Time Frequency Filter Bank: A Simple Approach for Audio and Music Separation

Ning Yang, Muhammad Usman*+, *Student Member, IEEE,* Xiangjian He*, *Senior Member, IEEE,* Mian. A. Jan, *Member, IEEE* and Liming Zhang, *Member, IEEE*

***Abstract*—Blind Source Separation techniques are widely used in the field of wireless communication for a very long time to extract signals of interest from a set of multiple signals without training data. In this paper, we investigate the problem of separation of human voice from a mixture of human voice and sounds from different musical instruments. The human voice may be a singing voice in a song or may be a part of some news, broadcast by a channel with background music. This paper proposes a generalized Short Time Fourier Transform (STFT)-based technique, combined with filter bank to extract vocals from background music. The main purpose is to design a filter bank and to eliminate background aliasing errors with best reconstruction conditions, having approximated scaling factors. Stereo signals in time frequency domain are used in experiments. The input stereo signals are processed in the form of frames, and passed through the proposed STFT-based technique. The output of the STFT-based technique is passed through the filter bank to minimize the background aliasing errors. For reconstruction, first an inverse STFT is applied and then the signals are reconstructed by the OverLap Add method to get the final output, containing vocals only. The experiments show that the proposed approach performs better than the other state-of-the-art approaches, in terms of Signal to Interference Ratio (SIR) and Signal to Distortion Ratio (SDR), respectively.**

***Index Terms*—Blind Source Separation, Short Time Fourier Transform, OverLap Add, SIR, SDR.**

I. Introduction

AUDIO source separation is always considered as a challenging task with many applications, such as polyphonic music separation, speech recognition, and automatic meeting transcriptions. In such tasks, usually the channel characteristics between sources are always unknown and this is called a Blind Source Separation (BSS) problem [1]. As the sources and underlying mixture operators are assumed to be unknown, the term "Blind" can easily be justified. Many techniques to solve the BSS problem have been explored deeply in recent years to generate unknown signal sources from known signal mixtures, especially the speech mixtures [2], [3]. There are many popular and effective algorithms, available for exploring such signal mixtures, which include Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Canonical Correlation Analysis (CCA), Dependent Component Analysis (DCA), Independent Component Analysis (ICA), Non-negative Matrix Factorization (NMF), Low Complexity Coding and Decoding (LCCD), Stationary Subspace Analysis (SSA) and Common Spatial Pattern (CSP) [4]–[7]. These algorithms have also been widely used in many applications, such as bio-medical signal processing, radar signal processing, wireless communication, geographical analysis, Electroencephalography (EEG) and Electrocardiogram (ECG) technologies [8], [9]. Based on the separation procedures and mathematical representations, the BSS algorithms can be classified into two general categories, namely linear and non-linear algorithms [10].

Time domain frequency representation has been widely used as a tool to classify a mixture of signals into N components (i.e., signals), and has applications in bio-medical processing, radar processing, speech processing, and audio signal processing [11]–[13]. For these applications, many convolution-based BSS approaches target at continuous signals. However, these approaches are computationally very expensive and demand complicated hardware resources for implementation [14]. To tackle the concerns, Short Time Fourier Transform (STFT) has been used to determine the individual/local frequency components and phase values, as the signal changes over time. The STFT divides the whole signal into smaller segments of equal lengths to calculate Fourier spectrum of individual components in order to plot the changing spectrum as a function of time [15], [16].

The STFT uses Time Window (TW) of fixed sizes in order to obtain the local signals. After getting the local components, Fourier Transform (FT) is applied for further analysis. This analysis may produce poor results in temporal domain [17]. In order to minimize fluctuations from obtained results, many adaptive algorithms are introduced in STFT domain [18]–[22]. These adaptive algorithms can broadly be classified into two categories, i.e. Chirp Rate (CR) class and Concentration Measure (CM) class [23]–[25]. In the CM-based approaches, specific parameters of input signal are examined in time frequency domain to find out energy variation before applying the STFT at selected parameters [26], [27]. Such techniques are useful in finding optimum results at the cost of computational complexity. On the other hand, the CR-based approaches use wavelets and their derivation to calculate the optimum size of the STFT window in order to find out signal characteristics under the selected window with fixed sampling rate and less computational complexity [28].

Although the CM-based approaches are computationally

Manuscript is submitted on 14 August, 2017.

* indicates corresponding authors and + indicates the equal first author.

Muhammad Usman and Xiangjian He are with School of Electrical and Data Engineering, University of Technology Sydney, Australia. (E-mail: Muhammad.Usman@uts.edu.au, Xiangjian.He@uts.edu.au).

Mian Ahmad Jan is with Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan (E-mail: mianjan@awkum.edu.pk)

Ning Yang is with College of Automation, Northwestern Polytechnical University, China (E-mail: ningyang@nwpu.edu.cn)

Liming Zhang is with School of Computer Science, University of Macau. (E-mail: lmzhang@umac.mo)

complex, they are still preferred due to their fine results. Another main reason behind using the CM-based approaches is that, in many cases, especially the speech and music signals, various signal bands are very narrow and appear around a certain range of frequencies [29]. Therefore, sometimes it becomes possible to reduce computational complexity by restricting the computation onto the narrow range in the frequency domain, in order to extract spectral information. However, our proposed approach is a combination of both CR and CM-based techniques with the contributions as follows.

- Estimation of individual signal sources along with phase and channel coefficients information using an STFT-based technique
- Development of a filter bank to minimize the aliasing error with a check of minimum computational cost and stages
- Less stages of filter bank and fixed point STFT-based processing to support real-time processing

The rest of this article is structured as follows. The Section II describes the principles and notations used in the STFT. The proposed technique is presented in Section III. Section IV contains experimental setup and simulation results. Finally, the article is concluded in Section V.

II. Theoretical Detail

In general, the blind source separation is used to extract the original signal, $x(n)$, from a perceived signal, $\bar{x}(n)$. Here, the term "Blind" stresses on the fact that there is no prior information available about the source signal. The BSS can fall into the *Semi-Blind* category when making some assumptions on the source signal's characteristics. The perceived signal can be a natural mixture or a studio mixture of different signals. Studio mixtures can be linear or instantaneous, and they have no guarantee to be like natural mixtures. However, noise is always present in either type of mixtures, and becomes challenging for a BSS method to deal with it. In this paper, we focus on the linear studio mixtures. Independent Component Analysis (ICA) is very popular in the BSS approaches. It exploits the non-Gaussianity of a perceived signal to estimate the original signal sources. Many BBS approaches have been proposed using ICA in either the time or frequency domain [30]. The approaches in the time domain are computationally expensive, as they require multiple convolutions. The approaches in the frequency domain are preferable because 1) they transform convolutions into multiplications, and 2) signals have non-Gaussian nature in the frequency domain, on which the STFT is an ideal framework for ICA.

The STFT is usually used for the time frequency representations of local sections of a time varying signal with the help of a TW function. Fig. 1 represents a sample of a time varying signal with varying frequencies over time.

If the input signal is represented by $x(t)$ ($t \in (-\infty, \infty)$), then its STFT can be performed as shown in Eq.1.

$$\text{STFT}\{x(t)\}(m,\omega) \equiv X(m,\omega) = \int_{-\infty}^{\infty} x(t)w(t-m)e^{-j\omega t}dt, \tag{1}$$

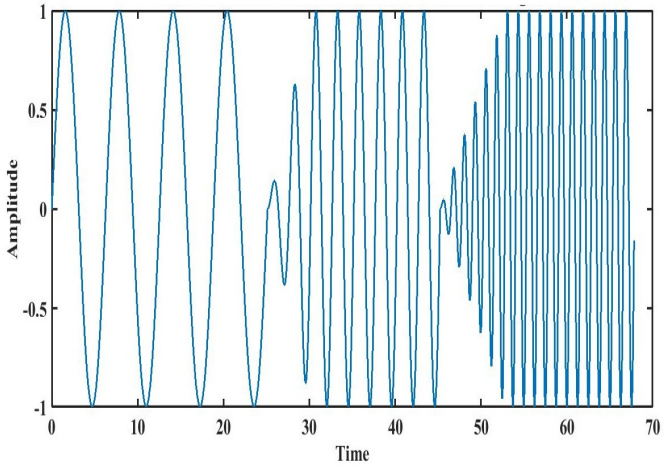


Fig. 1: Sample Time Varying Signal

where $w(t)$ represents the sliding window function, $X(m,\omega)$ is the FT of $x(t)w(t-m)$, m represents the time axis, and ω represents the frequency axis [31].

