Robotic Sound Source Separation using Independent Vector Analysis

Martin Rothbucher, Christian Denk, Martin Reverchon, Hao Shen and Klaus Diepold

Technical Report

Robotic Sound Source Separation using Independent Vector Analysis

Martin Rothbucher, Christian Denk, Martin Reverchon, Hao Shen and Klaus Diepold

March 30, 2014

Institute for Data Processing
Technische Universität München

Contents

Abstract

Beside haptic and vision, mobile robotic platforms are equipped with audition in order to autonomously navigate and interact with their environment. Speaker and speech recognition as well as the recognition of different kind of sounds are vital tasks for human robot interaction. In situations where more than one sound source is active, the mixture has to be separated before being passed to the reasoning unit. Independent Component Analysis (ICA) has been proposed to solve the blind source separation problem. For audio signals however, ICA cannot be applied directly. Due to non-instantaneous mixtures in the time domain, the problem is usually transferred to multiple separately performed ICAs in the frequency domain, which causes the well-known permutation problem. For robotic sound separation, in this paper we propose a method called Independent Vector Analysis (IVA) to separate audio mixtures while avoiding the permutation problem. Performance of the method is evaluated for synthetic data as well as for anechoic and echoic recordings. Furthermore, a new method to evaluate the separation results for real recordings is introduced.

1 Introduction

Humans are able to engage in conversations at a noisy cocktail party. They may understand one particular speaker even though a cacophony of voices is heard around them. This is the well-known “cocktail-party effect”[1].

Technical systems like teleconferecencing systems and mobile robots would benefit of such capabilities in many fields. The improvement of telepresence, high-end hearing aids or speech recognition and auditory scene analysis are areas in which the emulation of humans’ extraordinary perception is needed, however not yet available.

To emulate some of these perception capabilities and thus enabling a technical system to focus on one specific sound source within a mixture, source separation techniques are required that process the observed mixture into its underlying signal parts. In literature, this separation problem is often called Blind Source Separation (BSS), which seeks to separate the sound sources within a mixutre. The term BSS refers to methods for the estimation of source signals using only information acquired by the analysis of recorded mixtures. This excludes a priori information about e.g. the frequency characteristics, the location or the mixing process. Yet some information like the location can in some cases be obtained by analyzing the physical properties of sound and be used to improve the performance of BSS algorithms.

Up to now, numerous algorithms for the BSS problem have continuously evolved since the 90s. These algorithms can be divided into two groups: The first group uses spatial information, the second group statistical information of the signals to achieve separation.

Beamforming for example applies a spatial filter to separate signals which originate from different locations by linearly combining spatially sampled time series of the sensor data, similar as a FIR-Filter would do [17]. Beamforming can be combined with Direction of Arrival (DOA) estimation algorithms like MUSIC [16] or ESPRIT [15] to attain segregated signals. Another algorithm that exploits spatial information is applying spatial spectral masking. Spatial spectral masking first finds the DOA for different sources for each frequency. Afterwards, a spatial filter in the frequency domain is applied [12]. However, the algorithms that are based on spatial information lack the possibility to separate signals that are mixed in echoic environments (e.g. an office room).

Beside the algorithms that use spatial information, there are methods based on the evaluation of the signals’ statistics. Independent Component Analysis (ICA) [8, 5] is one of the most popular approach of this group of algorithms that successfully perform the BSS for instantaneous mixtures. However, ICA cannot separate convolutive mixtures. To overcome this problem, statistical separation methods that are based on ICA extend its capabilities to tackle convolutive mixtures. This is usually done by transforming the convolutive mixture

into frequency domain, which results in an instantaneous mixture model per frequency bin. Then, algorithms like Multidimensional Independent Component Analysis (MICA) [2] or Independent Subspace Analysis (ISA, the ML-equivalent to MICA) [4] seek to group dependent scalar mixtures and thus achieve the desired separated signals.

When using these algorithms for speech separation, several problems arise. First, the mixing model of ISA and MICA is not designed to fit realistic mixing conditions for speech signals in reverberant environments. For speech mixtures, the assumption holds that only signal parts within the same frequency interval are mixed due to the signal propagation in real environments which usually do not alter the frequency of certain signal parts. Therefore, the mixing model of MICA/ISA does not perfectly fit for speech separation, as it allows both for arbitrary mixing of frequencies and different numbers of scalar variables within the outcomes. MICA/ISA is actually designed to have one large mixing layer for all frequencies, i.e. the highest frequencies are allowed to mix with the lowest frequencies. Second, due to these extensive mixture models, MICA and ISA are complex and computationally expensive algorithms, which makes it difficult to achieve real-time speech separation on a robot.

Recently, a promising approach called Independent Vector Analysis (IVA) has been proposed to inherently solve the permutation problem [10]. Although the basic ideas behind IVA resemble MICA and ISA, its mixing model is designed especially for the task of audio source separation, grouping dependent frequencies of sources together within the separation step. Also, IVA is not as computationally expensive as MICA or ISA, as the mixing model is simpler allowing for fast computation of the outcomes.

In this paper, necessary principles of IVA in sufficient detail to implement them, are discussed. Then, the performance of IVA is evaluated, which involves various cost functions, the impact of the Short-Time Fourier Transform and computation time. IVA is tested with real-world records in both a semi-anechoic room with low reverberations and in an echoic classroom environment.

As we carried out the evaluation in real room environments, we also propose a new evaluation function for the BSS evaluation toolbox that directly takes the possible distortions into account that are induced by the room environment and allows for a fast an fair comparison of separation algorithms.

The paper is organized as follows: The first section provides an overview on how to model sound mixing. Then, the necessary preprocessing steps for source separation are described, followed by an detailed description of IVA. The paper concludes with an explanation of the usual performance criteria in the field of blind source separation and results from our own recordings.

2 The Sound Mixing Model

For a scenario of i active sound sources $s_1(t), \ldots, s_i(t)$ and j microphones that capture the mixtures $x_1(t), \ldots, x_j(t)$, the most intuitive mixture model is the Instantaneous Mixture Model,

$$x_j(t) = \sum_{i=1}^{N} h_{ij} \cdot s_i(t), \tag{2.1}$$

where, h_{ij} describes an attenuation factor due to different volumes of the sources at each microphone. This instantaneous mixture model is one of the basic models used for Independent Component Analysis (ICA). For these instantaneous mixtures, a huge variety of separation algorithms exist and have been tested, for example FastICA. An overview can be found in [8] and [3].

