Extension of BSS Algorithms to Different Array Geometries

Session: Spring 2017

Submitted by:

Muhammad Umair Khan 2017-MS-CE-15

Supervisor:

Dr. Tania Habib

Department of Computer Science and Engineering

University of Engineering and Technology

Lahore Pakistan

Extension of BSS Algorithms to Different Array Geometries

Submitted to the faculty of the Computer Science and Engineering Department of the University of Engineering and Technology Lahore in partial fulfillment of the requirements for the Degree of

Master of Science

in

Computer Engineering.

Internal Examiner	**External Examiner**
Signature:	Signature:
Name:	Name:
Designation:	Designation:
Chairperson	**Dean**
Signature:	Signature:
Prof. Dr. Shazia Arshad	Name:

Department of Computer Science and Engineering

University of Engineering and Technology

Lahore Pakistan

Declaration

I declare that the work contained in this thesis is my own, except where explicitly stated otherwise. In addition this work has not been submitted to obtain another degree or professional qualification.

Signed: _____

Date: _____

Acknowledgments

This document is based on the research work conducted in the field of speaker localization using condensed and static microphone array. I am thankful to a number of elders, friends and colleagues who helped me through their sincere guidance throughout this research work.

First of all, I am thankful to my thesis supervisor Dr. Tania Habib for the unique research topic she suggested. I also thank her for the continued encouragement, understanding, guidance and support. Without her encouragement, I could never have achieved the desired results. I must say, Thank You Madam!

I would also thank my uncle-in-law, Shahid Ikram. He was the main driving force behind me seeking admission in the M.Sc. program. He kept on persuading and, finally, convinced me. With his motivational talks, I was able to manage the studies with the office workload.

I also acknowledge the support from my under-graduate juniors who lend their recording hardware to me whenever I needed it and also made themselves available for the repeated recording sessions. Without their support, this research work was logistically impossible.

I am thankful to my company Mentor Graphics who allowed me to take admission in the M.Sc. program alongside the office.

Finally, I am thankful to my father and my younger brother. They contributed more than their due share to manage the home stuff and kept me going for this program. Without their support, this would not have concluded successfully. Thank you all of you.

Dedicated to my family — my father, (late) mother and younger brother.

Contents

List of Figures

List of Tables

Abbreviations

BSS	**B**lind **S**ource **S**eparation
TDoA	**T**ime **D**ifference **o**f **A**rrival
DoA	**D**irection **o**f **A**rrival
ASR	**A**utomatic **S**peech **R**ecognition
SVD	**S**ingular **V**ector **D**ecomposition
MLE	**M**aximum **L**ikelihood **E**stimator
SNR	**S**ignal to **N**oise **R**atio
RMSE	**R**oot **M**ean **S**quare **E**rror
SRP-PHAT	**S**teered-**R**esponse **P**ower **PHA**se **T**ransform

Abstract

Human beings have long desired to create intelligent computer systems that can make decisions, process raw data and produce the desired results with minimal human intervention. Offloading raw data processing to computer systems can save time as well as produce accurate and reliable results. The field of audio processing is no exception. Blind source separation is a statistical technique that can estimate individual source signals without any apriori information about their characteristics. The idea is important because, in real-world audio recordings, the acoustic characteristics of the speaker's voice is unknown. The problem of variable speaker localization – a case when the number of simultaneously active speakers differs from the number of available sensor nodes – is particularly important as the installed recording setups are usually fixed while the speakers can vary. The algorithm must be robust enough to adapt to any number of simultaneously active acoustic sources.

In this thesis, a novel method for localization of concurrent speakers using blind source separation by exploiting microphone array geometry is presented. The method uses TRINICON BSS [23] algorithm as the baseline for determining the Direction of Arrival of speech signals. The algorithm is evaluated in real-world scenarios such as low-noise labs, meeting rooms and reverberant environments. We used Root Mean Square Error for performance evaluation of the algorithm which stayed below 10 for most of the cases. The algorithm is also extended to track dynamic speakers. Detailed results are presented in the last chapter.

Chapter 1

Introduction

1.1 Overview

Multichannel speech processing using microphone arrays has been a hot topic of research lately. The reason being the additional information that we get by exploiting the spatial diversity of the acoustic signal. This additional information can be used to not only improve sound quality, like, noise reduction but also improve the performance of various speech processing algorithms like Automatic Speech Recognition (ASR), etc. In addition, microphone arrays can be used for speaker localization given that the separation between microphones is large enough as compared to the speed of sound in air. A method of determining DoA using a spherical microphone array is presented in [4]. This DoA information can be used in speaker localization [5], speaker tracking [6] and camera movement [7]. In addition to this, DoA can also be used to improve the performance of the ASR's [8].

This research work uses DoA with a very popular speech processing algorithm called Blind Source Separation (BSS). The most charming feature of BSS is that it is purely a statistical method that does not assume any prior information about the characteristics of the source signals. This increases the adaptability of the algorithm to speakers of different vocal characteristics. We use redundancy in Direction of Arrival (DoA) to solve the problem of Under-determined BSS.

1.2 Motivation

In real-world scenarios, the human hearing has the fascinating capability of localizing acoustic sources even in reverberant and noisy environments [9] which

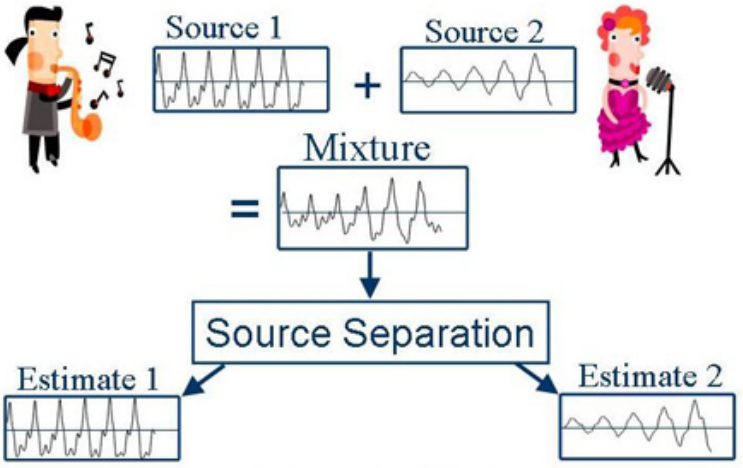


FIGURE 1.1: Blind source separation from audio mixture. [1]

also helps humans in concentrating on speech/sound from a particular speaker/-source. But in recorded audio, such estimation is very difficult to make. With the advancement in artificial intelligence, computers are now equipped with the capability to handle different situations smartly and most of the record-keeping is done in the form of recordings. Inspired by the human auditory system, different machine-listening techniques have been developed like Computational Auditory Scene Analysis (CASA). CASA is different from the BSS as it is based on the human auditory system, and thus, can use no more than a pair of microphones. The number of usable microphones limits the source location estimation accuracy of the system. Blind source separation algorithms, on the other end, do not have any limitation on the number of usable microphones. Researchers believe that the performance of ASR's can be improved by considering the location and distance of target speech source[8]. In [10], the author has proved that the performance of the speaker recognition system can be improved by considering the speaker's location information. BSS algorithms intrinsically use Time Difference of Arrival (TDOA) information for source separation, which can be exploited for speaker source localization. Such a system will be very helpful in speaker tracking, hands-free communication, teleconference, and virtual classrooms.

1.3 Problem Statement

To design and develop an algorithm to solve the problem of under-determined BSS by exploiting the geometry of the microphone array as existing BSS algorithms are geometrically restrictive.

1.4 Research Objectives

As stated in the problem statement, we are aiming to extend the BSS algorithms for different array geometries. The proposed algorithm will help achieving the goal of hands-free communication where the speaker would not need to have a microphone tied to his clothing all the time, rather he'll be free to move around and the sensor network will have the intelligence to track the speaker. Following are the main objectives of the proposed work:

1. Corpus collection.
2. Analysis of state-of-the-art BSS technologies.
3. Algorithm improvement from 1D to 2D microphone array geometry.
4. Proposing a solution for Under-determined BSS.
5. Measurement of performance improvement of the algorithm.
6. Obtaining some publishable results from this work.

1.5 Thesis Organization

Chapter 1 introduces the reader with this research work conducted under this thesis. Blind Source Separation, the types of mixing systems and the heuristics of the process with the underlying hardcore mathematics are discussed in Chapter 2. Chapter 3 presents a comprehensive literature survey of the recent research work conducted in the area of speaker localization using microphone arrays. Chapter 4 is focused on how the system design was designed to carry out experiments. It explains the recording set up and the details of the corpus collection. Chapter 5 proposes the methodology for speaker localization using condensed and static microphone array by extensively explaining every aspect of it with the help of graphical illustrations and algorithms. Chapter 6 summarizes the results of the proposed algorithm on the collected corpus. The stability of the output estimates generated by the proposed algorithm is compared against those of the Steered Response Power Phase Transform (SRP-PHAT) algorithm. The performance of the proposed algorithm is measured based on the Root Mean Square Error (RMSE). It explains how the algorithm has been extended for dynamic speaker tracking. It also sheds light on the future prospects of the proposed algorithm.

1.6 Publication

As a contribution to the ongoing research, a manuscript of the work conducted under this thesis, titled *Concurrent Speakers Localization using Blind Source Separation and Microphone Array Geometry*, has been submitted for publication to *Springer Circuits, Systems, and Signal Processing Journal*.

Chapter 2

Blind Source Separation

2.1 Motivation

Today is the era of technological development and we are surrounded by all kinds of sounds. The world has progressed so far that the distances have shrunk. Today, we have become used to conversing with people across the world. Such communications start with a single or multiple microphones. The noise makes communication difficult and harder to concentrate on the target speech. If there are other people also speaking concurrently, their audio also acts as noise and further degrade the sound quality. This raises the requirement of a sound extraction mechanism in human-human and human-machine interactions.