In the case of discrete time processing, input data can be chopped into chunks or frames. To reduce artifacts at the boundaries, the frames are usually overlapped. The FT is computed for each frame, resulting in the form of a complex function. The results of the complex function are recorded into a matrix, which records the phase and magnitude of each point in both frequency and time domains. The discrete-time STFT with signal $x[n]$ and window $w[n]$ can be expressed by the following equation [31].

$$\text{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n}, \tag{2}$$

where, both m and ω are discrete and quantized.

Size and shape of the TW play important roles in signal analysis. The size of the TW shares a complicated relationship with the time and frequency resolutions of the STFT. The shorter the size of the TW is, the higher the time resolution and the lower the frequency resolution are. On the other hand, increasing the size of the TW increases the frequency resolution but decreases the time resolution. This rule is known as the Heisenberg's uncertainty principle [32], [33].

There are different TW shapes available to handle various needs in different situations. The overlapping sections, especially, the non-zero overlapping sections in the TW also play important roles and require a careful consideration when their sizes are chosen. The size of a small non-zero overlapping section helps to detect smaller changes in adjacent data frames at the cost of computational complexity. The size of a non-zero overlapping section is also directly related to the size of the STFT matrix [22]. Some sample TWs are shown in Fig. 2. The schematic diagrams for forward and inverse STFTs are shown in Fig. 3.

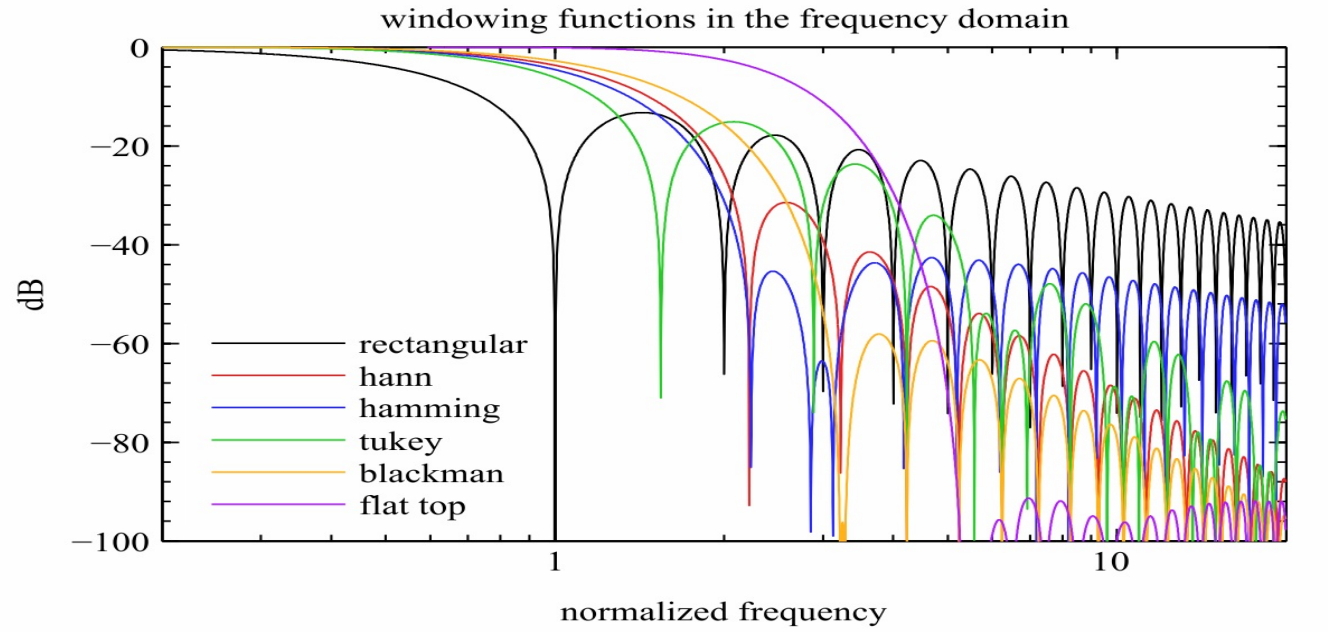


Fig. 2: Time Windows in Frequency Domain

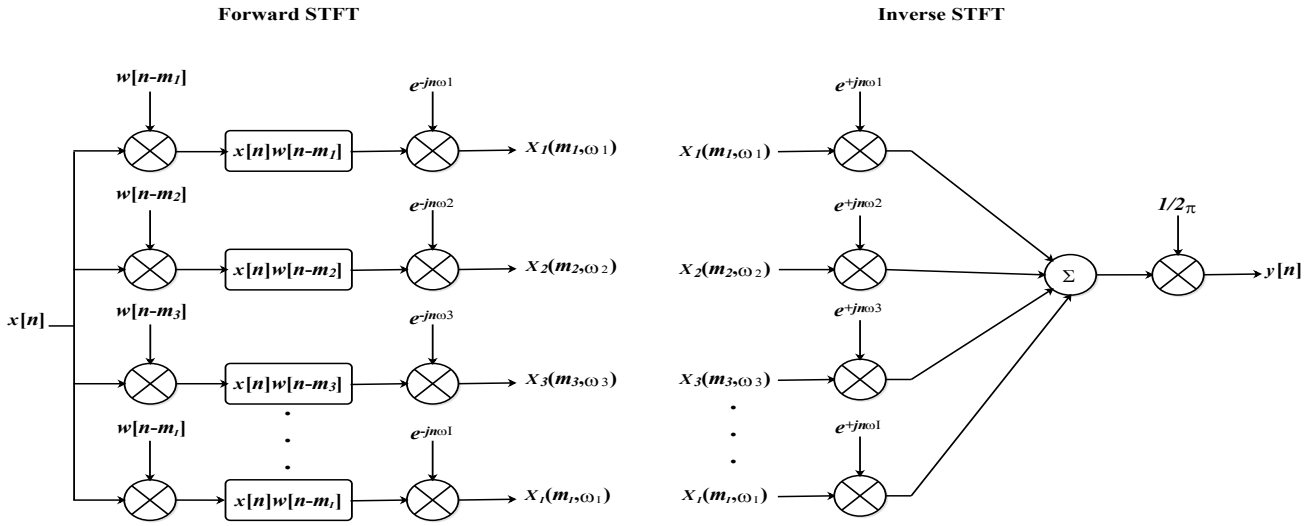


Fig. 3: Forward and Inverse STFTs

III. The Proposed Approach

In the proposed approach, we first convert an input audio signal into time frames. The obtained time frames are then passed through a selected TW before the STFT is applied. After the STFT is applied, the time frames are filtered through a filter bank. The output of the filter bank is passed through an inverse STFT module to generate an output signal. The generated output signal is passed through an OverLap Add (OLA) module to get vocals only. The details are described in the following subsections.