For audio signals however, there are better mixing models that enable superior separation results by taking physical properties of sound into account. Besides the aforementioned volume differences, it is advantageous to also exploit time lags within the recordings, which occur between the microphones.

Furthermore, reverberant rooms often render instantaneous mixture models useless because of time-delayed and scaled versions (echoes) in the microphone recordings.

Scaling, time-delay and echoes can be altogether described as a linear filter which is applied to the sound source. Applying a filter mathematically means convolving the original sound source with the corresponding filter that is dependent on the position of the sound source and the microphone within a reverberant room. According to our previous instantaneous mixing model, the filter functions are denoted by $h_{ij}(t)$. For microphone j, the recorded signal is the superposition of the filtered sources:

$$x_j(t) = \sum_{i=1}^{N} h_{ij}(t) * s_i(t), \tag{2.2}$$

where $*$ denotes the convolution operation. In literature, this model is called the *Convolutive Mixture Model*. The convolutive mixture model allows us to give a better description of the mixing process for sound sources and consequently enables us to reach better sound source separation results, but there are two major drawbacks of the convolutive mixture model. One problem of the convolutive mixture model is that it omits sensor noise, as this would require a more complex separation algorithm, which however turned out to improve separaton performance. The other problem is that separation algorithms, like ICA, that are designed for instantaneous noise-free mixtures could not be utilized.

There are a number of algorithms that perform separation on the mixtures directly [13] which are computationally quite expensive. To circumvent this problem, the convolutive mixture model can be transformed to frequency domain, where the convolution operation is described by a multiplication and consequently a convolutive mixture model (2.2) turns into an instantaneous mixture model (2.1).

3 Frequency Domain Transformation

The convolutive mixture model can be transformed to a frequency domain representation by applying the Fourier Transform. Due to Fourier Transform properties, convolution operations in time domain transform to multiplications in the frequency domain. Equation (2.2) is thus transformed to an *Instantaneous Mixture* similar to Equation (2.1):

$$X_j(f) = H_{1j}(f) \cdot S_1(f) + \cdots + H_{Nj}(f) \cdot S_N(f). \tag{3.1}$$

In contrast to the time domain instantaneous ICA model, the mixing coefficients are also dependent on the frequency variable, which renders the direct use of instantaneous mixture algorithms such as FastICA useless. The frequency domain representation of the model can be described by

$$\mathbf{x}(f) = \mathbf{H}(f) \cdot \mathbf{s}(f). \tag{3.2}$$

It is obvious that Equation (3.2) indeed corresponds to the Instantaneous Mixture Model with the flaw that we do not only have one mixing matrix $\mathbf{H}$, but one for each frequency bin $\mathbf{H}(\mathbf{f})$. As shown for ICA [8], we need many sound samples to estimate one mixing matrix. To overcome this problem, we basically assume that the mixing matrices are constant over a certain frequency interval in order to get more than just one sample per mixing matrix.

In addition, the intuitive approach of applying Fourier transform to the whole signal fails in this case, as time information, i.e. the correspondency between time and frequency samples is lost. Instead of Fourier transform, *Short-Time Fourier Transform* is usually applied due to non-stationary of speech signals.

4 Short-Time Fourier Transform

As we know, it is advantageous to perform the sound source separation in frequency domain. One popular method to obtain the frequency domain representation of the mixture is called Short-Time Fourier Transform (STFT). The STFT divides the whole sound mixture into blocks of an a-priori defined number of samples. Each block is individually transformed to the frequency domain using the Discrete Fourier Transform (DFT).

Speech can be considered as a non-stationary process. Therefore, stationarity only can be assumed in short signal blocks of about $10 - 100$ ms, which roughly corresponds to 1024 samples by a sampling rate of 16 kHz.To avoid artifacts within the blocks caused by the fragmentation process, some issues have to be considered. Cutting the sound mixture into blocks leads to an inaccurate frequency spectrum of the mixture due to the so-called spectral leakage effect [7]. By windowing, this effect could be avoided, but in turn, windowing introduces disturbances at the edge of the blocks. To overcome this problem, overlapping window processing as illustrated in Figure 4.2 is used. This procedure allows for an invertible signal transformation of a sound mixture.

A windowing function $w(n)$ for the overlapping windowing process with length L, for example a cosine window, is defined as follows:

$$w(n) = \begin{cases} 0 & |n| > L \\ \cos(n) & |n| \leq L. \end{cases} \tag{4.1}$$

If we slice the mixture $x(n)$ into blocks and apply the windowing with an overlap $S < L$, the slices $x_i(n)$ can be computed by

$$x_i(n) = x(n) \cdot w(n - i \cdot S). \tag{4.2}$$

Subsequently, the DFT is applied to the time intervals $x_i(n)$. For each time interval (block), a frequency representation $x[t, k]$ of the mixture $x(n)$ is generated, where t corresponds to the time index for the signal block and k to the frequencies within the mixture. This way, a time-frequency representation of the mixture, as illustrated in Fig. 4.1, is generated. After the separation of the mixtures, inverse STFT is applied to derive the time domain representation of the separated mixtures. A frequently used method is the so-called overlap-add method, which is defined as follows.
First, the sum of all squared windows $W(n)$ is computed by

$$W(n) = \sum_{i=-\infty}^{\infty} w(n - i \cdot S)^2. \tag{4.3}$$

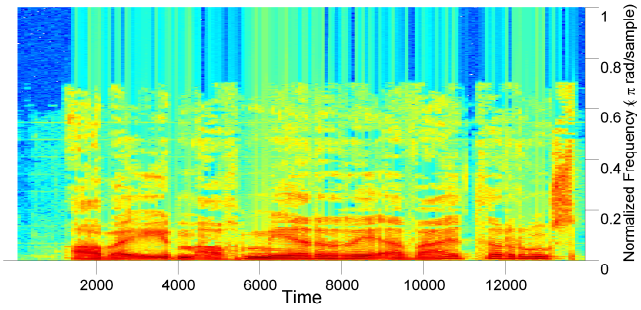

Figure 4.1: Spectrogram: Blue colors correspond to low absolute frequency coefficients, red colors to large ones.