2.2 Overview

Blind Source Separation (BSS) is a technique that is used to estimate and recover source signals from a mixture only with the information received at each sensor node. There is no need for any apriori information about the source signals, such as the mixing system and the frequency spectrum in the estimation process. The use of BSS is widely accepted by the research community [11].

BSS has a wide range of applications that are not limited to speech processing. When any communication signals pass through a convolutive transmission medium, their spectral and temporal characteristics get altered, sometimes, to a level, that the received signal is completely different from the signal that was originally transmitted. In speech processing, BSS also takes inspiration from the human auditory system that humans can focus their attention on a particular sound source even in noisy and reverberant environments. Blind source separation, when applied to speech processing is called *Blind Speech Separation*. Onwards

in this chapter, the acronym BSS would refer to Blind Speech Separation.

Figure 2.1 shows a complete block diagram of the blind source separation process. Three acoustic sources concurrently generate acoustic signals designated as $s_1(t)$, $s_2(t)$ and $s_3(t)$. The sound signals propagate through the propagation channel and arrive at the three microphones. It is worth noting that each microphone receives all the three acoustic signals in different proportions depending on the location of the acoustic sources and the placement of the microphones. A microphone closer to an acoustic source will receive a high Signal to Noise Ratio (SNR) signal than a distant microphone. The microphones generate sensor signals that are labelled as $x_1(t)$, $x_2(t)$ and $x_3(t)$ respectively. The task of a BSS un-mixing system is to receive sensor signals as input and estimate the source signals. $\hat{s}_1(t)$, $\hat{s}_2(t)$ and $\hat{s}_3(t)$ represent the source signals separated by the BSS un-mixing system.

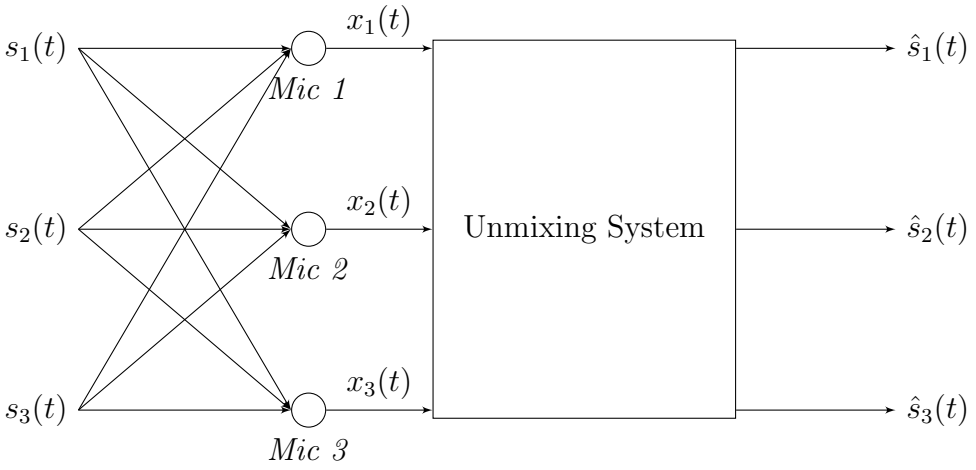


FIGURE 2.1: A block diagram describing 3×3 Blind Source Separation. $s_1(t)$, $s_2(t)$ and $s_3(t)$ are the acoustic signals generated from sources 1, 2 and 3 respectively. $x_1(t)$, $x_2(t)$ and $x_3(t)$ are the sensor signals received by the microphones. $\hat{s}_1(t)$, $\hat{s}_2(t)$ and $\hat{s}_3(t)$ are the estimated source signals.

2.3 BSS and the mixing models

In a multi-speaker environment, the sound that arrives at a sensor node is a mixture of all the sound sources with varying SNRs. The characteristics of the propagation channel define the type of mixture. To solve the BSS problem, we create a mixing model to reverse the mixing procedure by estimating the channel characteristics. Primarily, there are three types of mixing models that are considered in BSS:

1. Instantaneous mixture.

2. Instantaneous and delayed mixture.

3. Convolutive mixture.

Figure 2.2 presents a simplified block diagram of the BSS mixing system. Additive noise is not shown in the figure.

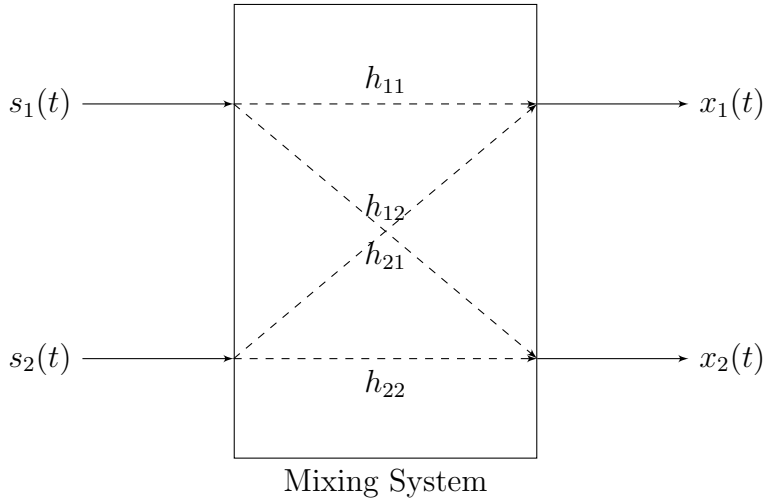


FIGURE 2.2: A 2×2 BSS mixing system. $s_1(t)$ and $s_2(t)$ are the original source signals while $x_1(t)$ and $x_2(t)$ are the signals received at the two sensor nodes. h_{11} and h_{22} are direct path impulse responses while h_{12} and h_{21} are cross path impulse responses.

Instantaneous mixing

It is the type of mixing in which n source signals mix instantaneously to produce m measured signals at sensor nodes. The signal received at the sensor nodes just contains instantaneous components of the source signals mixed. This is the simplest type of mixing in which the signals do not have any time dependence on each other.

Mathematically, instantaneous mixture is modelled as:

$$x_j(t) = \sum_{i=1}^{m} s_i(t)h_{ij} + p_j(t) \quad 1 \leq i \leq N \quad 1 \leq j \leq M \tag{2.1}$$

where,

$$\begin{aligned} x_i(t) &= \text{sensor signal received at } i^{th} \text{ microphone} \\ s_j(t) &= \text{time varying } j^{th} \text{ source signal} \\ h_{ij} &= \text{channel mixing co-efficient from source } i \text{ to sensor } j \\ p_j(t) &= \text{additive noise in the } j^{th} \text{ observed signal} \\ N &= \text{total number of sound sources} \\ M &= \text{total number of microphones} \end{aligned}$$

Instantaneous and Delayed Mixture

In this type of mixture, instantaneous and delayed components of the source signals mix to create the observed sequence. The instantaneous part is the same as described in the previous section. The delayed components are created due to the multipath propagation of the source signals. This kind of mixing is modelled as in (2.2).

$$x_j(t) = \sum_{i=1}^{m} s_i(t - t_{ij})h_{ij} + p_j(t) \quad 1 \leq i \leq P \quad 1 \leq j \leq M \tag{2.2}$$

Convolutive Mixture

In this type of mixture, the source signals convolve together to create the observed signals. The convolution is due to the heavy impact of the propagation channel on the source signals. The convolution also takes care of the multipath effect. This kind of mixing is modeled as in (2.3).

$$x_j(t) = \sum_{l=-\infty}^{+\infty} \sum_{i=1}^{m} s_i(t-1)h_{ijl} + p_j(t) \quad 1 \leq i \leq P \quad 1 \leq j \leq M \tag{2.3}$$

2.4 Types of Blind speech Separation

In a blind speech separation system, the speech signals are recorded by the microphones that are then fed to the un-mixing system. With respect to the number of simultaneously active speech sources and the number of microphones available, a BSS problem can be classified into three categories:

1. Under-determined BSS

2. Determined BSS

3. Over-determined BSS

Under-determined BSS

If the number of available microphones M is less than the number of simultaneously active speakers N in a recording environment, the problem is called *Under-determined Blind Speech Separation*. It is a difficult problem to solve in the area of speech processing and is an active area of research these days because the mixing matrix is not invertible. We shall explore the mixing matrix in the coming sections.

Determined BSS

If the number of available microphones M is equal to the number of simultaneously active speakers N, the problem is called *Determined Blind Speech Separation*. This is the simplest case of the complex BSS problem. The system shown in Figure 2.1 is a 3×3 determined BSS system.

Over-determined BSS

If the number of available microphones M is greater than the number of simultaneously active speakers N, the problem is classified as *Over-determined Blind Speech Separation*. Like under-determined BSS, it is also a difficult problem to solve and suffers from the same matrix non-invertibility problem.

2.5 Problem Formulation

Blind source separation is a complex statistical problem as it does not have any prior information about the source characteristics and the mixing system. Independent Component Analysis (ICA) is a widely used method for solving the BSS problem. ICA is a statistical model for extracting sources from a mixture that are statistically independent. The key to ICA is spatial diversity. ICA exploits this spatial diversity to separate desired components from undesired components by forming a spatial null towards them [11].