A. Time Framing

For analyzing an audio signal, the most famous and suitable approach is a short term analysis. The reason to perform the short term analysis is because audio signals are usually stable within a short duration of time, e.g., between 10ms to 30ms. Therefore, the audio signals are usually divided into short time frames, although there may be some overlapping in neighboring frames. We set the frame duration to be 20ms, which is the average of 10ms and 30ms. If the duration value is too large, time varying properties of audio signals may not be properly obtained. On the other hand, if the duration value is too small, valid acoustic features may not be extracted [34].

In our proposed approach, we pay attention on the following parameters: frame size, size of frame overlapping, frame step and frame rate.

The frame size is the total number of sampling points in a frame. If the frame size is represented by α, sampling frequency by Ω, and frame duration by d, then the frame size is calculated by the following equation.

$$\alpha = \Omega \times d. \tag{3}$$

Data set that we use in our approach is sampled at 16KHz. Therefore, the frame size calculated by Eq. 3 is 320 sample points in our approach.

The size of frame overlapping (i.e., $\bar{f}$) is calculated by the following equation.

$$\bar{f} = \frac{\alpha}{2}. \tag{4}$$

It is very common that neighboring frames are usually identical. Overlapping of up to $\frac{1}{2}$ of data is still common between a frame and its neighboring frame. To save the computational time, we set the overlapping data to be $\frac{2}{3}$ of the sample points. Therefore, the frame overlapping calculated by Eq. 4 is 160 sample points in our approach.

The frame step (i.e., γ) is computed by the following equation.

$$\gamma = \alpha - \bar{f}. \tag{5}$$

In our proposed approach, the frame step computed by Eq. 5 is 160 sample points.

The frame rate (i.e., β) is the total number of frames per time unit and is computed by the following equation.

$$\beta = \frac{\Omega}{\gamma}. \tag{6}$$

The frame rate calculated by Eq. 6 is 100 frames per second in our proposed approach.

B. Windowing

Window is a mathematical function in signal processing, also known as apodization or tapering function, having zero values outside a defined interval [35]. The applications of Window Functions (WFs) can be found in spectral analysis, filter design and beam-forming [36]. There are different classes of WFs, such as B-spline, polynomial, Hamming, higher-order generalized cosine, power of cosine, adjustable and hybrid windows. Each class of WF contains sub-categories as well [37]. The most commonly used class of WF is B-spline, containing rectangular, triangular, and parzen windows. We use Blackman-Harris Window (BHW), a sub-category of polynomial windows, as a WF in our proposed approach. The reason to choose BHW is because of more cosine terms as compared to the other windows. The additional cosine terms lead to more powerful computations and accurate results. Furthermore, the additional cosine terms reduce the sizes of side lobe areas, and hence ultimately control the leakage of power. Because of smaller side lobe areas, the BHW has been considered to be an ideal to design finite impulse response filters [38].

The WF is used to eliminate the effects of signals before and after a specified interval. When a time frame of an input signal is multiplied with the WF, the effects of discontinuity at each corner of the time frame can be minimized. This multiplication of the WF is performed by modifying Eq. 2 to

$$X_i(m_i, \omega_i) = \sum_{i=1}^{I} \sum_{n=0}^{N-1} x_i[n] w[n - m_i] e^{-j\omega_i n}, \tag{7}$$

where $x_i[n]$ represents a time frame of the input signal $x[n]$, m_i represents the corresponding time, ω_i represents the corresponding frequency, i is the time frame index, $X_i(m_i, \omega_i)$ is the STFT of the $x_i[n]$, I is the total number of time frames, and N is maximum number of sampling points in a time frame (i.e., the frame size α).

In this paper, $w[n]$ is the BHW function and is represented by the following equation [39]:

$$\begin{aligned} w[n] =& a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right) \\ & - a_3 \cos\left(\frac{6\pi n}{N-1}\right). \end{aligned} \tag{8}$$

For each n, the a_0 is 0.35875, a_1 is 0.48829, a_2 is 0.14128, and a_3 is 0.01168 [39].

Fig. 4a shows an input audio signal, Fig. 4b shows a short duration sample of the input audio signal, Fig. 4c shows a sample BHW function, and Fig. 4d shows the output signal as a result of the convolution process (shown in Eq. 7) of the sample signal and the window function.

C. Enhanced Forward Short Time Fourier Transform

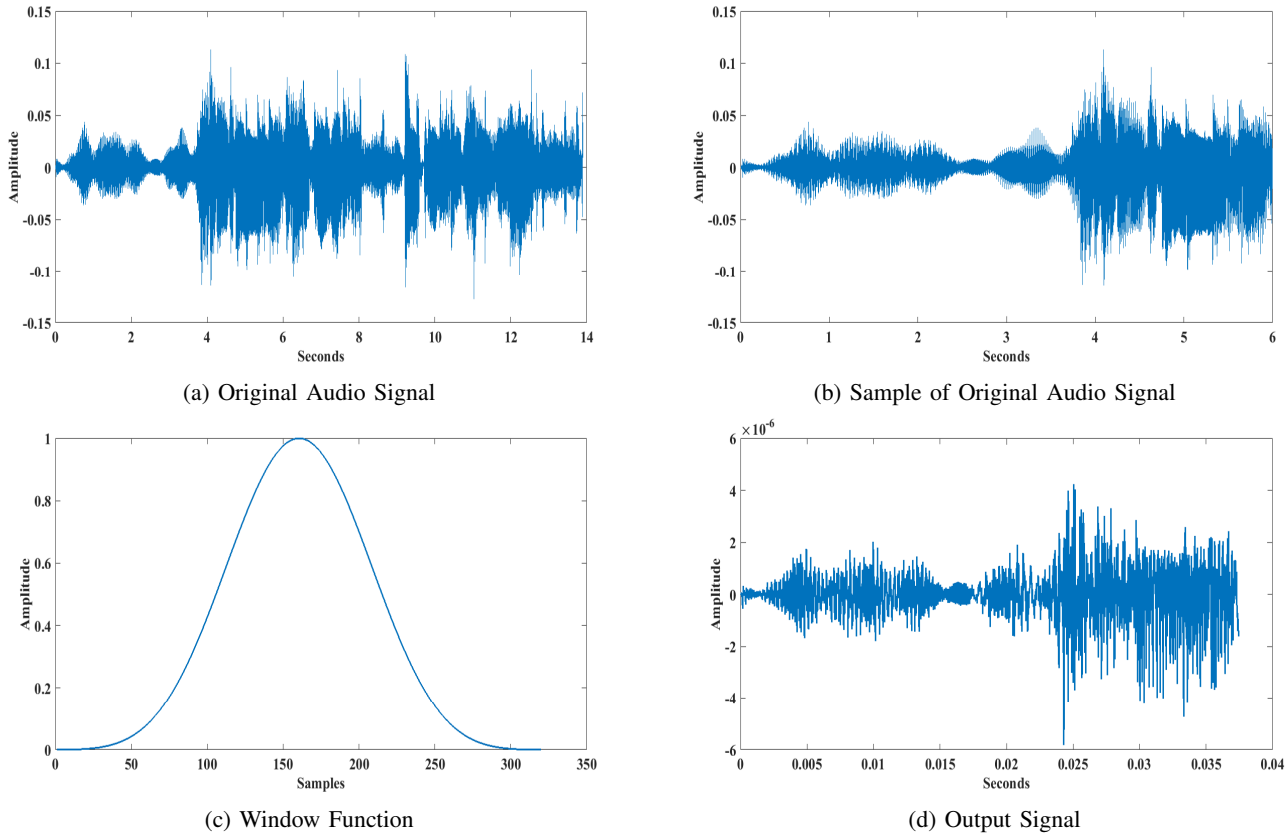
Compared Eq. 2 with the above time framing and windowing processes (Eq.7), it is obvious that both equations are the same. Therefore, we can say that the output obtained after the windowing is actually the output of the STFT. Furthermore, we can improve the spectral information of the windowing output. If the output of the time framing and windowing processes is represented by $X_i(m_i, \omega_i)$, then the spectral resolution of the STFT of the $x[n]$ can be improved till N_1 points by modifying Eq. 7 as follows [40].

$$X_1(m_1, \omega_1) = \sum_{n_1=0}^{N_1-1} x[n_1 + N_1 m_1] w[n_1 - m_1] . e^{\frac{-j2\pi n_1 \omega_1}{N_1}}, \tag{9}$$

where N_1 (i.e., α) represents the total number of samples in the time frame and n_1 represents the index of samples in the time frame.