Finally, the time domain representation of the signal is derived by

$$x(n) = \frac{\sum_{i=-\infty}^{\infty} x_i(n) \cdot w(n - i \cdot S)}{W(n)}. \tag{4.4}$$

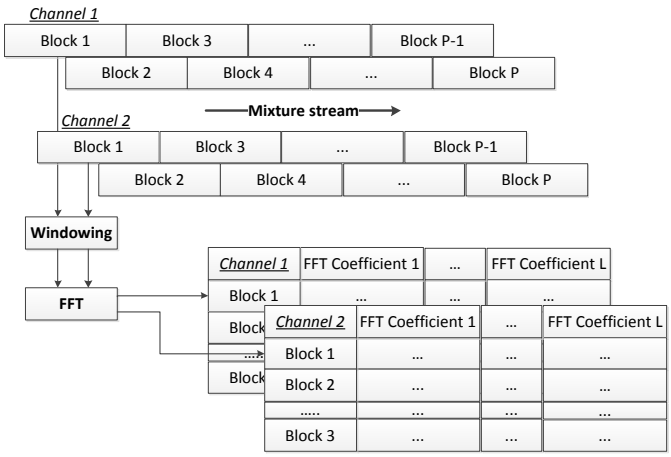

Figure 4.2: Schematic view of the overlapping window processing.

With a STFT of the mixtures, the convolutive mixture (time-domain) can be considered as an instantaneous ICA problem in the frequency domain. For instantaneous ICA, each sample of the observed data can be regarded as a realisation of a scalar random variable. The estimation of the mixing matrices and consequently of the demixing matrices can be achieved by exploiting the statistical properties of the observed mixtures over a large number of sound samples.

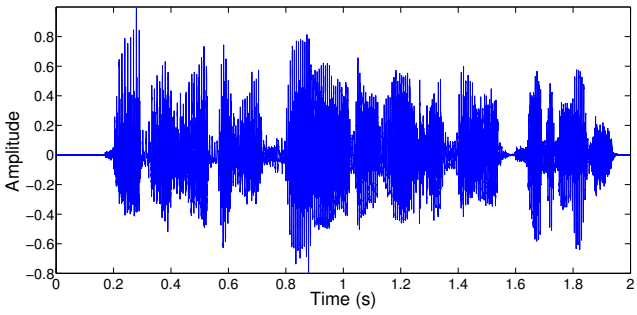

Figure 4.3: STFT

The direct transformation of the convolutive mixtures to the frequency domain brings up the problem of frequency-dependent mixing matrices, which is solved by the STFT.

5 ICA in Frequency Domain

Applying the STFT to the mixture, transforms the convolutive ICA problem into one instantaneous ICA problem per frequency bin. Instantaneous ICA algorithms can unmix the mixture for each frequency bin independently, however, there is a major drawback when treating the convolutive frequency-domain BSS problem as multiple independent ICA problems in the frequency domain: ICA algorithms are only capable of achieving source separation up to an arbitrary permutation of the outcomes [8]. Therefore, the order of the separeted outcomes in each frequency bin is not known which causes problems in combining the individual frequency bins to one complete output signal. There are numerous approaches which seek to align the separated frequencies *after* the separation, promising good separation results, robustness, however, is an issue.

6 IVA Frequency Domain Mixing Model

In contrast to solve the permutation problem after demixing, Independent Vector Analysis (IVA) seeks to avoid the permutation problem within the separation process itself. IVA is designed to separate speech mixtures and is computationally not as intense as ICA extensions like Independent Subspace Analysis or Multidemensional ICA.

IVA uses the same mixture model as frequency domain ICA algorithms, i.e. the mixing is restricted to each frequency bin, and different frequency bins are not allowed to be mixed in this model. Instead of treating each frequency bin as an independent ICA problem, IVA assumes each source to be composed of many scalar sources in the frequency bins, that are dependent on each other, which is the reason why IVA is capable of solving the permutation problem inherently. These sources are called vector sources.

In Figure 6.1, two sources $\mathbf{s}_1$, $\mathbf{s}_2$ and the frequency bins are illustrated. Mixing is constrained to the frequency slices, represented by the mixing matrices $\mathbf{H}^f$. However, the scalar components of the sources are treated as a multivariate vector source that keeps track of the dependencies within each source.

The multivariate source vectors are assumed to be mutually independent. The crucial point in IVA is that within one source, the components are statistically dependent. Taking this into account with a suitable cost function, the IVA algorithm manages to identify the dependent frequency components of each source and thus avoids the problem of finding corresponding frequency components. In order to apply IVA, certain source priors (probability distributions) $p_i(\mathbf{s}_i)$ have to be assumed, where i refers to source i.

The vector source distributions are considered to be mutually independent, i.e.

$$p(\mathbf{s}_1, ..., \mathbf{s}_N) = \prod_{i=1}^{N} p_i(\mathbf{s}_i). \tag{6.1}$$

The dependency among one source implies that this source cannot be factorized, i.e.

$$p(\mathbf{s}) \neq \prod_{\forall i} p(s_i). \tag{6.2}$$

6.1 Whitening in IVA

Prior to the separation process, the frequency bin mixtures are whitened, i.e. they are transformed to uncorrelated mixtures and the bins are assigned to the same variance (power) which is one. Although zero correlation is not equal to independence, whitening simplifies the problem. We refer to [8] for more details on whitening.