To formulate the problem, lets consider an observation vector $\boldsymbol{x}$ received at the sensor nodes. The vector consists of the source signals $\boldsymbol{s}$ mixed together. The mixing system in modelled as a matrix $\boldsymbol{H}$ of direct path and cross path impulse responses of the propagation channel. We define $\boldsymbol{x}$ and $\boldsymbol{s}$ as follows:

$$\boldsymbol{x} = \{x_1, x_2, x_3, ...\}^T \tag{2.4}$$

$$\boldsymbol{s} = \{s_1, s_2, s_3, ...\}^T \tag{2.5}$$

The observed source signals can be modelled as:

$$\boldsymbol{x} = \boldsymbol{H}\boldsymbol{s} \tag{2.6}$$

for a determined 2-input 2-output system, the mixing matrix is defined as follows:

$$\boldsymbol{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \tag{2.7}$$

where, h_{11} and h_{22} are direct path impulse responses between source 1 and sensor 1 and source 2 and sensor 2 respectively. Similarly, h_{21} is the cross path mixing coefficient between source 2 and sensor 1 and h_{12} is the cross path mixing coefficient between source 1 and sensor 2.

Figure 2.3 shows a typical joint probability distribution of the observed signals at the microphones. The x-axis represents the $x_1(t)$ component and the y-axis represents the $x_2(t)$ component.

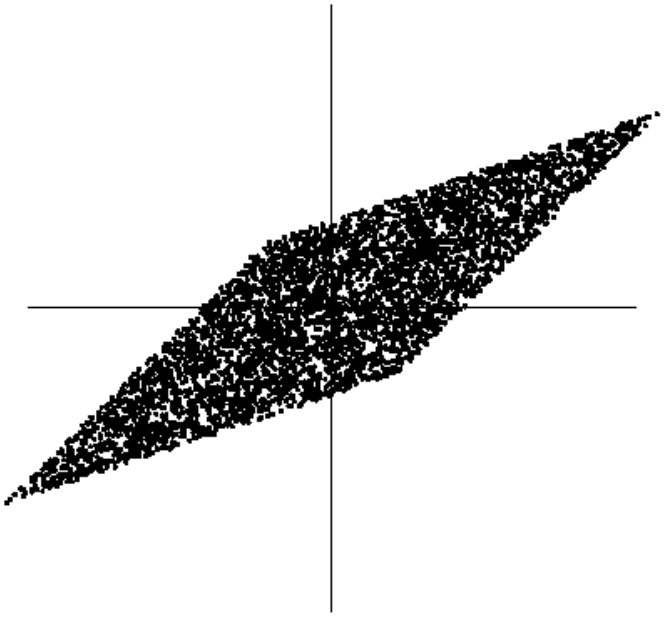


FIGURE 2.3: Typical joint PDF of the observed sequence. [2]

Let $\hat{\boldsymbol{s}}$ be the estimated source signals and $\boldsymbol{W}$ be the unmixing matrix, we can write the BSS problem as:

$$\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x} \tag{2.8}$$

where,

$$\boldsymbol{W} = \boldsymbol{H}^{-1} \tag{2.9}$$

2.5.1 Challenges involved in BSS

As BSS assume very little apriori knowledge about the source signals and the unmixing system, this creates several challenges in solving the BSS problem and also leads to creation of different ambiguities regarding the possible solution. Considering a non-singular matrix $\boldsymbol{M}$. The problem defined by the (2.6) can be redefined as:

$$\boldsymbol{x} = \boldsymbol{H}\boldsymbol{M}(\boldsymbol{M}^{-1}\boldsymbol{s}) \tag{2.10}$$

The matrix $\boldsymbol{M}$ can further be decomposed into a diagonal matrix $\boldsymbol{D}$ and a permutation matrix $\boldsymbol{P}$ as:

$$\boldsymbol{M} = \boldsymbol{D}\boldsymbol{P} \tag{2.11}$$

The structure of $\boldsymbol{M}$ determines the feasibility constraints of the BSS as specified by waveform permutation relations [12]. The form of $\boldsymbol{M}$ shown in (2.11) shows permutation and scaling ambiguities. To solve the permutation ambiguity, BSS algorithms also estimate the Direction of Arrival (DoA) of the speech signal. It helps in aligning separated components in block processing.

ICA is the most popular algorithm used to solve the BSS problem. ICA is applicable to a problem given some initial constraints are met:

1. $\boldsymbol{H}$ is invertible.
2. $\boldsymbol{s}$ is statistically independent.
3. $\boldsymbol{s}$ is non-gaussian.

The speech signals are inherently independent. The speech of two concurrent speakers logically has no dependence on each other. So, the second condition is

naturally met in speech. Figure 2.4 shows the PDF of two typical statistically independent sources.

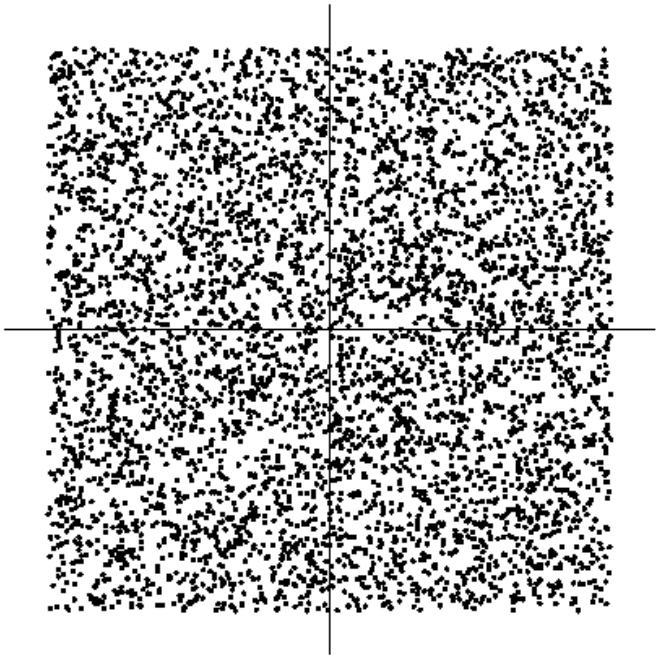


FIGURE 2.4: Typical joint PDF of the whitened observed sequence. [2]

2.5.2 Non-Gaussianity

One of the fundamental assumptions of the ICA algorithm is that the source signals are non-gaussian. To understand this restriction and why the gaussian signals can not be estimated by the ICA algorithm, let us consider that the mixing matrix is orthogonal and the source signals are gaussian. The observed signals will also be gaussian, uncorrelated, and of unit variance. Their joint PDF is given by:

$$p(x_1, x_2) = \frac{1}{2\pi} \exp\left(-\frac{x_1^2 + x_2^2}{2}\right) \tag{2.12}$$

The joint PDF is plotted in Figure 2.5. It can be seen that the distribution is completely symmetric. There is no information available that can be used to uncorrelate the source components. In other words, the mixing matrix is unidentifiable. This is why gaussian source components can not be recovered by the ICA.

2.6 Mathematical Summary of ICA

Having studied the previous section, one can formulate the recovered source signals as:

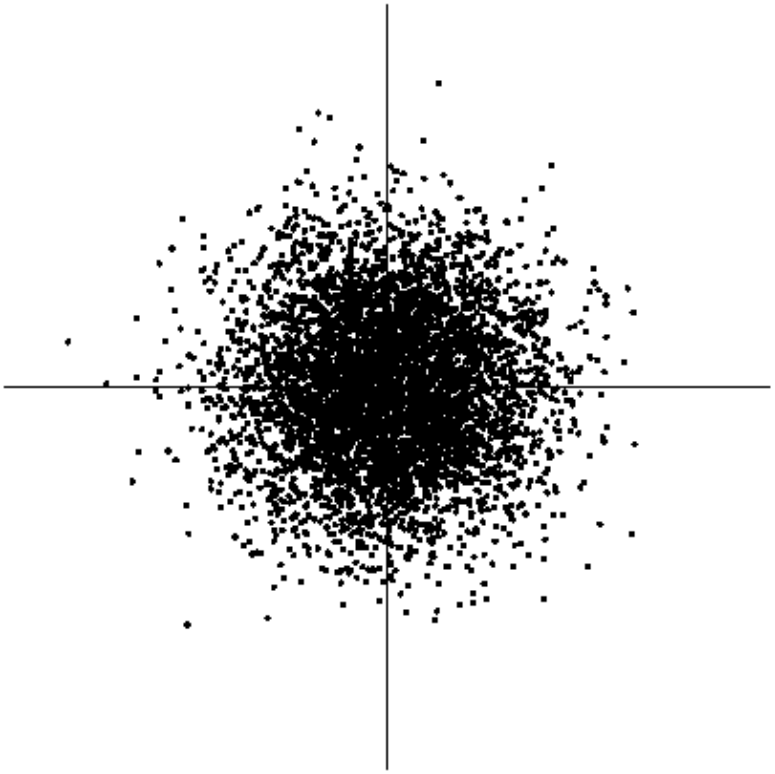


FIGURE 2.5: Joint PDF of two gaussian random variables. [2]

$$\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x} \tag{2.13}$$

Using Singular Vector Decomposition (SVD), we have:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{2.14}$$

this implies:

$$\boldsymbol{W} = \boldsymbol{A}^{-1} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T \tag{2.15}$$

Covariance of the observed signal $\boldsymbol{x}$ is given by,

$$\langle \boldsymbol{x}\boldsymbol{x}^T \rangle = \langle (\boldsymbol{A}\boldsymbol{s})(\boldsymbol{A}\boldsymbol{s})^T \rangle \tag{2.16}$$

Using (2.14) in (2.16), we get

$$\begin{aligned} \langle \boldsymbol{x}\boldsymbol{x}^T \rangle &= \langle (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{s})(\boldsymbol{s}^T(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T)^T \rangle & (2.17) \\ &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\langle \boldsymbol{s}\boldsymbol{s}^T \rangle \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T & (2.18) \end{aligned}$$

Assuming that the source signals are statistically independent, we can write that their covariance is unity.

$$\langle \boldsymbol{s}\boldsymbol{s}^T \rangle = 1 \tag{2.19}$$

This implies that,

$$\langle \boldsymbol{x}\boldsymbol{x}^T \rangle = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T \tag{2.20}$$

Since the matrix $\boldsymbol{V}$ is a unitary matrix, we can write 2.20 as:

$$\langle \boldsymbol{x}\boldsymbol{x}^T \rangle = \boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^T \tag{2.21}$$

2.6.1 Whitening

The next step in ICA is *whitening*. Whitening refers to the process of uncorrelating the observed signals. It means the before applying the ICA algorithm, we apply a linear transform to the observed vector $\boldsymbol{x}$ to get another vector $\boldsymbol{x_w}$ which is white, i.e., its components are now uncorrelated and have unit variance. Mathematically, it means that $E[\boldsymbol{x}\boldsymbol{x_w}] = \boldsymbol{I}$.