The Eq. 9 can be used to produce finer resolution till N_2 points as shown in the following equation.

$$\begin{aligned} X_2(m_2, \omega_2) = & \sum_{n_1=0, n_2=0}^{N_1-1 N_2-1} x[n_1 + N_1 n_2 + N_1 N_2 m_2] w[n_2 - m_2] \\ & . e^{\frac{-j2\pi (n_1 + N_1 n_2)\omega_2}{N_1 N_2}}, \end{aligned} \tag{10}$$



(a) Original Audio Signal

(b) Sample of Original Audio Signal

(c) Window Function

(d) Output Signal

Fig. 4: Plots of Signals and Window

where $\omega_2 = \omega_0 + N_2\omega_1$ and ω_0 represents the finer spectral components generated by N_2 times wider time frame. Moreover, $m_2 \in \{1, 2, ..., M_2\}$, where $M_2 = N_2N_3$.

The N_2-points STFT can be rearranged in terms of N_1-points as shown in the following equation.

$$X_2(m_2,\omega_2) = \sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} x[n_1 + N_1n_2m_2]w[n_1 - m_2] .e^{\frac{-j2\pi(\omega_0+N_2\omega_1)(n_1+N_1n_2)}{N_1N_2}}, \tag{11a}$$

$$X_2(m_2,\omega_2) = \sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} x[n_1 + N_1n_2m_2]w[n_1 - m_2] .e^{\frac{-j2\pi\omega_0n_1}{N_1N_2}}.e^{\frac{-j2\pi\omega_0n_2}{N_2}}.e^{\frac{-j2\pi\omega_1n_1}{N_1}}.e^{-j2\pi\omega_1n_2}. \tag{11b}$$

Eq. 11b can also be seen as a two Dimensional Fast Fourier Transform (2D-FFT) as shown in the following equation.

$$X_2(m_2,\omega_2) = \sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} Q[n_1,\omega_0].e^{\frac{-j2\pi\omega_1n_1}{N_1}} .e^{\frac{-j2\pi\omega_0n_2}{N_2}}, \tag{12}$$

where

$$Q[n_1,\omega_0] = x[n_1 + N_1n_2m_2]w[n_1 - m_2].e^{\frac{-j2\pi\omega_0n_1}{N_1N_2}} .e^{-j2\pi\omega_1n_2}. \tag{13}$$

The relationship between $X_1(m_1,\omega_1)$ and $X_2(m_2,\omega_2)$ can be derived by using the multiplication-convolution duality property of the FT as shown in the following equation.

$$X_2(m_2,\omega_2) = \sum_{n_2=0}^{N_2-1}\Bigg[\Bigg(\sum_{n_1=0}^{N_1-1} x[n_1 + N_1n_2m_2]w[n_1 - m_2] .e^{\frac{-j2\pi\omega_1n_1}{N_1}}\Bigg) \circledast \Bigg(\sum_{n_1=0}^{N_1-1} e^{\frac{-j2\pi\omega_0n_1}{N_1N_2}}.e^{\frac{-j2\pi\omega_1n_1}{N_1}}\Bigg)\Bigg].e^{-j2\pi\omega_1n_2}, \tag{14a}$$

$$X_2(m_2,\omega_2) = \sum_{n_2=0}^{N_2-1}\big(X_1(m_2,\omega_1) \circledast H_{12}(\omega_0,\omega_1)\big).e^{-j2\pi\omega_1n_2}. \tag{14b}$$

D. Filter Bank

After applying STFT, we get STFT coefficients matrix. The next step is to select and filter out certain STFT coefficients from the STFT matrix. Time-frequency analysis is very popular in signal processing, which helps in analyzing a signal

in both time and frequency domains, simultaneously. Filters designed through time-frequency analysis help in eliminating the unwanted components of a signal [41], [42]. To eliminate the unwanted STFT coefficients, we construct a filter bank. The filter bank mainly uses Wiener filters, proposed in [17]. The STFT matrix is processed through the filter bank in order to keep the selected STFT coefficients and making rest of the coefficients zero.

The filter bank consists of two independent modules. The first module is based on panning while the second module is based on inter-channel phase difference. Panning is a process to distribute an audio monotonic signal into two or multiple stereo channels. In recording and mixing, panning law is widely used for this purpose [43]. The inter-channel phase differencing is used to minimize the time or phase difference between different channels of a recording [44]. It is possible that the Pan-based module may leave noise, especially the drum residuals and reverberations [45]. Therefore, the main purpose of the inter-channel phase difference module is to eliminate any left noise and reverberations by making phase difference zero between the stereo channels.

E. Pan Filter

Mono signals are panned in both channels to form a stereo mixture. The non-reverberated tracks do not show significant overlapping and it is easier to define a range in order to select their STFT coefficients. However, if the tracks are overlapping, their coefficients may change in time and cannot be estimated correctly, as the coefficients belong to either one source or the other source in the mixture. Here, we assume that we are not using such files in which stereo reverberation is added to one mono track to form a stereo file. As voice is a pure mono track present in an audio signal, therefore, we define a mask in pan filter to select the STFT coefficients of this mono signal only. In our proposed approach, we define the mask for the pan filter, based on the following panning law,

$$\begin{cases} \Delta_i^L = cos\left(\frac{\delta\pi}{2}\right) \\ \Delta_i^R = sin\left(\frac{\delta\pi}{2}\right) \\ \delta = arctan\left(\frac{\Delta_i^R}{\Delta_i^L}\right)\frac{2}{\pi} \end{cases} \tag{15}$$

where δ is the value of pan knob and $\delta \in [0 \quad 1]$, L denotes the left channel, R denotes the right channel, and i denotes the sample index. As the range of the pan is between 0 and 1, the center of this range is 0.5. The mono signal voice is always found around the central pan values; therefore, we set the mask range of the pan filter to be from 0.4 to 0.6. This defined mask passes those STFT coefficients that are within its range, while setting other values to zero. The mask should be defined very carefully. If one of its boundaries is near to zero, it pans maximum values to the left channel while boundary near one means maximum values panned to the right channel [43].

F. Inter-Channel Phase Difference Filter

To further refine the output and to minimize the processing overhead, the output generated by the pan filter is passed through the Inter-Channel Phase Difference Filter (IPDF). Sometimes, it is possible to have monotonic signals with following types of channels:

- Pure stereo channels
- Channels with artificial stereo reverberation
- Identical channels

In the first two types, the phase spectrum of both channels is different while in the third type, both channels contain identical phase spectrum. If the phase spectrum is identical, the phase difference computed by Discrete Fourier Transform (DFT) will be zero, i.e.,

$$|Arg(\text{DFT}(L)) - Arg(\text{DFT}(R))| = 0. \tag{16}$$

In the first two types, the phase difference will always be non-zero, i.e.,

$$|Arg(\text{DFT}(L)) - Arg(\text{DFT}(R))| > 0. \tag{17}$$

However, in the case of artificial stereo reverberation, some DFT coefficients may have identical phases, which make it easy to differentiate. Therefore, IPDF performs two tasks, i.e., firstly, classify the STFT coefficients as either pure stereo or non/artificial stereo, and secondly, minimize the noise effect, if any. In the case of pure non-stereo channels, only one channel is targeted, which minimizes the computing load. In the case of artificial stereo, it is treated just like pure stereo signal. To minimize any remaining noise effect, we define a mask to perform further filtering. The range of the mask is defined from $-\pi$ to $+\pi$. If the coefficients have the phase values around the zero, then it can be assumed that the input signal is a pure non-stereo track which is mirrored in both channels with a constant ratio. If the phase values are not around zero, it means that artificial reverberations have been introduced and it is an artificial stereo track. This mask also eliminates all those coefficients, having phases out of this range.