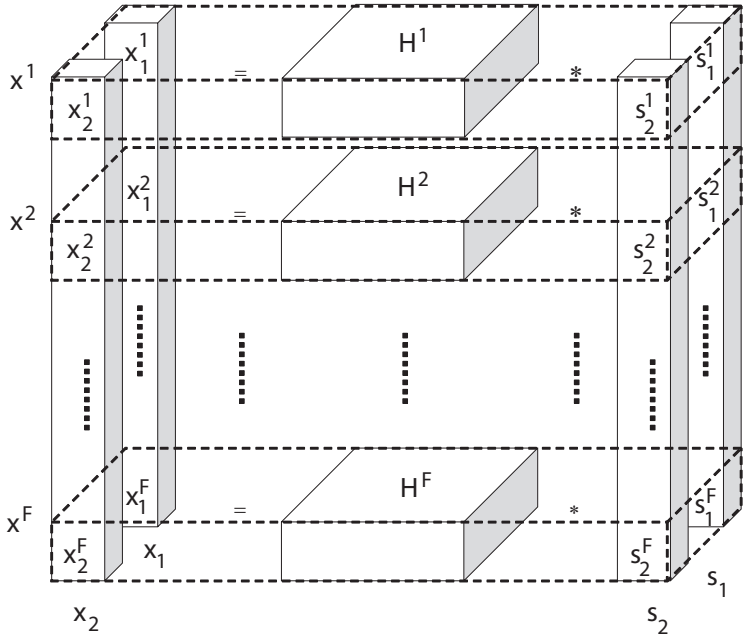

Figure 6.1: IVA Mixture Model in accordance with [10]

For one frequency bin x^f, the mixing process can be described by

$$\mathbf{x}^f = \mathbf{H}^f \mathbf{s}^f. \tag{6.3}$$

A whitened mixture $\mathbf{x}_0^f$ can be retrieved by applying the matrix $\mathbf{Q}^f = (E\{\mathbf{x}^f \mathbf{x}^{fH}\})^{-\frac{1}{2}}$ to x^f [6]:

$$\mathbf{x}_0^f = \mathbf{Q}^f \mathbf{x}^f. \tag{6.4}$$

6.2 IVA Cost Functions

As we know, IVA inherently avoids the permutation problem within the frequency bins. In [8], a maximum-likelihood approach was utilized to separate the sound mixture for ICA. The maximum-likelihood approach seeks to find separation matrices $\mathbf{W}^f$ for each frequency bin which make the separated sources *most likely* with respect to the assumed probability distribution. Therefore, we need suitable source prior estimates.

It was shown that speech signals can be modeled as supergaussian, i.e. sparse distributions after the STFT. Refer to [14] for further details.

For IVA, spherically symmetric Laplacian (SSL) and spherically symmetric exponential norm (SEND) distributions were proposed as source priors. These are sparse distributions, and are mathematically easy to handle as they are based on exponentials. It was

shown in [11] that the distributions allow for good approximation of speech and are capable of keeping depending frequencies together. Recent investigations propose a Gaussian Mixture Model (GMM) as IVA source priors [6], which is capable of modeling speech better than SSL or SEND. However, GMMs are more complex and computationally intense than SEND and SSL. This is why we prefer SSL and SEND cost functions for Robotic Sound Source Separation. The SEND distribution for the sound source $\mathbf{z}(t)$ is given by

$$p_{\mathrm{SEND}}(\mathbf{s}(t)) = c\frac{e^{-\sqrt{(2/F)}||\mathbf{s}||_2}}{||\mathbf{s}||_2^{2F-1}} \quad \forall t, \tag{6.5}$$

where F is the number of discrete frequencies, c is a normalization factor and $||\mathbf{s}||_2$ denotes the L2-norm. The SSL distribution is computed by

$$p_{\mathrm{SSL}}(\mathbf{s}(t)) = c \cdot e^{-2\cdot||\mathbf{s}||_2} \quad \forall t. \tag{6.6}$$

Both distributions have the property that the sources are uncorrelated, i.e., $E\{\mathbf{s}_i\mathbf{s}_i^H\} = \mathbf{I}$. The source priors are then utilized to construct a likelihood-maximizing cost function. We assume that the whitened mixtures x_0^f are separated by a demixing matrix $\mathbf{W}^f$ to yield the estimates $\hat{\mathbf{y}}^f$ for each frequency bin, computed by

$$\hat{\mathbf{y}}^f = \mathbf{W}^f\mathbf{x}_0^f \tag{6.7}$$

The estimates $\hat{\mathbf{y}}^f$ are then combined to estimates $\hat{\mathbf{y}}_i$. Then, the source distributions can be used to calculate the likelihood of the estimates $\hat{\mathbf{y}}_i$. Assuming independence among the samples, the likelihood C_i of one separated source $\hat{\mathbf{y}}_i$ is computed by

$$C_i(\mathbf{W}^1, ..., \mathbf{W}^F) = \prod_{n=1}^{T} p(\hat{\mathbf{y}}_i). \tag{6.8}$$

This is only a function of the demixing matrices, as the estimates $\hat{\mathbf{y}}_i$ are calculated using the observed mixtures $\mathbf{x}$. The overall likelihood is thus the product of the individual likelihoods:

$$C(\mathbf{W}^1, ..., \mathbf{W}^F) = \prod_{i=1}^{N} C_i(...) = \prod_{i=1}^{N}\prod_{n=1}^{T} p(\hat{\mathbf{y}}_i) \tag{6.9}$$

The probability densities are based on exponentials, which makes the mathematical treatment of the log-likelihoods

$$L(\mathbf{W}^1, ..., \mathbf{W}^F) = \ln(C) = \sum_{i=1}^{N}\sum_{n=1}^{T} \ln(p(\hat{\mathbf{y}}_i)) \tag{6.10}$$

advantageous. As we are interested in the separation matrices that maximize the log-likelihood, we can divide L by the number of available samples T, which yields

$$L(\mathbf{W}^1, ..., \mathbf{W}^F) = \sum_{i=1}^{N} \tilde{E}\{\ln(p(\hat{\mathbf{y}}_i))\} \tag{6.11}$$

and does not change the matrices that maximize this expression [10]. The expectation operator in this equation (6.11) is the heuristical expectation, i.e. the sum over all concerned terms divided by their number.

So far, a function that *measures* the “quality" (in terms of likelihood) of the separation matrices is derived, which is only dependent on the observed mixtures and the separation matrices. Now, separation matrices that maximize this function have to be found.

6.3 Cost Function Optimization

Using the log-likelihood cost function, the optimization problem can be described by

$$\underset{\mathbf{W}^1,\ldots,\mathbf{W}^F}{\operatorname{argmax}} L(\mathbf{W}^1,\ldots,\mathbf{W}^F) \qquad \text{s.t. } \mathbf{W}^f\mathbf{W}^{fH} = \mathbf{I} \quad \forall f. \tag{6.12}$$

By solving the optimization problem in Equation (6.12), the demixing matrices $\mathbf{W}^F$ to estimate the source signals can be found. The optimization of the cost function can be solved as follows: By assuming circular symmetry in the source variables, a 2nd order Taylor Expansion of the cost function can be calculated. This allows the implementation of a Newton optimization step, which eventually yields a fixed-point equation under some approximations that allows for an update of the demixing matrices. For further details, refer to [10]. This fixed-point equation will be used to develop an update step that refines the quality of the demixing matrices iteratively. Algorithm 1 summarizes the iterative separation algorithm.