Since we know that

$$\boldsymbol{W} = \boldsymbol{A}^{-1} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T \tag{2.22}$$

so,

$$\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x} \tag{2.23}$$

We define $\boldsymbol{x}_w$ as,

$$\begin{aligned} \boldsymbol{x}_w &= (\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T)\boldsymbol{x} & (2.24) \\ \hat{\boldsymbol{s}} &= \boldsymbol{V}\boldsymbol{x}_w & (2.25) \end{aligned}$$

Figure 2.6 shows the whitened version of the PDF shown in Figure 2.3.

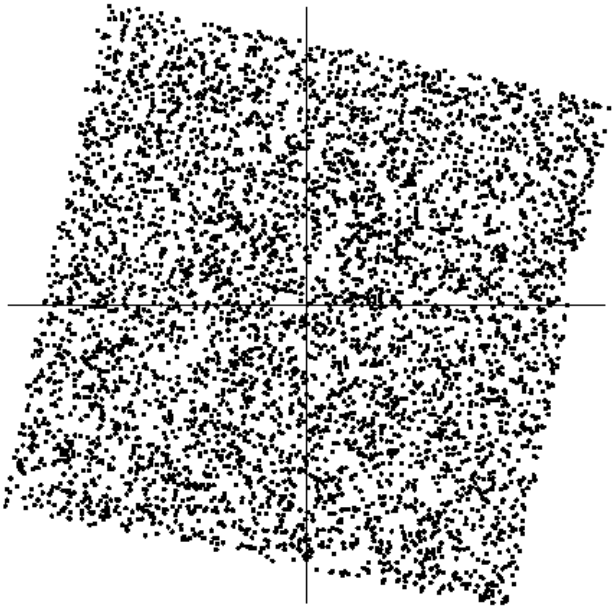


FIGURE 2.6: Typical joint PDF of the whitened observed sequence. [2]

2.6.2 Estimation of $\boldsymbol{V}$

We can estimate $\boldsymbol{V}$ using MLE as,

$$\boldsymbol{V} = \underset{\boldsymbol{V}}{\operatorname{argmin}} \sum_{i} \boldsymbol{H}[(\boldsymbol{V}\boldsymbol{x}\boldsymbol{w})_i] \tag{2.26}$$

where $\boldsymbol{H}$ is the entropy.

(2.26) represents an optimization problem. Below is a summary of the $\hat{\boldsymbol{s}}$ estimation:

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{A}\boldsymbol{s} && (2.27)\\ \hat{\boldsymbol{s}} &= \boldsymbol{W}\boldsymbol{x} && (2.28)\\ \boldsymbol{W} &= \boldsymbol{A}^{-1} && (2.29)\\ &= \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T && (2.30) \end{aligned}$$

Source signals can be estimated using observed data as:

$$\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x} \tag{2.31}$$

2.7 Blind Source Localization

BSS algorithm operates in Time-Frequency bins called as TF bins. These TF bins are the division of the audio recording or, in mathematical terminology, the observation vector, into frames to keep the dimensions of the unmixing matrix computationally viable. This makes BSS prone to the permutation ambiguity problem. This means that after the source signals are recovered or separated, the alignment of the components separated in each time-frequency bin is the next challenge.

To solve this challenge, BSS algorithms calculate the Direction of Arrival (DoA) of the separated components. The separated components having the same DoA are aligned together to create the separated source signals. The following is the detail of how DoA is calculated in a BSS system.

2.7.1 Direction-of-Arrival

DoA of an acoustic source can be estimated by finding the Time Difference of Arrival (TDoA) of the signal at sensor nodes. TDoA can be measured by locating the peak in the cross-correlation of the signal received at the two sensor nodes [13]. It is assumed that the distance between the acoustic source and the microphones is fairly large as compared to the separation of the microphone pair i.e., far-field assumption. The far-field assumption is valid if the separation between the speaker and the microphone is greater than or equal to $\frac{2D^2}{\lambda_{min}}$ [14], where, d is the distance between the speaker and the microphone, D is the array aperture and λ_{min} is the minimum wavelength in the acoustic signal. This ensures that the wave-front arriving at the microphones is planer and not spherical. We also assume that the individual source signals are non-gaussian. The assumption means that only one of the acoustic sources is active during each time-frequency (TF) bin of the BSS unmixing filter. Using this widely accepted assumption [15] also called W-disjoint orthogonality [16], we can measure the Time Difference of Arrival (TDoA) of the source signals. With the arrangement shown in Figure 2.7, (2.32) can be used to calculate DoA from measured TDoA:

$$\theta = sin^{-1}\left(\frac{\Delta\tau \times v}{d}\right) \tag{2.32}$$

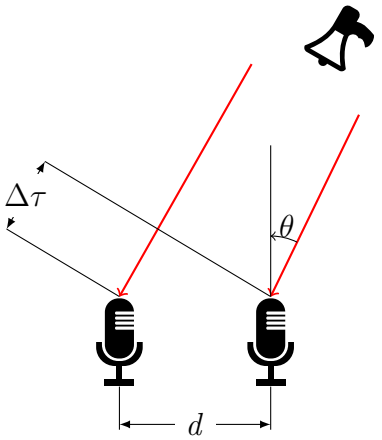


FIGURE 2.7: Signals from an acoustic source arriving at a two-microphone array. Note that the parallel sound waves indicate that the acoustic source is located in far-field of the microphone array.

where,

$$\begin{aligned} \theta &= \text{DoA of the sound signal} \\ \Delta\tau &= \text{time difference of arrival} \\ v &= \text{speed of the sound signal} \\ d &= \text{microphone pair separation} \end{aligned}$$

Chapter 3

Literature Survey

3.1 Overview

Every research work has to be unique. There is no value, as the famous proverb says, in *Reinventing the Wheel* no matter how much effort someone puts in it. The uniqueness of research work is determined by a survey of the related work conducted in a certain area of research. This chapter focuses on some of the most recent and prominent research work conducted in the area of speaker localization using microphone arrays.

3.2 Related Work

Source localization has been a hot research topic for many years due to its wide range of applications in speaker tracking, camera assistance, teleconferences, speech enhancement, military surveillance, hearing aid devices, and seismic activity positioning, etc. When combined with source separation, it forms the basis for acoustic scene analysis [17]. [18] proposed a robust method of speaker localization using distributed microphone arrays with the baseline DoA estimates coming from BSS algorithms run on each microphone pair. The proposed algorithm employs two microphone arrays each having two omnidirectional microphones for localization of two speakers – *Determined BSS* scenario. The paper studies the effect of separation between microphone arrays, distance of acoustic sources on the calculated DoAs and estimated source positions in a 10 m $\times$ 10 m room. The results show that as the distance between the microphone arrays is increased, the region of the room where localization is successful increases. If the error in DoA estimation is large, the algorithm performs poorly no matter how close the sources are to the microphone arrays and how much the microphone arrays are separated. The algorithm is tested in two configurations: when the microphone arrays are 7 m

apart (case 1) and when they are 1 m apart (case 2). The overall performance of case 1 is 33% better than that of case 2.

An *over-determined source localization* and separation approach is presented in [19]. The proposed algorithm reformulates the conventional over-determined mixing procedure and applies a determined Independent Component Analysis (ICA) to each frequency bin directly. This new method shows better separation accuracy. The paper also proposed a method to solve the problem of permutation ambiguity. The separated frequency components are first clustered using a time-activity based clustering method. Then, the channels from the same source are detected by analyzing time and frequency activities, spatial likeliness and spatial location in a remixing procedure. The spatial location is estimated using a time-frequency based algorithm. Results show that by using more microphones than the active speakers, the algorithm provides improved Signal to Interference Ratio (SIR). The time required for localization increases almost linearly as the number of microphones increase. When run on the real data, the algorithm successfully localized 7 out of 10 sources. The sources need to be static for a sufficiently long time and extension to dynamic speakers is left for future work.

An *under-determined source localization* algorithm is presented in [20] using acoustic sensor networks and Directivity Pattern Approach (ADP). It explicitly models the influence of reverberation in its DoA estimation and is found robust to reverberation and noise. Determined TRINICON BSS systems are employed at each sensor node to calculate the demixing filters which are later used to calculate the ADP-energy map. The algorithm is tested in a simulated environment with three active speakers and eight two microphone arrays distributed along the walls of the room. The presented results show that the algorithm produces precise results for three simultaneously active speakers.

[15] presents an extension of the GMM based localization method proposed in [21] by using a more appropriate probabilistic model based on a von Mises mixture model. The algorithm is capable of localizing an unknown number of acoustic sources in an enclosure. The algorithm chooses the DoA of each time-frequency (TF) bin as a feature for clustering. The results show that the proposed algorithm outperforms the GMM based localization algorithm.

Table 3.1 presents a summary of above discussion.

TABLE 3.1: Summary of related work.