G. Signal Reconstruction

In this last step, first the inverse of filtered STFT coefficients is calculated. The inverse procedure starts with the inverse STFT of $X_i(m_i, \omega_i)$, i.e.,

$$x_i[n] = \text{STFT}^{-1}\{X_i(m_i, \omega_i)\}. \tag{18}$$

In the inverse STFT, we first multiply $X_i[m_i, \omega_i]$ with $e^{+j\omega_i t}$ and then divide the result by 2π, in order to get back the product of time frames (i.e., $x_i[n]$) and window function (i.e., $w[n]$), i.e.,

$$x_i[n]w[n] \qquad 0 \leq n \leq N-1. \tag{19}$$

The final step is to add all the processed frames to get the extracted vocal back. For this purpose, we use OverLap Add (OLA) technique, which adds partially overlapped frames together. For this process, we use Eq. 19 with some modification, i.e.,

$$y[n] = \sum_{i=1}^{I} \sum_{n=0}^{N-1} x_i[n]w[n - m_i]. \tag{20}$$

Here we keep the window size the same as we previously have used; otherwise reconstruction results will be different. Due to multiple time frames, each time frame needs to be convolved with the same BHW. In terms of different time frames, the signal $x[n]$ can be rewritten as

$$x_i[n] = \begin{cases} x[n + i(N-1)], & n = 0, 1, 2, ..., N-1 \\ 0, & \text{Otherwise.} \end{cases}, \quad (21)$$

$$x[n] = \sum_i x_i[n - i(N-1)]. \quad (22)$$

Based on Eq. 22, the Eq. 20 can be rewritten as

$$y[n] = \Big(\sum_i x_i[n - i(N-1)] \Big) w[n] = \sum_i y_i[n - i(N-1)]. \quad (23)$$

The entire procedure of the proposed approach as described in the above subsections is summarized in Algorithm 1.

Algorithm 1: Proposed Algorithm

Input: $x[n]$ - Input Signal
Output: $y[n]$ - Extracted Vocal Signal/Data
Distribute $x[n]$ into $X_i[n]$
where
$x[n] = \sum_{i=1}^{I} X_i[n]$
while *True* **do**
  $\bar{X}_i = X_i(n)w(n)$
end
Calculate Forward STFT $X_i[m_i, \omega_i]$ using (9) to (14b)
while *True* **do**
  $\bar{\bar{X}}_i[m_i, \omega_i] = \text{PFB}(X_i[m_i, \omega_i])$
  $X_i[m_i, \omega_i] = \text{IPDFB}(\bar{\bar{X}}_i[m_i, \omega_i])$
end
Calculate Inverse STFT $x_i[n]$ using (18)
Obtain $y[n]$ via (23)
while $0 \leq n \leq N-1$ **do**
  $y[n] = \sum_i y_i[n - i(N-1)]$
end

IV. Experimental Setup

In this section, we evaluate the performance of our proposed approach in two phases. The evaluation of the first and second phases is performed based on the detection of the singing vocal in a test audio sample and the extraction of singing vocal from background music, respectively. The proposed approach is evaluated on a publicly available data sets, i.e., TIMIT and MIR-1K [46]. As compared to the other publicly available data sets, TIMIT and MIR-1K are basically designed to extract singing vocals. There are almost 100 random mixtures of musical instruments. The mixtures also contain male-female singing and speech voices, with $16KHz$ sampling rate. The duration of samples ranges from 4 to 13 seconds. These samples are mixtures of vocal and musical instruments and recorded at Texas Instruments, Inc (TI), transcribed at Massachusetts Institute of Technology (MIT) and verified by National Institute of Standards and Technology (NIST). The evaluation phases are illustrated in Fig. 5.

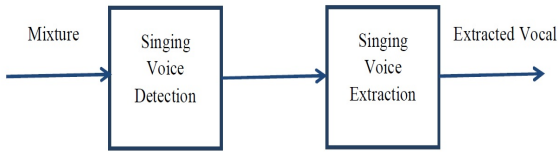


Fig. 5: Evaluation Phases

A. Performance Evaluation of Singing Vocal Detection

In this phase, we use twenty different samples from MIR-1K and TIMIT data sets. The samples contain both male and female voices with varying durations. In the selected samples, the singing vocals and background music are mixed at $-5dB$, $0dB$, and $5dB$ Signal-to-Noise-Ratio (SNR). The performance evaluation is based on a three-level accuracy, i.e., Fair Detection (FD), Better Detection (BD) and Accurate Detection (AD). The FD represents the accuracy percentage in terms of frames, classified as vocal frames over all the sample frames. The BD represents the accuracy percentage in terms of frames, classified as vocal frames over all the FD classified frames. The AD represents the overall accuracy percentage in terms of frames, classified as vocal frames over all the BD classified frames. To classify the frames as vocal and/or non-vocal, the Viterbi algorithm [47] is modified and used iteratively to produce the three-level accuracy.

Fig. 6 shows the performance of vocal detection. The algorithm is evaluated for all selected samples. As shown in Fig. 6, the accuracy level of detection increases with an increased number of iterations of modified Viterbi algorithm.

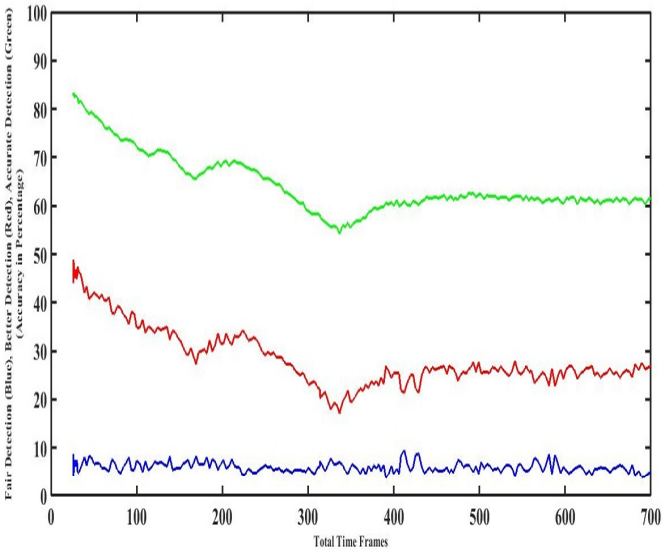


Fig. 6: Accuracy Performance of Voice Detection

In Fig. 6, the blue, red and green lines show the first, second and third iterations, respectively. Here, we consider the blue line as the performance of the original form of Viterbi algorithm. It is very clear that the performance of classification increases with the iterative approach. The experimental results show that the iterative approach works better in the presence of multiple musical instruments in the background.

B. Performance Evaluation of Singing Vocal Extraction

For singing vocal extraction, we choose frame size (i.e., α) as 320 for a better perceived quality of the output sound. We set hop size as $\frac{\alpha}{4}$. The BHW is used as the window function, as it shows good performance with reduced spectral leakage. Fig. 7a shows a test sample, mixed with both music and singing vocals while Fig. 7b shows the singing vocal extracted from the selected test sample. In Fig. 7, the horizontal and vertical axes represent time duration and amplitude of the selected test sample, respectively. It can be seen clearly in Fig. 7, that the maximum number of frequencies is omitted from the selected test sample. The spectrograms of both channels of the selected test sample and the output of the filter bank are shown in Fig. 8. Fig. 8a and Fig. 8b show the spectrograms of the left and right channels of the selected test sample, respectively. Fig. 8c and Fig. 8d show the spectrograms of the left and right channels, respectively, after passing through the pan filter. Fig. 8e and Fig. 8f show the spectrograms of the left and right channels, respectively, after passing through the IPDF. The horizontal and vertical axes represent the total number of frames and the frequency of the samples, respectively.