Algorithm 1 Update Step

1: **for** $f = 1 \ldots F$ **do**
2: $\quad \hat{\mathbf{y}}^f \leftarrow \mathbf{W}_0^f \mathbf{x}_0^f$ ▷ Frequency Bin Updates
3: **end for**
4: Combine $\hat{\mathbf{y}}^f$ and yield estimates $\hat{\mathbf{y}}_i$
5: **for** $f = 1 \ldots F$ **do**
6: $\quad \mathbf{W}^f \leftarrow \text{Update}(\mathbf{W}_0^f, \hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_N, \mathbf{x}_1, \ldots, \mathbf{x}_N)$
▷ Separation Matrix Update
7: $\quad \mathbf{W}^f \leftarrow (\mathbf{W}^f\mathbf{W}^{fH})^{-\frac{1}{2}}\mathbf{W}^f$ ▷ Decorrelation
8: **end for**

6.4 Iterative Separation Algorithm

Using the fixed-point equation of section 6.3, an algorithm is developed that includes all necessary steps to improve the quality of the separation matrices iteratively. This algorithm involves three steps:

1. Update each separation matrix $\mathbf{W}^f$.
2. Decorrelate the separation matrices $\mathbf{W}^f$ (make them unitary).
3. Stop algorithm if the cost function does not improve any longer.

The next sections give a detailed description of the three steps of the iterative algorithm.

6.4.1 Separation Matrix Update

The separation matrix update has to be computed for all frequency bins. Exemplarily, we describe the update procedure for one frequency bin. Each row vector $\mathbf{w}_i^f$ of the separation matrix is updated by

$$\mathbf{w}_i^f \leftarrow \tilde{E}\{G'(||\hat{\mathbf{y}}_i||_2^2) + |\hat{y}_i^f|^2 G''(||\hat{\mathbf{y}}_i||_2^2)\}\mathbf{w}_i^f - \tilde{E}\{(\hat{y}_i^f)^* G'(||\hat{\mathbf{y}}_i||_2^2)\mathbf{x}_0^f\}. \quad (6.13)$$

The functions G' and G'' have been introduced in accordance with [10] to simplify the derivation of the update rule and its notation. G' and G'' depend on whether SSL or SEND is used, and are the first and second order derivatives of

$$G_{\mathrm{SSL}}(s) = \sqrt{s} \quad (6.14)$$

and

$$G_{\mathrm{SEND}}(s) = \sqrt{(2/L)s} + (L - 0.5)\ln(s) \quad (6.15)$$

The current cost L can be evaluated using the G functions as

$$L = -\sum_{i=1}^{N} \tilde{E}\{G(||\hat{\mathbf{y}}_i||_2^2)\}. \quad (6.16)$$

L can be used to decide when to stop iterating: If no improvement of the cost function is made any longer, the algorithm can be terminated.

Decorrelation

The new separation matrices that were calculated using Equation (6.13) do not fulfill the unitary constraint, i.e. $\mathbf{W}^f\mathbf{W}^{fH} \neq \mathbf{I}$. There are two ways to overcome this problem: The first is to perform a Gram-Schmidt orthogonalization, the second method is using a symmetric decorrelation method.

The Gram-Schmidt method however is not applicable, as it performs orthogonalization successively. Consequently each separation vector is not treated identically. The symmetric decorrelation yields an unitary demixing matrix that minimizes the distance to the

estimated separation matrix in Equation (6.13) with respect to the Frobenius norm [8]. The unitary demixing matrix $\hat{\mathbf{W}}^f$ is retrieved by

$$\hat{\mathbf{W}}^f \leftarrow (\mathbf{W}^f \mathbf{W}^{fH})^{-\frac{1}{2}} \mathbf{W}^f. \tag{6.17}$$

This decorrelation procedure has to be conducted for each frequency bin.

6.5 Spectral Compensation

In section 6.1, to yield uncorrelated mixtures, we transformed the mixtures in each frequency bin using a whitening matrix. As the “real" scaling cannot be determined, variance one is assigned to each of the mixtures. By nature, speech signals do not have the same power in each frequency. Just consider a human speech, which contains a lot of power in the range of 300Hz to 3000Hz and only very little in higher frequencies. Figure 6.2 illustrates the STFT of female speech in both original form and whitened frequency bins. By whitening, each row is assigned to the same spectral power density and the same

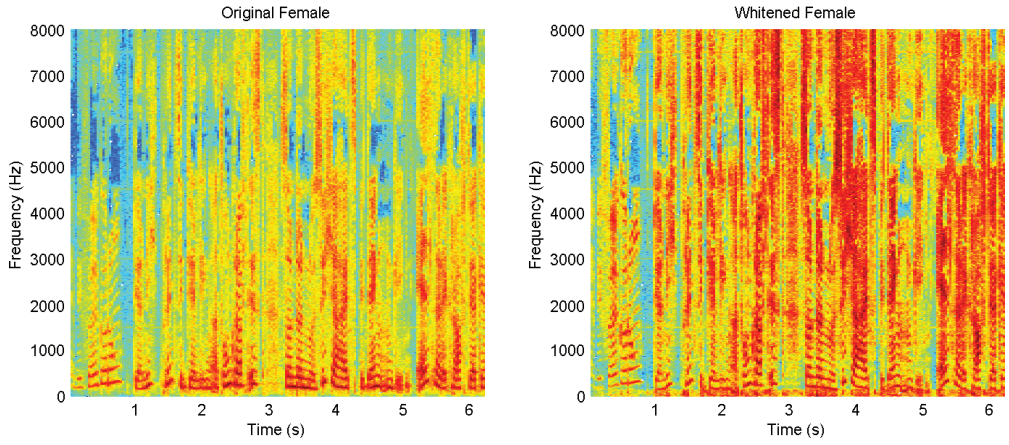

Figure 6.2: Original and Whitened Female Speech

power is assigned to all frequencies. To achieve natural sound of the separated signals, the whitening has to be reversed by a procedure called “Spectral Compensation".