Paper ref.	Sensor type	Reverb. Env.	Field of view	Separation	BSS as baseline	Mic. separation
[18]	Distrubuted	No	Far	Determined	Yes	unknown
[19]	Distrubuted	No	Far + Near	Determined	No	variable
[20]	Distrubuted	Yes	Far	Under-Determined	No	20 cm
[15]	Distrubuted	Yes	Far	Determined	No	20 cm
Proposed Work	Condensed	Yes	Far	Under-Determined	Yes	9.26 cm

Chapter 4

System Design

4.1 Overview

For every speech processing task, the basic elements are microphones and the speech signal. Microphones record the speech signal for using later in off-line processing. In this section, we introduce the reader with the infrastructure developed for carrying out the experiments. We explain the recording setup, the microphone array that we used, recording configurations, and the surroundings.

4.2 Microphone Array

Microphones arranged in a certain geometry with the capability to synchronously receiving a sound signal is called a *Microphone Array*. In this research work, we used a microphone array from Seeed Studio called ReSpeaker Core v2.0. The array houses six microphones on the periphery arranged in the shape of a hexagon. The module has 1GB on main memory and is powered by a quad-core ARM® Cortex-A7 microprocessor that can clock up to 1.5 GHz. The diagonal facing microphone separation is 9.26cm. The details can be viewed at the wiki page http://wiki.seeedstudio.com/ReSpeaker_Core_v2.0/.

The main motivation behind using a six microphone array is that many different geometries can be made from such an array such that hexagonal, triangular and rectangular.

4.3 Corpus Collection

For this research work, we conducted audio recording sessions in real-world scenarios, for example, reverberant and studio environments. The speakers were made to sit around the microphone array in a way similar to a round table conference.

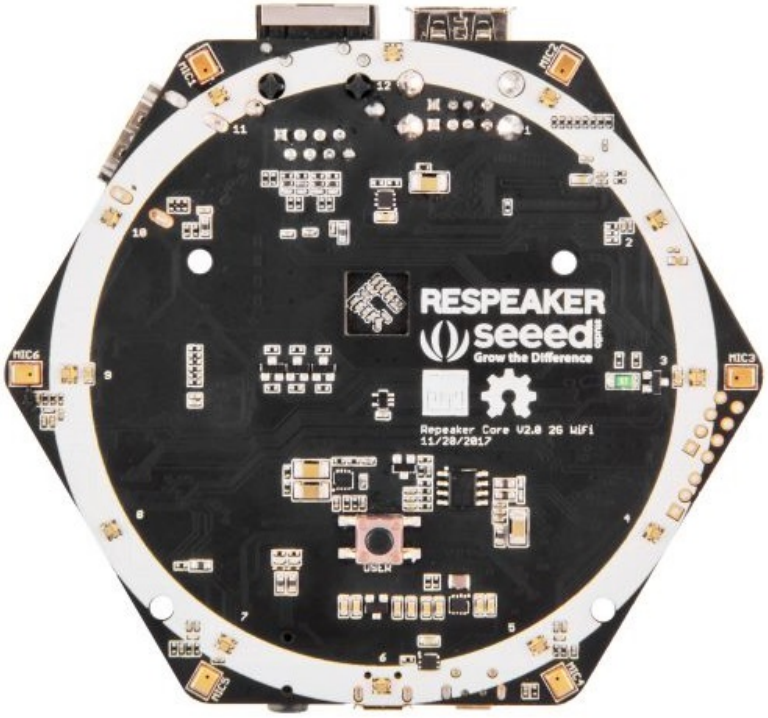
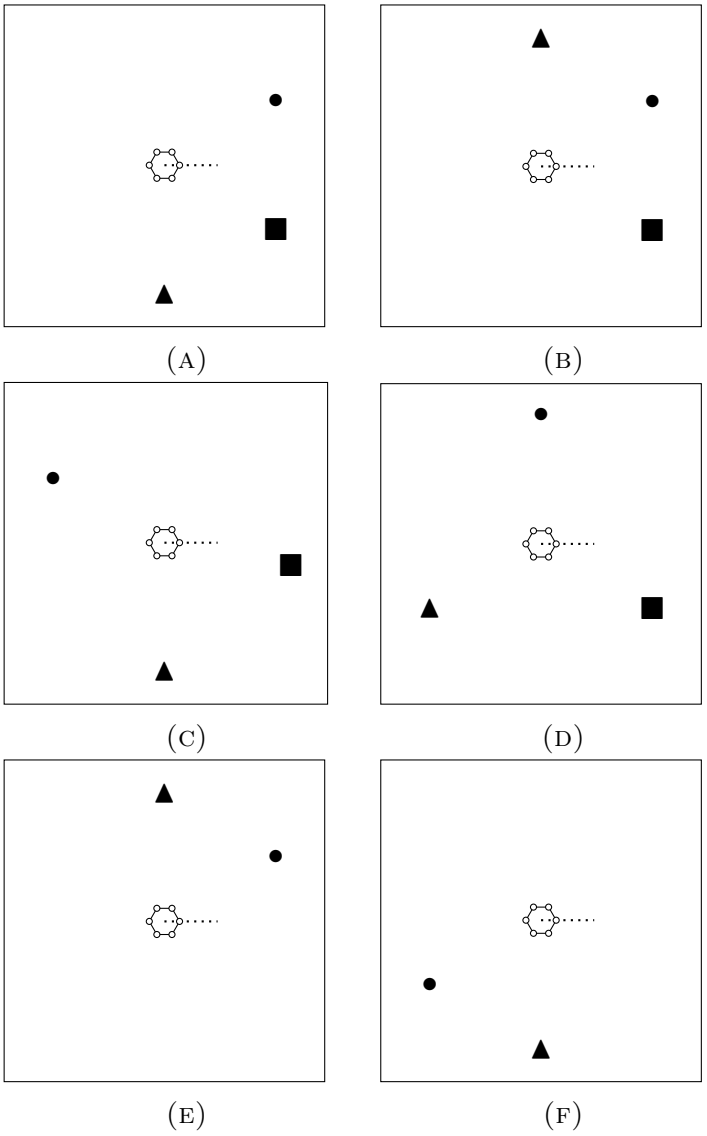


FIGURE 4.1: ReSpeaker Core v2.0 from Seeed Studio

The corpus contains five three-speaker recordings and five two-speaker recordings each with a duration of over 10 sec with speakers sitting in different configurations around the microphone array at a sampling frequency of 48 kHz. Figure 4.2 shows six out of ten different speaker configurations used in corpus collection. We made sure that the speakers remain in the far-field of the microphone array and do not come too close to it. The reason for the far-field assumption has been explained earlier in this paper. We also made sure that the speakers and the microphone array remain in a horizontal plane. The algorithm is tested on these real-world recordings and the results are presented at the end of this paper.

The selection of the microphone array for carrying out our experiments was vital. In this research work, we used ReSpeaker Core v2.0 from Seeed Studio. The array houses six microphones on the periphery arranged in the shape of a hexagon. The diagonal facing microphone separation is 9.26cm. Further details can be viewed on the wiki page http://wiki.seeedstudio.com/ReSpeaker_Core_v2.0/.

The motivation for choosing a hexagonal microphone array is the benefits it offers. First of all, the spatial resolution is high as two adjacent microphones subtend an angle of sixty degrees at the center of the microphone array. Another advantage is that many different regular shapes can be made out of hexagonal geometry, such as triangular, circular and rectangular.



(A) (B) (C) (D) (E) (F)

FIGURE 4.2: Figure showing different speaker arrangements around the microphone array. The hexagon in the center is the microphone array. The bubbles on the periphery of the hexagon represents the six microphones. The horizontal dotted line from the center of the microphone array and outwards towards the right is the global reference axis. Different marks are assigned to distinguish between the three different speaker locations. • denotes the first speaker. ▲ denotes the second speaker and ■ denotes the third speaker. 4.2a has speakers arranged as 30°, 270° and 330° respectively with respect to the global reference axis. 4.2b has speakers arranged as 30°, 90° and 330° respectively. Similarly 4.2c has speakers arranged as 150°, 270° and 350° and 4.2d has speakers arranged as 90°, 210° and 330° respectively with respect to the global reference axis. 4.2e and 4.2f depict two of the many different tried speaker recording configurations.

Chapter 5

Proposed Methodology

5.1 Basic Idea

We consider a microphone array of M microphones and P concurrently active speakers located in the far-field of the microphone array. The microphone array is held static in its place while the speakers can be static, periodically moving or continuously moving. A mathematical model of the speech signals received at the microphones is presented in (5.1) [22]:

$$x_i(t) = \sum_{g=1}^{P} s_g h_{ig}(t - t_i(\theta_g)) + n_i(t) \tag{5.1}$$

where s_g is the g^{th} speaker, h_{gi} is coefficient of the mixing matrix between the g^{th} speaker and the i^{th} microphone, $t_i(\theta_g)$ is the propagation time of the signal from g^{th} source to the i^{th} microphone. θ_g is the DoA of the source s_g and $n_i(t)$ is the additive white noise.