We measure Signal to Interference Ratio (SIR) and Signal to Distortion Ratio (SDR) to evaluate the performance of our proposed approach. Both SDR and SIR are quite similar. The SDR measures the total amount of distortion, introduced to the original signal by both interfering signals and the processing algorithms while the SIR focuses on distortions, introduced by interfering signals only. For input signal $x[n]$, the total relative distortion (i.e., D) can be measured by the following equation.

$$D \triangleq \frac{\|\hat{x}[n]\|^2 - \mid \langle \hat{x}[n], \frac{x[n]}{\|x[n]\|} \rangle \mid^2}{\mid \langle \hat{x}[n], \frac{x[n]}{\|x[n]\|} \rangle \mid^2}, \tag{24}$$

$$\hat{x}[n] = \left\langle \hat{x}[n], \frac{x[n]}{\|x[n]\|} \right\rangle \frac{x[n]}{\|x[n]\|} + e, \tag{25}$$

where $\|.\|^2$ is the second norm, $\hat{x}[n]$ is the estimated signal, D corresponds to the ratio of energy of two norms in the decomposition and e is the total relative energy.

The e can be computed by the following equation.

$$e = e_i + e_n + e_a, \tag{26}$$

where e_i, e_n and e_a are the relative energies of interference, noise and artifacts, introduced in the source signal, respectively.

The e can also be computed by the following equation.

$$\|e\|^2 = \|\hat{x}[n]\|^2 - \left| \left\langle \hat{x}[n], \frac{x[n]}{\|x[n]\|} \right\rangle \right|^2. \tag{27}$$

If the estimated signal is orthogonal to the original signal, then $\|\langle \hat{x}[n], x[n] \rangle\| \to 0$, which makes $D \to +\infty$. Thus, the SDR can be computed by the following equation.

$$SDR \triangleq 10 log_{10} D^{-1}. \tag{28}$$

The relative distortion due to the interference (i.e., D_i) and SIR can be estimated by the following equation.

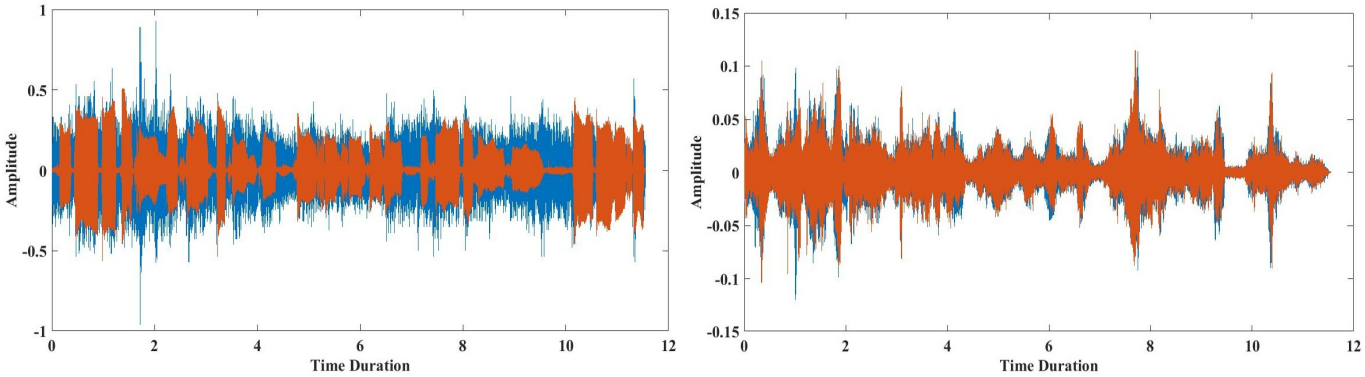
$$D_i \triangleq \frac{\|e_i|^2}{\left| \left\langle \hat{x}[n], \frac{x[n]}{\|x[n]\|} \right\rangle \right|^2}, \tag{29}$$

$$SIR \triangleq 10 log_{10} D_i^{-1}. \tag{30}$$

To compute performance measurements, such as the SIR and SDR, we use BSS eval toolbox [49]. Table I summarizes the computational results, based on the SIR and SDR. To test the performance of our proposed approach, we use various combinations of audio signals and music, e.g., human speech mixed with trumpet, human singing voice mixed with trumpet, etc. For a fair comparison, we select a BSS technique based on the spatial covariance, presented in [48]. This targeted work separates the original sound sources from a mixture of vocal and music by using the spatial features, which is common to our proposed approach. For completeness, we follow the same test conditions, both for targeted and our proposed approach. Columns two to five represent the values of SIR and SDR, computed by comparing the output of the targeted and proposed approaches, with the original sources, respectively. The mixtures are also compared with the original sources for a better baseline, as shown in sixth and seventh columns. A significant increase in both the SIR and SDR indicates a better performance. It is clearly shown in Table I that our proposed approach outperforms the targeted approach by achieving a higher increase in both the SIR and SDR values. In the last two columns, only one of the sources, i.e., audio signal in terms of speech or singing vocal or music only is compared with the mixture at a time. As can be seen, our proposed approach outperforms the simple comparison by showing a significant improvement, both in the SIR and SDR values. For different test samples, the performance comparisons between the proposed approach and the approach presented in [48], in terms of SIR and SDR values, are shown in Fig. 9a and Fig. 9b, respectively.