Because we cannot determine the true scaling of the sources, the true scaling of the sources has to be estimated, whereas only the product of the diagonal elements of the true mixing matrices and the sources is calculated [6]. Although the scaling factors of the mixing matrix are involved, this approach results in a good spectral compensation.

In accordance with [6] the spectral compensation is conducted as follows. Although we cannot determine the exact mixing (or demixing) matrix, it is possible to determine the product of its diagonal elements with the true sources, which is used as compensated outcome. Recall the mixing model of one frequency bin of Equation (3.2), where the mixtures $\mathbf{x}^f$ are not whitened yet. After whitening, we yield uncorrelated mixtures

$$\mathbf{x}_0^f = \mathbf{Q}^f \mathbf{H}^f \mathbf{s}^f \tag{6.18}$$

with variance one. The matrix $\mathbf{H}^f$ can however not be determined. Thus, the whitened mixtures are considered as the mixed signal. This way, the whitened mixtures $\mathbf{x}_0^f$ satisfy also

$$\mathbf{x}_0^f = \mathbf{H}_0^f \mathbf{s}_0^f, \tag{6.19}$$

where $\mathbf{H}_0^f$ and $\mathbf{s}_0^f$ denote the whitened mixture matrix and source signals, respectively. Combining equation (6.18) and equation (6.19) yields

$$\mathbf{Q}^f \mathbf{H}^f \mathbf{s}^f = \mathbf{H}_0^f \mathbf{s}_0^f \tag{6.20}$$

$$\mathbf{s}_0^f = (\mathbf{H}_0^f)^{-1} \mathbf{Q}^f \mathbf{H}^f \mathbf{s}^f. \tag{6.21}$$

As the components of $\mathbf{s}_0^f$ and $\mathbf{s}^f$ are independent and IVA prevents permutations, the matrix $\mathbf{D}^f = (\mathbf{H}_0^f)^{-1} \mathbf{Q}^f \mathbf{H}^f$ must be diagonal. This allows the computation of the product of diagonal elements of $\mathbf{D}$ with the sources without knowing $\mathbf{H}^f$:

$$\begin{aligned} \operatorname{diag}(\mathbf{H}^f)\mathbf{s}^f = \operatorname{diag}((\mathbf{Q}^f)^{-1}\mathbf{H}^f\mathbf{D}^f)\mathbf{s}^f = \\ = \operatorname{diag}((\mathbf{Q}^f)^{-1}\mathbf{H}^f)\mathbf{s}_0^f. \end{aligned} \tag{6.22}$$

Spectral compensation can then be performed on the white demixed source estimates $\hat{\mathbf{y}}^f$ using the diagonal matrix $\operatorname{diag}((\mathbf{Q}^f)^{-1}\mathbf{H}^f)$ to yield compensated separations $\mathbf{y}^f$:

$$\mathbf{y}^f = \operatorname{diag}((\mathbf{Q}^f)^{-1}\mathbf{H}^f)\hat{\mathbf{y}}^f, \quad \mathbf{H}^f = \mathbf{W}^{fH}. \tag{6.23}$$

By applying inverse STFT on $\mathbf{y}^f$, the time-domain representation of the separated signals are computed.

7 Performance Evaluation

The BSS EVAL toolbox [18] is a frequently utilized toolbox to evaluate source separation algorithms. The evaluation by the BSS EVAL toolbox is done by decomposing the signal into several signal parts, depending on which signal deformation is allowed. The toolbox supports the following signal deformations:

- Signal Gain factor only
- Arbitrary linear filtering with defined filter length
- Time-Varying Gain
- Time-Varying linear filtering

A gain factor only decomposition is used for instantaneous mixtures, i.e. there are no echoes and no delays present in the mixture.

Arbitrary linear filtering is useful for echoic environments, if the separated signals should be compared to the original sound source.

The time-varying decompositions allow for time varying instantaneous environments (the time-varying gain decomposition) or entirely dynamic situations in case of time-varying linear filtering.

In our audio laboratory tests, we use sound signals that were presented by loudspeakers and recorded by a linear microphone array in a semi-anechoic room and an echoic classroom. Beside the mixture, each sound source is recorded individually, which is regarded as the reference recording.

IVA is not designed to cancel out echoes or other effects, it just yields independent outcomes. Consequently, we consider the best unmixing results to be achieved are close to the individual reference recordings, as these contain characteristic effects of the room, which are described by the so-called Room Impulse Responses (RIRs) beside hardware related distortions.

Therefore, the linear-filtering allowed distortion would allow too many distortions, as the recorded source already inherits the linear room impulse response. On the other hand, the gain factor decomposition does not allow for a slight time shift of the signal due to time differences between the microphones, and is thus also not suitable for our evaluation. The relative angles and distances between the microphones of the array and the sound sources are different due to the arrays geometry, cause time shifts as well as gain differences between the microphone recordings which have to be considered for the evaluation.

To take the effects of real test environment into account, the BSS toolbox is enhanced by the gain-shift decomposition function, which allows a gain factor *and* a time-shift of the original signal.

In accordance with the BSS EVAL toolbox, the the Signal to Distortion Ratio (SDR), Signal to Interference Ratio (SIR) and the Signal to Artifacts Ratio (SAR) are computed to evaluate the separation performance.

7.1 Signal Decomposition

Each decomposition function calculates a target signal, an interference signal and an artifacts signal using a certain ground truth, in this case the individual reference recordings, and the source estimates. The target signal incorporates the desired signal part with a certain amount of allowed distortions like gain, filter and time shift. The interference and artifacts signals incorporate the non-wanted signal parts. All three parts are then used to calculate the SDR, SIR and SAR ratios.

For the gain-shift decomposition, we used the gain decomposition function of the BSS toolbox and enhanced it to also incorporate time shifts of the source signal. Mathematically, this can be described by a gain factor and a time shift of a signal filtering the signal with a scaled Dirac impulse at the desired shift position,

$$s_{\text{gain\_shift}}(t) = c\delta(t - n) * s(t), \tag{7.1}$$

where c corresponds to the gain and n is the number of samples by which the original signal is shifted. In practice, we limit the search range for the number of shift samples to correspond roughly to the microphone distance, which is for our experiment setup 30 to 50 samples.