We aim to design and develop a localization algorithm by using the BSS algorithm as the baseline. As the existing BSS algorithms are geometrically restrictive, the proposed algorithm shall exploit the geometry of the microphone array to solve this problem, hence extending the localization capability of the BSS algorithm to variable geometries. The currently available BSS algorithms are limited to a linear microphone pair and can not go beyond it in two-dimensional space. We present an extensive scheme as presented in Figure 5.1. A multi-microphone array is used for recording audio samples of multiple concurrent speakers. The surroundings for the recording set-up can be studio environment, noisy outdoors or reverberant

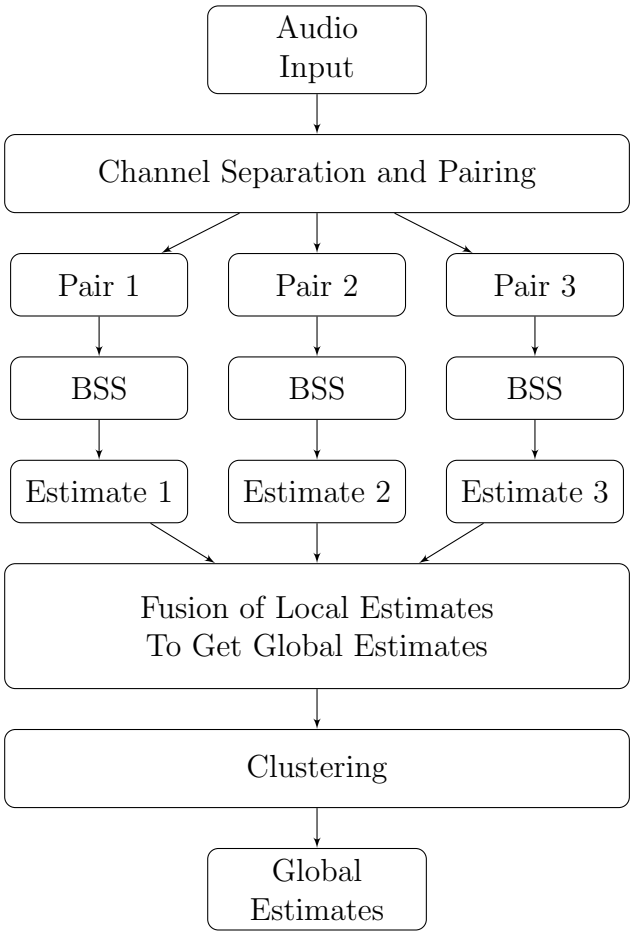


FIGURE 5.1: Methodology

rooms. Details about the corpus collection have already been explained. With the BSS algorithm as the baseline, the proposed localization algorithm does not need to have any apriori knowledge of the characteristics of the acoustic source or the propagation channel. The speakers and microphone array are assumed to be on the same plane.

5.2 Channel separation and pairing

As the existing BSS algorithms can not go beyond a linear one-dimensional microphone array, we propose resolving the geometry of the microphone array into pairs of equidistant microphones, hereby, making sets of bichannel audio recordings. The microphone pairing is done manually to utilize the spatial diversity of the microphone array to the maximum.

The microphones are numbered from 1 to 6 for convenience. Next, the diagonal facing microphones are grouped into pairs. This way, we get three pairs in total. We devise a notation [i-j] to denote a pair of i and j microphones. The three pairs we get in this way are [6-3], [5-2] and [4-1]. For simplicity, we shall call

them *Pair 1*, *Pair 2* and *Pair 3* respectively. This pairing scheme is depicted in Figure 5.2. The dotted lines indicate microphone pairs. This enables us running the BSS algorithm separately on each microphone pair. Channels of the audio file recording are first separated and then grouped according to the microphone pairs. This gives us a set of three bichannel audio files.

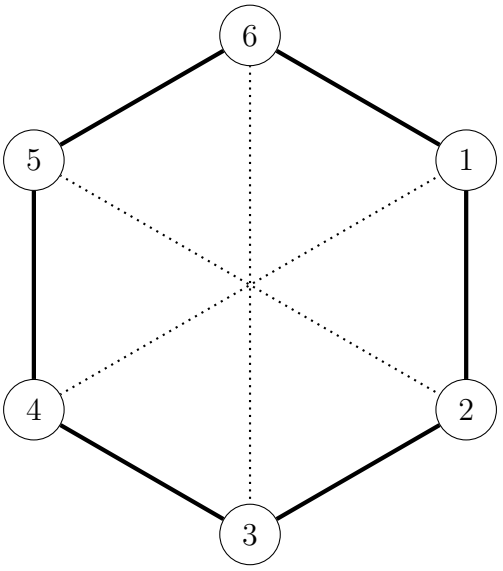


FIGURE 5.2: The pairing of microphones of the array. The numbered nodes from 1 to 6 indicate six microphones. Dotted lines indicate the pairing scheme.

5.3 Running the BSS algorithm

Each of the audio files is then fed separately to the TRINICON BSS [23] algorithm. The algorithm was used in offline block processing mode with the following settings:

- Unmixing filter length = 1024
- Offline block size = 8192 samples
- Block overlap = 50%
- BSS filter iterations = 20

The algorithm outputs local DoA estimates of the active speakers. For each BSS time-frequency bin, we obtain $N \times P$ DoA estimates, where, N is the active number of speakers and P is the number of microphone pairs. The estimates will be measured with respect to their respective microphone pair's reference axis. Figures 5.3, 5.5 and 5.4 plots the local estimates generated by the BSS algorithm

for each channel of the mentioned microphone pair for each TF bin of the audio sample. As the initial DoA estimates are provided by the baseline BSS algorithm, the accuracy of this system is directly impacted by the accuracy of the underlying BSS algorithm in identifying the mixing system.

5.4 Rotation and Fusion of Local Estimates

To get global speaker location estimates, the local DoA lines first need to be measured from the global reference axis. We call it *rotation of local estimates*. We consider the line joining the center of the microphone array to the microphone number 2 as the global reference axis. There is no specific reason for choosing this as the global reference axis. Any other line originating from the center of the microphone array and moving towards the periphery can also be chosen. Figure 5.6 shows a pictorial representation of this process. After choosing the global reference axis, conversion from local DoA estimates to global DoA estimates is adding the angle that the local reference axes make with the global reference axis to the local DoA estimates. This rotation is done to account for the angular displacement of the microphone pair with the global reference axis. Following equations are used for this purpose:

$$\theta_1 = \theta_{1\_l} + \theta_{1\_axis} \tag{5.2}$$

$$\theta_2 = \theta_{2\_l} + \theta_{2\_axis} \tag{5.3}$$

$$\theta_3 = \theta_{3\_l} + \theta_{3\_axis} \tag{5.4}$$

$$\vdots$$

$$\theta_n = \theta_{n\_l} + \theta_{n\_axis} \tag{5.5}$$

where, θ_1, θ_2 and θ_3 are the converted global DoA estimates, θ_{1_l}, θ_{2_l} and θ_{3_l} are the local DoA estimates and θ_{1_axis}, θ_{2_axis} and θ_{3_axis} are the angles that the reference axis of *Pair 1*, *Pair 2* and *Pair 3* make with the global reference axis. θ_{n_axis} can be any of θ_{1_axis}, θ_{2_axis} or θ_{3_axis} depending on the n^{th} DoA line. We defined our observation vector as:

$$\boldsymbol{\theta} = \{\theta_1, \theta_2, \theta_3, ..., \theta_n\} \tag{5.6}$$

5.5 Global Coherence Map

The rotated DoA lines are then drawn with respect to the global reference axis to create a Global Coherence Map (GCM) as shown in Figure 5.7. A GCM is created

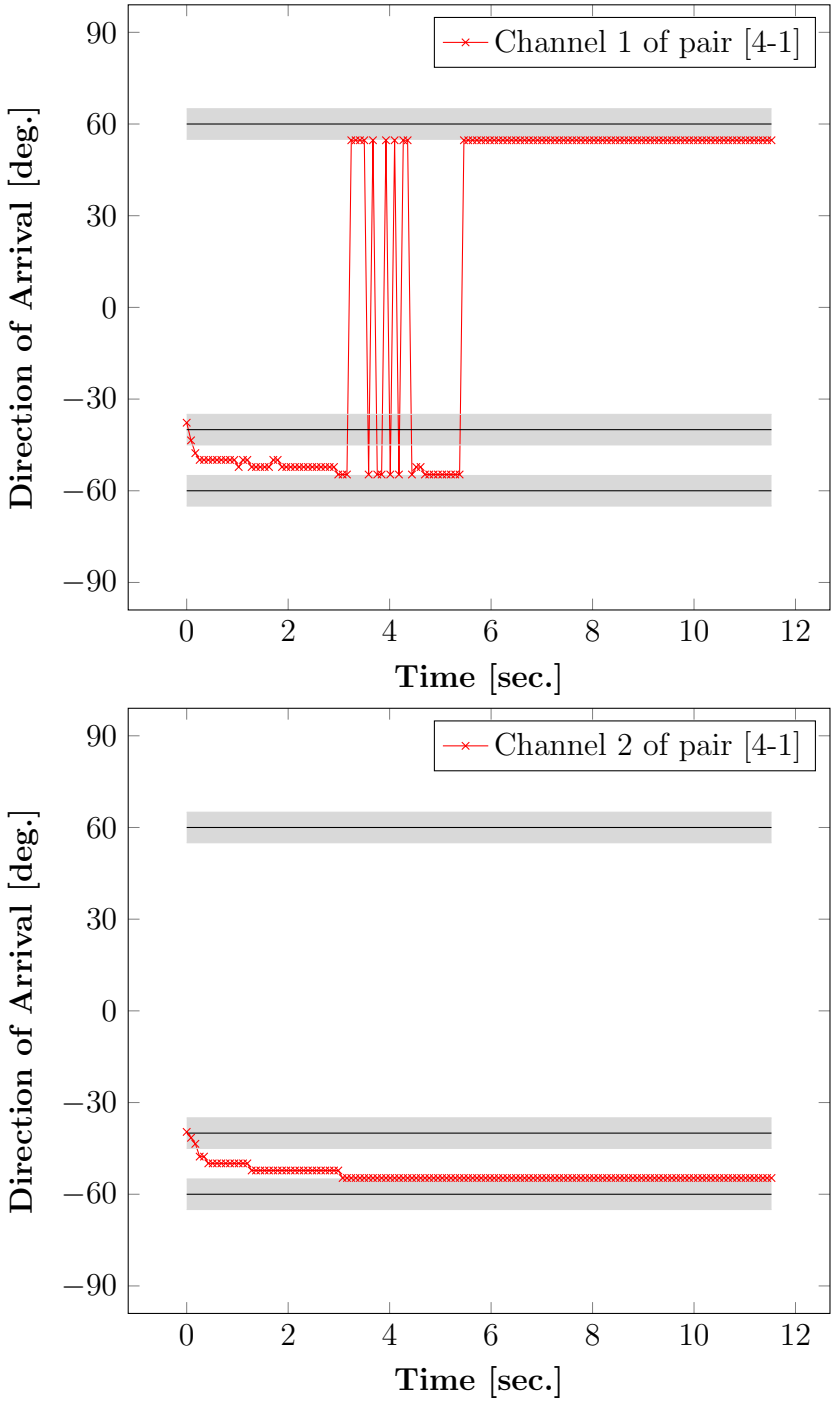


FIGURE 5.3: Channel-wise plot of local DoA estimates generated by the baseline TRINICON BSS algorithm for microphone pair [4-1]. The audio recording used is 11.6 seconds in duration and the speakers were arranged at 150°, 270° and 350° with respect to the global reference axis. The horizontal black lines indicate the ground truth of the three active speakers around the microphone array relative to the reference axis of the microphone pair under consideration. The grey region is the ±5° of error margin in the ground truth to account for the unintentional head movement during speech.