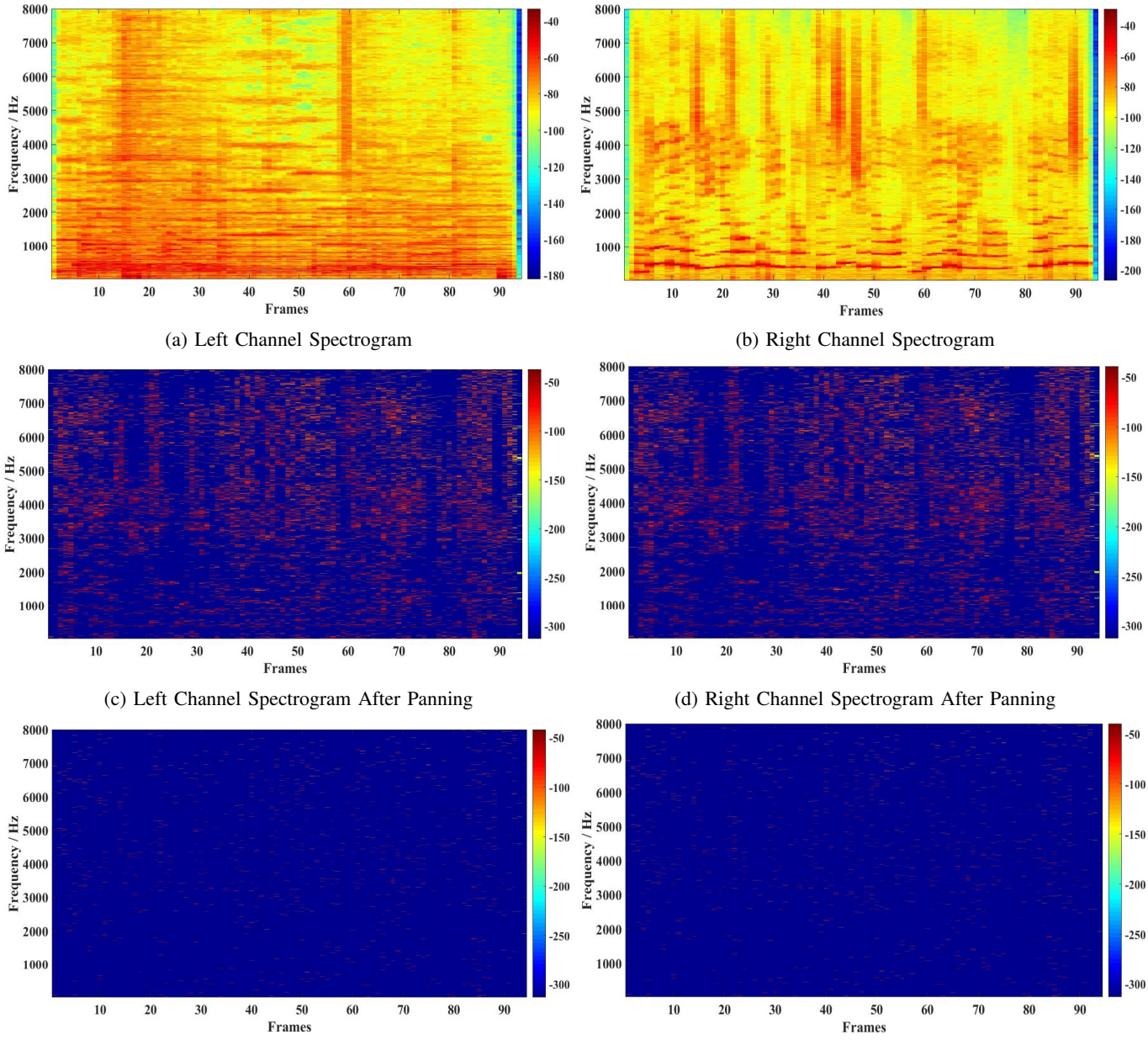
V. Conclusion

In this paper, we have dealt with the blind audio source separation problem by proposing an approach to remove the background music and leaving only foreground vocals in the stereophonic audio signals using modified STFT. Signals have been observed in the time-frequency domain. The TIMIT and MIR-1K data sets have been used for experimental purpose and contain synthetically created mixtures of songs. The evaluation has been performed using the SIR and SDR metrics. We have also shown the performance of the proposed approach on the original mixtures with significant improvements. The proposed approach may show degraded performance in situations where the musical instruments are



(a) Input Stereo Signal (b) Output Stereo Signal

Fig. 7: Stereo Outputs



(a) Left Channel Spectrogram

(b) Right Channel Spectrogram

(c) Left Channel Spectrogram After Panning

(d) Right Channel Spectrogram After Panning

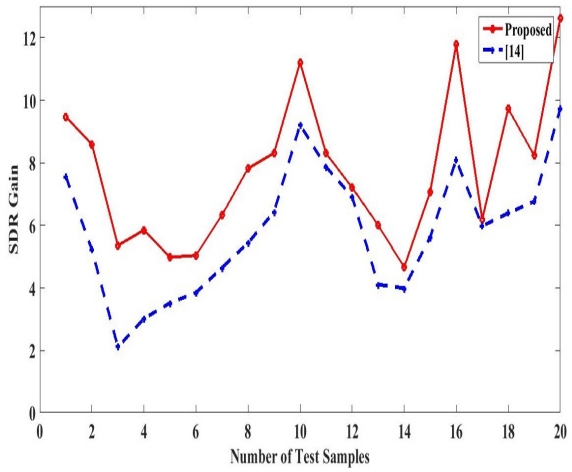
(e) Left Channel Spectrogram After Inter-Phase Difference

(f) Right Channel Spectrogram After Inter-Phase Difference

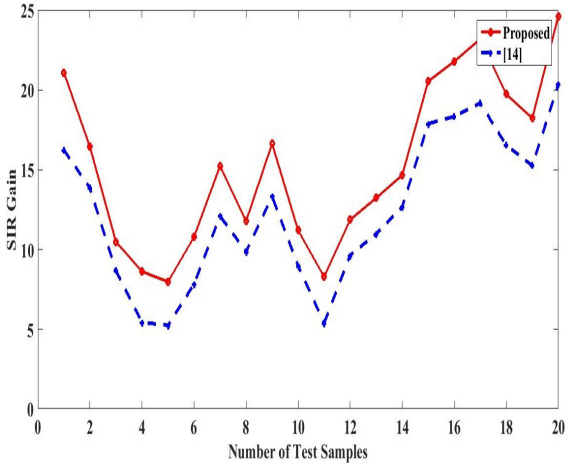
Fig. 8: Spectrogram, Pann and Inter-Phase Difference

Mixtures	SIR [48]	SDR [48]	SIR (Proposed)	SDR (Proposed)	SIR (Mix)	SDR (Mix)
Speech and Trumpet	16.22	7.56	21.09	9.47	7.98 and -4.71	7.98 and -4.71
Singing and Trumpet	13.85	5.22	16.44	8.59	-2.04 and 2.20	-2.05 and 2.19
Speech dB (Mean)	8.67	2.11	10.49	5.36	1.21 and -0.70	1.21 and -0.70
Music dB (Mean)	5.45	3.02	8.63	5.85	0.38 and 0.16	0.38 and 0.16
Singing and Cello	5.26	3.52	7.99	4.98	-3.00	-3.00
Speech and Bubbles	7.83	3.85	10.79	5.04	7.22	7.19

TABLE I: SIR and SDR Comparisons



(a) SIR Performance Graph



(b) SDR Performance Graph

Fig. 9: SIR and SDR Performance Graphs

mostly panned near the center, where the voice resides and as a result, the background music may dominate the singing vocal. In such a situation, there will be some interference between the music and the voice, and the background music cannot be removed completely. Future work needs to deal with scenarios having maximum overlapping of vocals and music or where the musical frequencies are dominating. Moreover, the proposed approach needs to be tested with modifications on test samples, having mixture of professional background music and professional singer's vocals.

REFERENCES

[1] S. Mirzaei, Y. Norouzi *et al.*, "Two-stage blind audio source counting and separation of stereo instantaneous mixtures using Bayesian tensor factorisation," *Signal Processing, IET*, vol. 9, no. 8, pp. 587–595, 2015.

[2] B. Gao, W. L. Woo, and S. S. Dlay, "Unsupervised single-channel separation of nonstationary signals using Gammatone filterbank and Itakura–Saito nonnegative matrix two-dimensional factorizations," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 60, no. 3, pp. 662–675, 2013.

[3] B. Peng, W. Liu, and D. P. Mandic, "Design of oversampled generalised Discrete Fourier Transform filter banks for application to subbandbased blind source separation," *Signal Processing, IET*, vol. 7, no. 9, pp. 843–853, 2013.

[4] C. Osterwise and S. L. Grant, "On over-determined frequency domain BSS," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 5, pp. 956–966, 2014.

[5] S. H. Sardouie, M. B. Shamsollahi, L. Albera, and I. Merlet, "Denoising of ictal EEG data using semi-blind source separation methods based on time-frequency priors," *Biomedical and Health Informatics, IEEE Journal of*, vol. 19, no. 3, pp. 839–847, 2015.

[6] B. Rivet, "Source separation of multimodal data: a second-order approach based on a constrained joint block decomposition of covariance matrices," *Signal Processing Letters, IEEE*, vol. 22, no. 6, pp. 681–685, 2015.

[7] S. Lee and H.-S. Pang, "Multichannel non-negative matrix factorisation based on alternating least squares for audio source separation system," *Electronics Letters*, vol. 51, no. 3, pp. 197–198, 2015.

[8] G.-S. Fu, R. Phlypo, M. Anderson, X.-L. Li, and T. Adali, "Blind source separation by entropy rate minimization," *Signal Processing, IEEE Transactions on*, vol. 62, no. 16, pp. 4245–4255, 2014.

