. We denote the computed background mask over the
In short, ground audio signals are recovered from the masked STFT. The best performance for our system is indicated by an asterisk, while the best performance across all algorithms is indicated by boldface. Note that SDR for foreground and background sources for our optimal settings are higher than those of REPET, but lower than REPET-SIM, which has the advantage of exploiting non-periodic patterns as well as periodic ones.

### Table 1: SDR/SIR/SAR for the singing voice and the music accompaniment as extracted from the mixture.

<table>
<thead>
<tr>
<th>Method</th>
<th>Voice SDR</th>
<th>Voice SIR</th>
<th>Voice SAR</th>
<th>Music SDR</th>
<th>Music SIR</th>
<th>Music SAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPCA</td>
<td>2.3 ± 1.5</td>
<td>11.0 ± 4.5</td>
<td>2.9 ± 4.0</td>
<td>5.0 ± 2.3</td>
<td>7.7 ± 2.9</td>
<td>10.5 ± 6.6</td>
</tr>
<tr>
<td>REPET-SIM</td>
<td>2.1 ± 1.5</td>
<td>15.2 ± 4.2</td>
<td>2.9 ± 3.8</td>
<td>6.3 ± 2.7</td>
<td>12.5 ± 2.7</td>
<td>10.5 ± 6.7</td>
</tr>
<tr>
<td>REPET</td>
<td>2.2 ± 1.5</td>
<td>15.6 ± 4.9</td>
<td>2.8 ± 3.8</td>
<td>5.0 ± 2.6</td>
<td>10.2 ± 2.7</td>
<td>10.4 ± 6.6</td>
</tr>
<tr>
<td>2DFT (1, 15)</td>
<td>2.6 ± 1.5</td>
<td>11.8 ± 3.9</td>
<td>2.8 ± 4.0</td>
<td>5.7 ± 2.5*</td>
<td>8.7 ± 3.0*</td>
<td>10.4 ± 6.7</td>
</tr>
<tr>
<td>2DFT (1, 35)</td>
<td>2.7 ± 1.6*</td>
<td>13.2 ± 3.9</td>
<td>2.8 ± 4.0*</td>
<td>5.1 ± 2.5</td>
<td>7.6 ± 2.9</td>
<td>10.4 ± 6.6*</td>
</tr>
<tr>
<td>2DFT (1, 100)</td>
<td>2.6 ± 1.5</td>
<td>13.5 ± 4.0*</td>
<td>2.7 ± 4.0</td>
<td>4.4 ± 2.4</td>
<td>6.7 ± 2.8</td>
<td>10.3 ± 6.6</td>
</tr>
<tr>
<td>Ideal Binary Mask</td>
<td>9.2 ± 2.7</td>
<td>30.0 ± 4.1</td>
<td>9.5 ± 3.2</td>
<td>14.9 ± 6.5</td>
<td>27.9 ± 8.0</td>
<td>15.2 ± 6.5</td>
</tr>
</tbody>
</table>

The separated audio is obtained by masking in the time-frequency domain. The time-frequency masks are simply computed by comparing the inverted magnitude spectrograms from the 2DFT for foreground and background:

\[
M_{bg}(\omega, \tau) = \begin{cases} 
1 & |X_{bg}(\omega, \tau)| > |X_{fg}(\omega, \tau)| \\
0 & \text{otherwise,}
\end{cases}
\]

and the foreground mask as \(M_{fg}(\omega, \tau) = 1 - M_{bg}(\omega, \tau)\).

In the last step, the time-domain background and foreground audio signals are recovered from the masked STFT. In short, \(x_{bg}(t) = ISTFT\{M_{bg}(\omega, \tau) \odot \tilde{X}(\omega, \tau)\}\), where \(ISTFT\{\cdot\}\) is the Inverse Short-Time Fourier Transform, computed through the overlap-and-add method. The foreground audio signal (the singing voice) can be similarly computed by applying the foreground mask to the complex spectrogram and taking the inverse STFT. The separation process can be seen in Figure 3.

### 3. EVALUATION

We evaluate our approach using DSD100 [20], a dataset consisting of 100 multitrack recordings of four sources - vocals, drums, bass, and other. We label the combination of the latter three sources the accompaniment. Our task is to separate the vocals from the accompaniment. We extract 30 second clips from each multitrack example. The four sources (vocals, drums, bass, other) are combined into a mono mixture for separation. We evaluate our approach using DSD100 [20], a dataset consisting of 100 multitrack recordings of four sources - vocals, drums, bass, and other. We label the combination of the latter three sources the accompaniment. Our task is to separate the vocals from the accompaniment. We extract 30 second clips from each multitrack example. The four sources (vocals, drums, bass, other) are combined into a mono mixture for separation.

---

4. CONCLUSION

We presented a simple and novel approach for music/voice separation. Our approach leverages how periodic patterns manifest in the scale-rate domain and is connected to research in biological auditory systems as well as image processing. We find that our system is competitive with existing unsupervised source separation approaches that leverage similar assumptions.

---

1Audio examples at https://interactiveaudiolab.github.io/demos/2dft.
5. REFERENCES