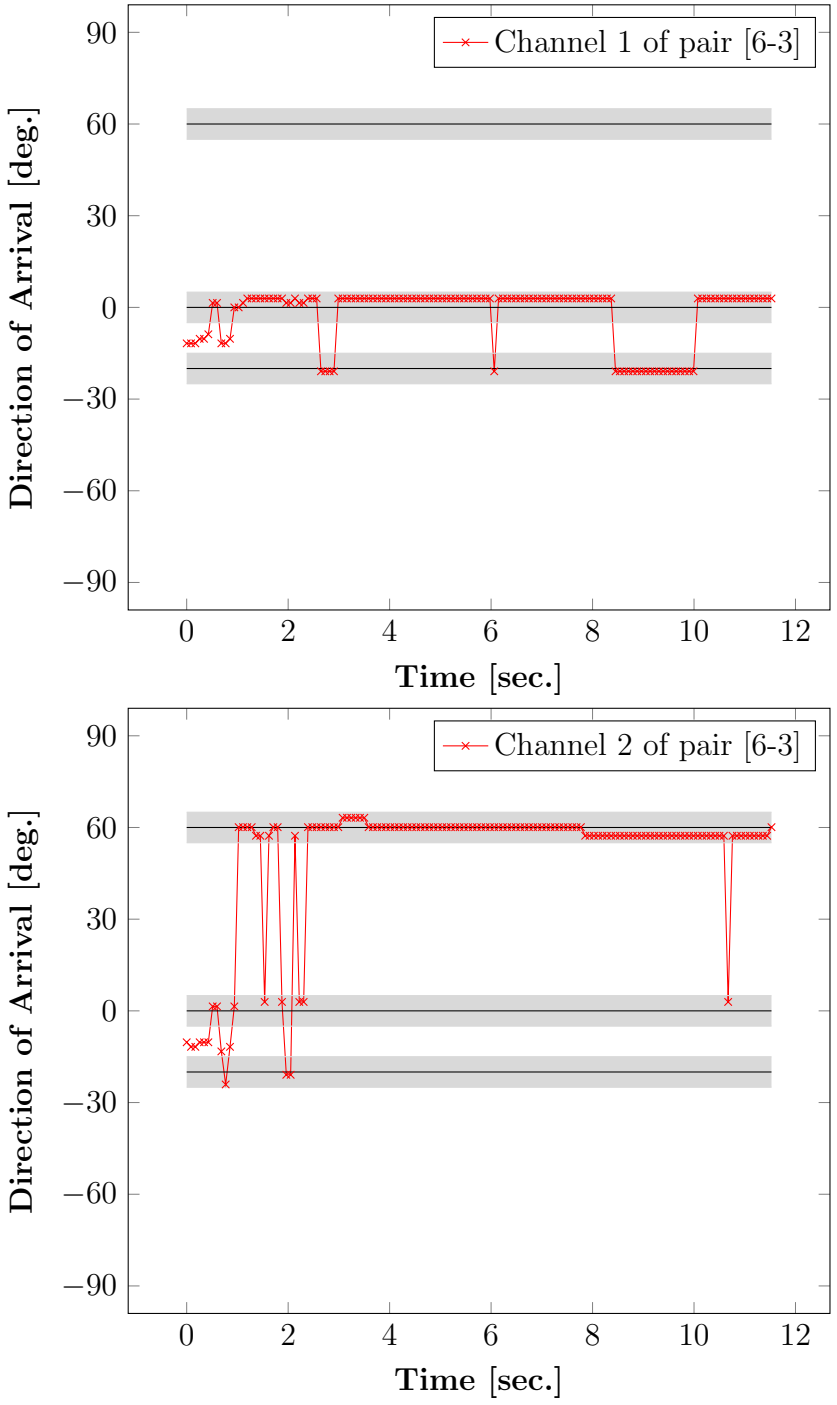


FIGURE 5.4: Channel-wise plot of local DoA estimates generated by the baseline TRINICON BSS algorithm for microphone pair [6-3]. The audio recording used is 11.6 seconds in duration and the speakers were arranged at 150°, 270° and 350° with respect to the global reference axis. The horizontal black lines indicate the ground truth of the three active speakers around the microphone array relative to the reference axis of the microphone pair under consideration. The grey region is the ±5° of error margin in the ground truth to account for the unintentional head movement during speech.

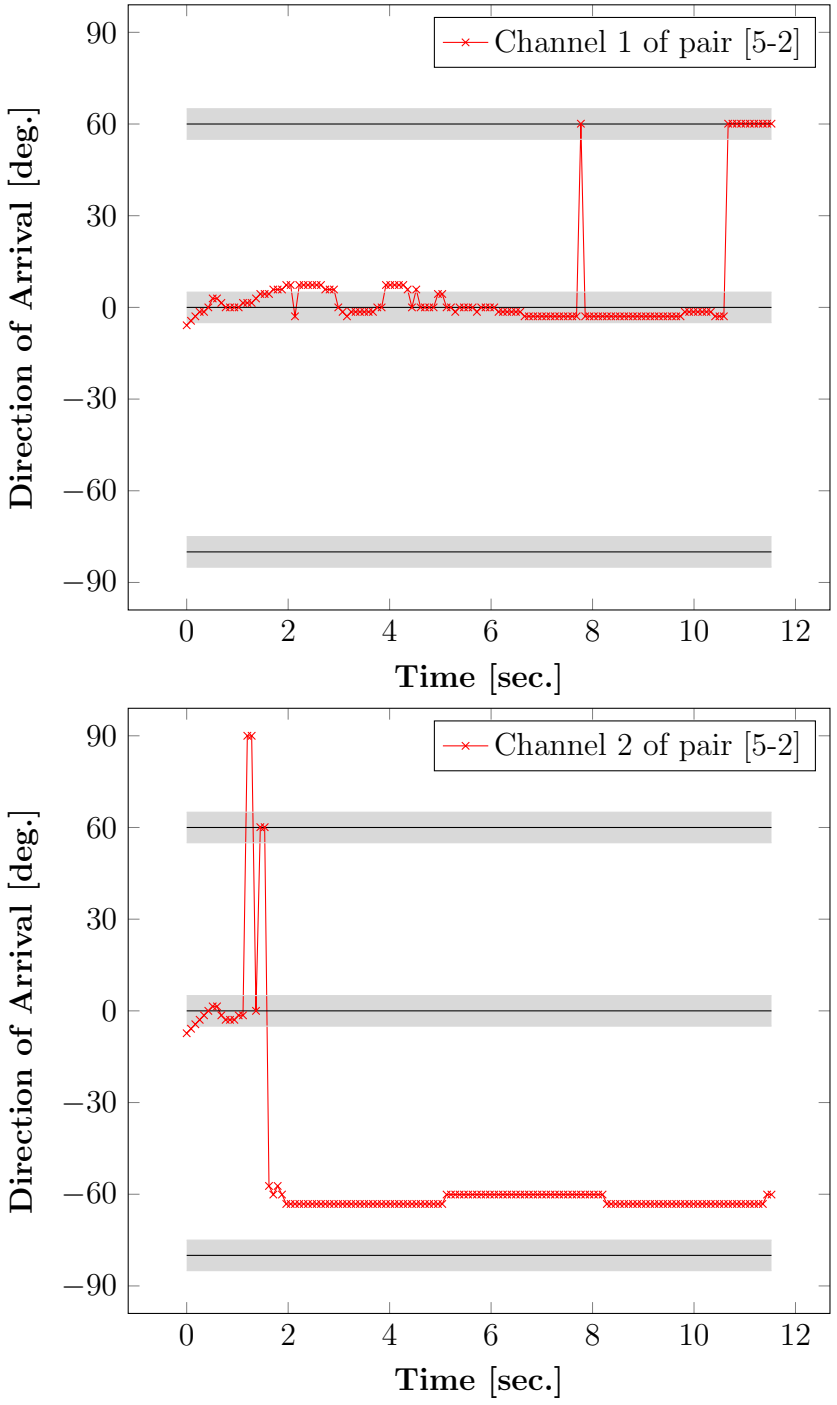


FIGURE 5.5: Channel-wise plot of local DoA estimates generated by the baseline TRINICON BSS algorithm for microphone pair [5-2]. The audio recording used is 11.6 seconds in duration and the speakers were arranged at 150°, 270° and 350° with respect to the global reference axis. The horizontal black lines indicate the ground truth of the three active speakers around the microphone array relative to the reference axis of the microphone pair under consideration. The grey region is the ±5° of error margin in the ground truth to account for the unintentional head movement during speech.