[9] J. Hofmanis, O. Caspary, V. Louis-Dorr, R. Ranta, and L. Maillard, "Denoising depth EEG signals during DBS using filtering and subspace decomposition," *Biomedical Engineering, IEEE Transactions on*, vol. 60, no. 10, pp. 2686–2695, 2013.

[10] O. Tichy and V. Smidl, "Bayesian blind separation and deconvolution of dynamic image sequences using sparsity priors," *Medical Imaging, IEEE Transactions on*, vol. 34, no. 1, pp. 258–266, 2015.

[11] J. Nikunen and T. Virtanen, "Direction of arrival based spatial covariance model for blind sound source separation," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 3, pp. 727–739, 2014.

[12] B. Liu, V. G. Reju, A. W. Khong, and V. V. Reddy, "A GMM post-filter for residual crosstalk suppression in blind source separation," *Signal Processing Letters, IEEE*, vol. 21, no. 8, pp. 942–946, 2014.

[13] B. Liu, V. G. Reju, and A. W. Khong, "A linear source recovery method for underdetermined mixtures of uncorrelated AR-model signals without sparseness," *Signal Processing, IEEE Transactions on*, vol. 62, no. 19, pp. 4947–4958, 2014.

[14] S. Hosseini and Y. Deville, "Blind separation of parametric nonlinear mixtures of possibly autocorrelated and non-stationary sources," *Signal Processing, IEEE Transactions on*, vol. 62, no. 24, pp. 6521–6533, 2014.

[15] Y. Zhang, P. Candra, G. Wang, and T. Xia, "2-D entropy and Short-Time Fourier Transform to leverage gpr data analysis efficiency," *Instrumentation and Measurement, IEEE Transactions on*, vol. 64, no. 1, pp. 103–111, 2015.

[16] G. Okopal, S. Wisdom, and L. Atlas, "Speech analysis with the strong uncorrelating transform," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 11, pp. 1858–1868, 2015.

[17] J. Le Roux and E. Vincent, "Consistent Wiener filtering for audio source separation," *Signal Processing Letters, IEEE*, vol. 20, no. 3, pp. 217–220, 2013.

[18] Y. G. Jin, J. W. Shin, and N. S. Kim, "Spectro-temporal filtering for multichannel speech enhancement in Short-Time Fourier Transform

domain," *Signal Processing Letters, IEEE*, vol. 21, no. 3, pp. 352–355, 2014.

[19] R. E. Turner and M. Sahani, "Time-frequency analysis as probabilistic inference," *Signal Processing, IEEE Transactions on*, vol. 62, no. 23, pp. 6171–6183, 2014.

[20] L. Stankovic, S. Stankovic, and M. Dakovic, "From the STFT to the Wigner distribution [lecture notes]," *Signal Processing Magazine, IEEE*, vol. 31, no. 3, pp. 163–174, 2014.

[21] V.-K. Mai, D. Pastor, A. Aïssa-El-Bey, and R. Le-Bidan, "Robust estimation of non-stationary noise power spectrum for speech enhancement," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 4, pp. 670–682, 2015.

[22] P. Flandrin, "Time–frequency filtering based on spectrogram zeros," *Signal Processing Letters, IEEE*, vol. 22, no. 11, pp. 2137–2141, 2015.

[23] J. Zheng, T. Su, L. Zhang, W. Zhu, and Q. H. Liu, "ISAR imaging of targets with complex motion based on the chirp rate–quadratic chirp rate distribution," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 11, pp. 7276–7289, 2014.

[24] W.-Q. Wang, "Large time-bandwidth product MIMO radar waveform design based on chirp rate diversity," *Sensors Journal, IEEE*, vol. 15, no. 2, pp. 1027–1034, 2015.

[25] Y. Doweck, A. Amar, and I. Cohen, "Fundamental initial frequency and frequency rate estimation of random-amplitude harmonic chirps," *Signal Processing, IEEE Transactions on*, vol. 63, no. 23, pp. 6213–6228, 2015.

[26] G. Bao, Y. Xu, and Z. Ye, "Learning a discriminative dictionary for single-channel speech separation," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 7, pp. 1130–1138, 2014.

[27] S. K. Jain, S. Singh, and J. G. Singh, "An adaptive time-efficient technique for harmonic estimation of nonstationary signals," *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 8, pp. 3295–3303, 2013.

[28] Q. Yin, L. Shen, M. Lu, X. Wang, and Z. Liu, "Selection of optimal window length using STFT for quantitative SNR analysis of LFM signal," *Systems Engineering and Electronics, Journal of*, vol. 24, no. 1, pp. 26–35, 2013.

[29] V. Arora and L. Behera, "Musical source clustering and identification in polyphonic audio," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 6, pp. 1003–1012, 2014.

[30] A. Hyvärinen, P. Ramkumar, L. Parkkonen, and R. Hari, "Independent component analysis of Short-Time Fourier Transforms for spontaneous EEG/MEG analysis," *NeuroImage*, vol. 49, no. 1, pp. 257–271, 2010.

[31] Wikipedia, "Short-Time Fourier Transform," 2017. [Online]. Available: https://en.wikipedia.org/wiki/Short-time_Fourier_transform

[32] T. Otsuka, K. Ishiguro, H. Sawada, and H. G. Okuno, "Bayesian nonparametrics for microphone array processing," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 2, pp. 493–504, 2014.

[33] T. Otsuka, K. Ishiguro, T. Yoshioka, H. Sawada, and H. G. Okuno, "Multichannel sound source dereverberation and separation for arbitrary number of sources based on Bayesian nonparametrics," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 2218–2232, 2014.

[34] N. Morgan, "Deep and wide: multiple layers in automatic speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 7–13, 2012.

[35] E. W. Weisstein, *CRC concise encyclopedia of mathematics*. CRC press, 2002.

[36] C. Roads, *Microsound*. MIT press, 2004.

[37] K. Toraichi, M. Kamada, S. Itahashi, and R. Mori, "Window functions represented by B-spline functions," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 1, pp. 145–147, 1989.

[38] J. O. Smith, *Spectral audio signal processing*. W3K, 2011.

[39] Wikipedia, "Blackman-Harris Window," 2017. [Online]. Available: https://en.wikipedia.org/wiki/Window_function#Blackman.E2.80.93Harris_window

[40] B. Kim, S.-H. Kong, and S. Kim, "Low computational enhancement of STFT-based parameter estimation," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 9, no. 8, pp. 1610–1619, 2015.

[41] E. Sejdić, I. Djurović, and J. Jiang, "Time–frequency feature representation using energy concentration: An overview of recent advances," *Digital Signal Processing*, vol. 19, no. 1, pp. 153–183, 2009.

[42] L. Cohen, *Time-frequency analysis*. Prentice hall, 1995, vol. 299.

[43] B. Owsinski, *The mixing engineer's handbook*. Nelson Education, 2013.

[44] W. Hoeg and T. Lauterbach, *Digital audio broadcasting: principles and applications of digital radio*. John Wiley & Sons, 2004.

[45] E. Perez_Gonzalez and J. Reiss, "A real-time semiautonomous audio panning system for music mixing," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, pp. 1–10, 2010.

[46] J. S. Garofalo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "The DARPA TIMIT acoustic-phonetic continuous speech corpus cdrom," *Linguistic Data Consortium*, 1993.

[47] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *Information Theory, IEEE Transactions on*, vol. 13, no. 2, pp. 260–269, 1967.

[48] G. Wolf, S. Mallat, and S. Shamma, "Rigid motion model for audio source separation," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1822–1831, 2016.

[49] C. Févotte, R. Gribonval, and E. Vincent, "Bss_eval toolbox user guide–revision 2.0," 2005.