The time shifts were estimated using the Generalized Cross Correlation with Phase Transform (GCC-PHAT) algorithm [9]. We call this decomposition method the gain-shift decomposition.

7.2 Performance Criteria

The SDR value is given by

$$\text{SDR} := 10 \log_{10} \frac{||s_{\text{target}}||_2^2}{||e_{\text{interf}} + e_{\text{artif}}||_2^2} \text{dB} \tag{7.2}$$

and incorporates arbitrary distortions, i.e. interference from other sources and artificial noise that was introduced by the separation algorithm.

The SIR and SAR are computed by

$$\text{SIR} := 10 \log_{10} \frac{||s_{\text{target}}||_2^2}{||e_{\text{interf}}||_2^2} \text{dB} \tag{7.3}$$

and

$$\text{SAR} := 10 \log_{10} \frac{||s_{\text{target}} + e_{\text{interf}}||_2^2}{||e_{\text{artif}}||_2^2} \text{dB}. \tag{7.4}$$

They represent two distortions separately, where the SAR value represents the ratio of all signal parts versus artificial noise and the SIR value the signal part versus interference. For further details, we refer to [18].

7.3 Gain shift vs. Gain only and linear filtering

In Table 7.1 the SIR, SDR and SAR values are compared, which were generated by different evaluation methods. The mixtures are recorded in an anechoic room. The optimal delays estimated by the GCC-Phat for our gain-shift allowed decomposition are denoted in brackets. The values of gain only and gain shift are nearly the same, which is due to a

	Gain Only		
Values in [dB]	SDR	SIR	SAR
Source 1	2.8744	26.5070	2.9030
Source 2	7.2820	21.8537	7.4645
Source 3	3.4742	9.0139	5.4099
	Gain Shift		
Values in [dB]	SDR	SIR	SAR
Source 1 [-2]	4.6773	29.3189	4.6973
Source 2 [0]	7.2820	21.8537	7.4645
Source 3 [0]	3.4742	9.0139	5.4099
	Linear Filter 32 tap		
Values in [dB]	SDR	SIR	SAR
Source 1	4.4927	21.3322	4.6154
Source 2	7.7884	18.0198	8.2890
Source 3	3.6561	7.9749	6.3046

Table 7.1: Comparison of gain only, gain shift and linear filter allowed decomposition functions

time shift of maximum of two samples shift estimated by GCC-Phat. The time shift would increase for higher distances between the microphones of the arrays. As for the linear-filtering allowed decomposition, the values are slightly better than the values of gain only and gain shift.

Beside the more appropriate measurement of the gain shift decomposition for real-world audio recordings, another advantage of the gain shift decomposition is the lower complexity compared to the linear filter decomposition, as no linear filter has to be estimated within the decomposition.

8 Experimental Results

In this section, we apply IVA on synthetic mixtures and audio mixtures, recorded in our laboratory environment and a classroom environment. Performance of IVA is investigated and discussed for different parameters of the IVA.

8.1 Experimental Setup

In this section, the different environments are described in detail. The original, unmixed sound sources are male and female speech signals with 48 kHz sampling rate and 16 bit quantization. Three different speech signals are played back by three loudspeakers that are placed within a semi-anechoic chamber. The mixture is recorded by a linear array that consists of three microphones with 5.5 cm spacing. Finally, the sound separation experiment is conducted in the anechoic and echoic environment. For evaluation, each source is recorded separately, before the mixture is captured by the microphones. Table 8.1 gives an overview of the environment and equipment we used for our experiments.

Throughout the experiment section, we use the following test data:

- Sound sources: speech of two Males and one Female
- Length: 10 s
- Mixtures recorded with 48kHz, downsampled to 16kHz for further processing
- Geometry: Loudspeakers aligned equidistant from the microphones at 40, 100 and 150 degree

8.2 IVA Cost functions and Convergence Speed

To enable a mobile robotic platform real-time sound processing, which is important for quick responses of the robot to sound inputs, the sound source separation algorithm has to be as fast as possible. Convergence speed of the separation algorithms therefore is an issue. In this experiment, three sound sources were separated using SEND and SSL cost function. In dependency of the numbers of iterations, the separation results are observed. The results for one separated sound source is illustrated in Figure 8.1. We can see that SSL converges very quickly and reliably, whereas SEND is slower and less reliable in terms

Source data	48kHz; 32bit male and female speech WAV-files of 10s length
Microphones (Mic)	3 x AVM MI 17/1928
Mic spacing	0.05m, uniform linear array
Amplifier	MFA IV81 IEPE
Preamp	GRAS-Type 26AC
Sound sources	KS digital Coax C5-Tiny
DSP	Hammerfall DSP Multiface II
Anechoic room (AR)	4.7 m x 3.7 m x 2.84 m
AR noise level	$<$30 dBA
AR reverb. time	t_{60} =0.08 s

Table 8.1: Information about the equipment used in our experiment.

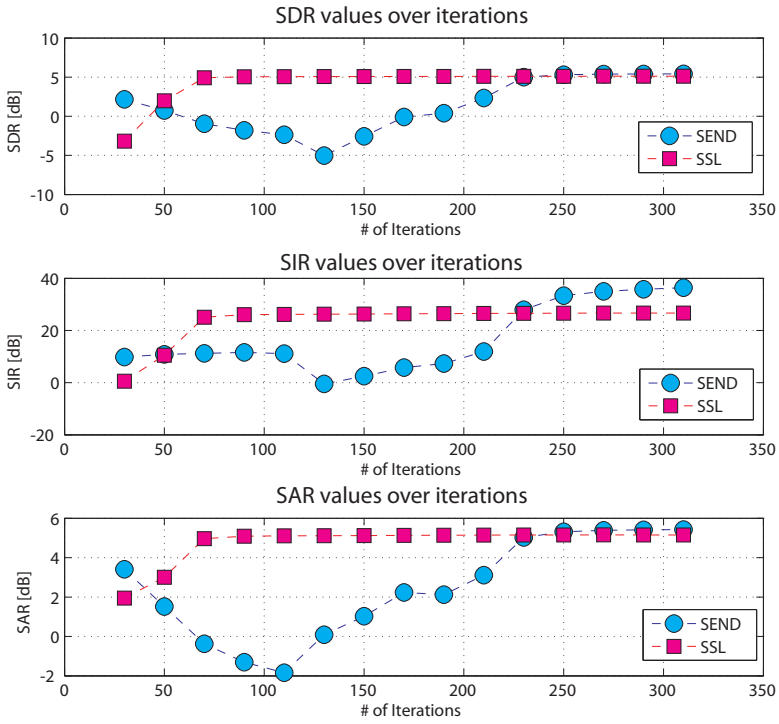

Figure 8.1: Convergence properties of SSL and SEND

of convergence. Concerning SDR, SIR and SAR values, SEND however enables slightly better separation results.

Due to faster computation while reaching nearly the same separation performance, SSL is more suitable for our scenario and thus we utilize the SSL costfunction in the following experiments.

8.3 STFT Overlap and Window Type

To investigate the effect of the window function and the overlap within the STFT, sound separation experiments are conducted with half-overlapping and $\frac{3}{4}$ overplapping sine and hann windows. As shown in Table 8.2, the results are nearly similar for the anechoic room

	Sine Window, half overlapping		
Values in [dB]	SDR	SIR	SAR
Source 1	4.6773	29.3189	4.6973
Source 2	7.2820	21.8537	7.4645
Source 3	3.4742	9.0139	5.4099
	Hann Window, $\frac{3}{4}$ overlapping		
Values in [dB]	SDR	SIR	SAR
Source 1	5.4241	30.0074	5.4435
Source 2	7.4095	21.5994	7.6082
Source 3	4.1513	9.1765	6.2862

Table 8.2: STFT Impact on Separation

environment, while the sine windows benefit lower computational complexity qualifying the sine windows for our robotic scenario.

8.4 Separation Results

In this section, we apply IVA with SSL cost function and half-overlapping sine windows on a sound separation problem. The mixtures are recordings in an anechoic environment and an echoic classroom environment.

The results can be seen in tables 8.3 and 8.4. We used podcast recordings of 10 s length each from a public radio station.

Although we have negative values for the 3-source echoic case, listening tests revealed that separation still took place for all sources which yielded positive SIR values. For negative SIRs, the outcomes mainly consisted of noise.

	Anechoic Room		
Values in [dB]	SDR	SIR	SAR
Source 1	11.5096	40.3683	11.5156
Source 2	8.4271	25.8522	8.5176
	Echoic Room		
Values in [dB]	SDR	SIR	SAR
Source 1	3.8114	14.0176	4.4156
Source 2	-2.0667	12.2217	-1.6490

Table 8.3: Separation Results for two sources

	Anechoic Room		
Values in [dB]	SDR	SIR	SAR
Source 1	4.6235	28.9825	4.6450
Source 2	7.2817	21.9638	7.4595
Source 3	3.4172	8.9415	5.3670
	Echoic Room		
Values in [dB]	SDR	SIR	SAR
Source 1	-2.1555	4.2178	0.3769
Source 2	-2.7026	3.3399	0.1933
Source 3	-7.1980	6.6388	-6.1624

Table 8.4: Separation Results for three sources

9 Conclusion and Future Work

In this paper, IVA is applied on sound mixtures of anechoic and echoic recordings. In contrast to frequency-domain ICA that treats all frequency bins separately, IVA assumes dependence within the scalar frequency components which successfully aligns the sources and prevents permutations with low computational complexity. Therefore, IVA is applicable in teleconference and robotic hearing applications. Evaluation was carried out with non-synthetic test data that was recorded in an anechoic room and an echoic seminar room. As we consider mono-recorded sources as ground truth for the evaluation with the BSS-Evaluation Toolbox, we propose a new decomposition function that allows for gain-shift allowed distortions.

Acknowledgment

This work was fully supported by the German Research Foundation (DFG) within the collaborative research center SFB-453 ”High Fidelity Telepresence and Teleaction“.

Bibliography

[1] B. Arons. A review of the cocktail party effect. *Journal of the American Voice I/O Society*, 12(7):35–50, 1992.

[2] J.F. Cardoso. Multidimensional independent component analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 1941–1944. IEEE, 1998.

[3] J.F. Cardoso. High-order contrasts for independent component analysis. *Neural computation*, 11(1):157–192, 1999.

[4] M.A. Casey and A. Westner. Separation of mixed audio sources by independent subspace analysis. *Proceedings of the International Computer Music Conference*, pages 154–161, 2000.

[5] P. Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[6] J. Hao, I. Lee, T.W. Lee, and T.J. Sejnowski. Independent vector analysis for source separation using a mixture of gaussians prior. *Neural computation*, 22(6):1646–1673, 2010.

[7] F.J. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1):51–83, 1978.

[8] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*. Adaptive and learning systems for signal processing, communications, and control. J. Wiley, 2001.

[9] C. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech and Signal Processing,*, 24(4):320 – 327, 1976.

[10] I. Lee, T. Kim, and T.W. Lee. Fast fixed-point independent vector analysis algorithms for convolutive blind source separation. *Signal Processing*, 87(8):1859–1871, 2007.

[11] I. Lee and T.W. Lee. On the assumption of spherical symmetry and sparseness for the frequency-domain speech model. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1521–1528, 2007.

[12] R. Lyon. A computational model of binaural localization and separation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pages 1148–1151, 1983.

[13] S. Makino, T.W. Lee, and H. Sawada. *Blind speech separation*. Signals and Communication Technology. Springer, 2007.

[14] T. Melia. *Underdetermined Blind Source Separation in Echoic Environments Using Linear Arrays and Sparse Representations*. PhD thesis, University College Dublin, 2007.

[15] R. Roy and T. Kailath. ESPRIT-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech and Signal Processing,*, 37(7):984–995, 1989.

[16] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation,*, 34(3):276–280, 1986.

[17] B.D. Van Veen and K.M. Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE Magazine on Acoustics, Speech, and Signal Processing*, 5(2):4–24, 1988.

[18] E. Vincent, R. Gribonval, and C. Févotte. Performance measurement in blind audio source separation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(4):1462–1469, 2006.