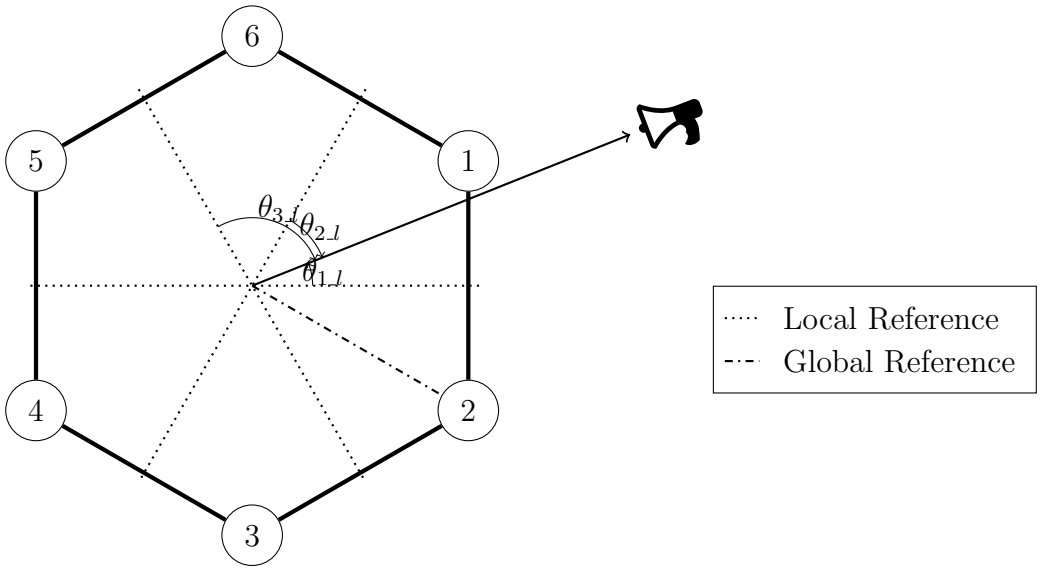


FIGURE 5.6: Depicts how the local estimates are converted into global estimates. θ_{1_1}, θ_{2_1} and θ_{3_1} are the local estimates with respect to the reference axes of *Pair 1*, *Pair 2* and *Pair 3* respectively.

by drawing all the DoA estimates in the observation vector $\boldsymbol{\theta}$ in a 2-dimensional plane.

5.6 Clustering and Finding Global Estimates

The next step is to cluster the observation vector to get the final speaker location estimates. The choice of the clustering algorithm is critical here. A study at http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/modules/clustering.html compares five of the well-known clustering algorithms. Among them, only DBSCAN [3] and Spectral Clustering algorithms correctly identify Gaussian clusters. The comparison is shown in Figure 5.8.

For our study, we select DBSCAN for two reasons. First, it is completely unsupervised and dynamically detects the total number of clusters in the dataset. The second being its execution speed. [3] claims that it only takes 41.7 seconds to finish its clustering on 12512 data points. Figure 5.7 shows the results of the DBSCAN clustering algorithm. It clusters the observation vector into m clusters:

$$\hat{\boldsymbol{\Theta}} = \{\hat{\boldsymbol{\Theta}}_1, \hat{\boldsymbol{\Theta}}_2, \hat{\boldsymbol{\Theta}}_3, ..., \hat{\boldsymbol{\Theta}}_m\} \tag{5.7}$$

where,

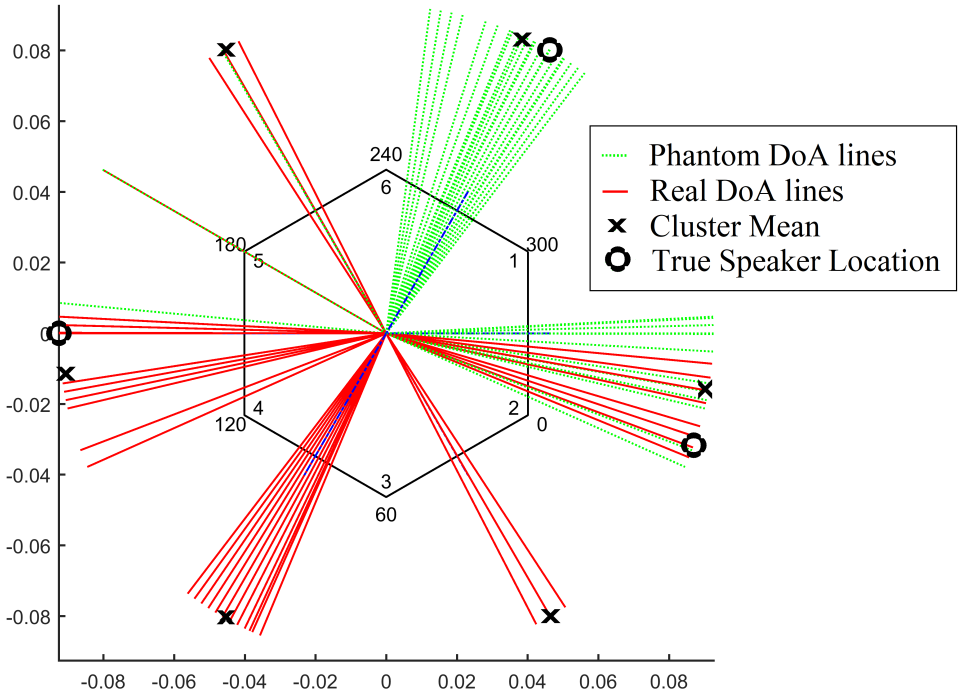


FIGURE 5.7: Global Coherence Map of three concurrent speakers. The hexagon in the middle represents the microphone array. The × marks are the cluster means of the clusters identified by the DBSCAN [3] clustering algorithm.

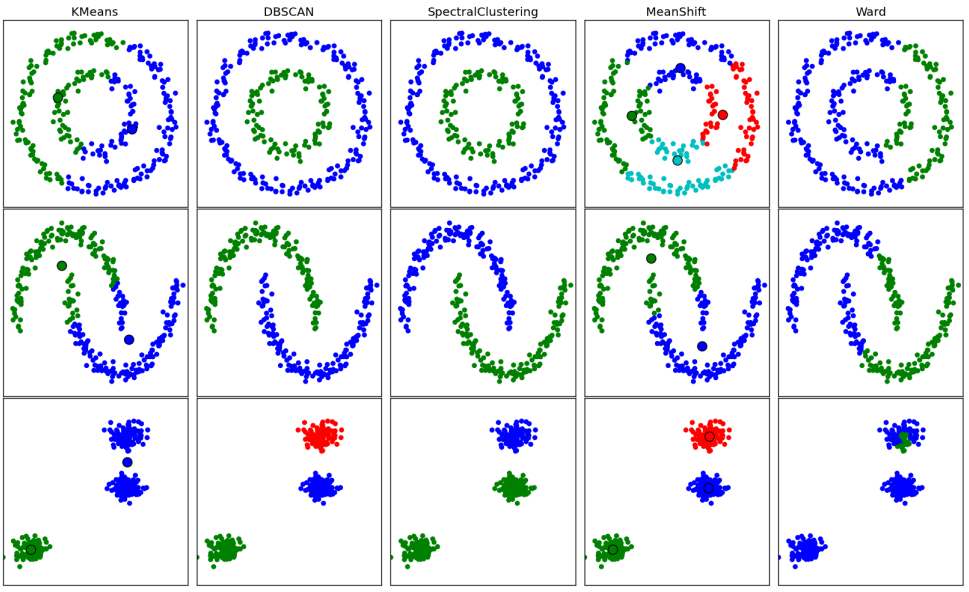


FIGURE 5.8: DBSCAN out performs all other clustering algorithms. [24]

$$\hat{\mathbf{\Theta}}_k = \{\theta_{1k}, \theta_{2k}, ..., \theta_{jk}\} \qquad 1 \leq k \leq m \tag{5.8}$$

5.7 Estimation of the speaker locations

The clusters identified in the last step indicate the global speaker positions on the GCM. Ideally, there should be as many clusters as the number of active concurrent speakers in the audio recording. The exact angles can be found by calculating the mean of the clusters. This is formulated in (5.9).

$$\mu_k = \overline{\hat{\mathbf{\Theta}}_k} = \frac{1}{n_k} \sum_{i=1}^{n} \theta_{ik} \qquad 1 \leq k \leq m \tag{5.9}$$

where, n_k is the number of elements in the k^{th} cluster.

During this research work, we encountered some challenges. Following sections explain those challenges and the solutions we devised to get over them:

5.8 Main Challenges

So far, the development of the algorithm seemed like a smooth ride from audio recording to finding the global speaker estimates. In the next sections, we explain the challenges we encountered during this thesis.

5.8.1 Front-back Disambiguation

BSS algorithms, as they can only operate on a linear two microphone linear array, intrinsically suffer from the problem of front-back ambiguity as the TDoA of the signal coming from the front side or the backside is the same. The range of vision is limited to -90 $\leq 0 \leq$ +90 degrees i.e., front horizontal space or the rear horizontal space [25]. This not only limits the localization capability to 1D but also makes it ambiguous as to which side of the array, the speaker is located. For a distributed microphone array setup, this is not a problem at all as one side of the array is fixed against a wall. But for a condensed microphone array, this complicates the localization problem even further.

To overcome this problem, we assign equal probability to both front and rear sides of the microphone pair i.e., for every θ_N, we assume θ'_N which is a reflection of θ_N in the rear horizontal space. This is indicated in Figure 5.9. Only one of θ_N and

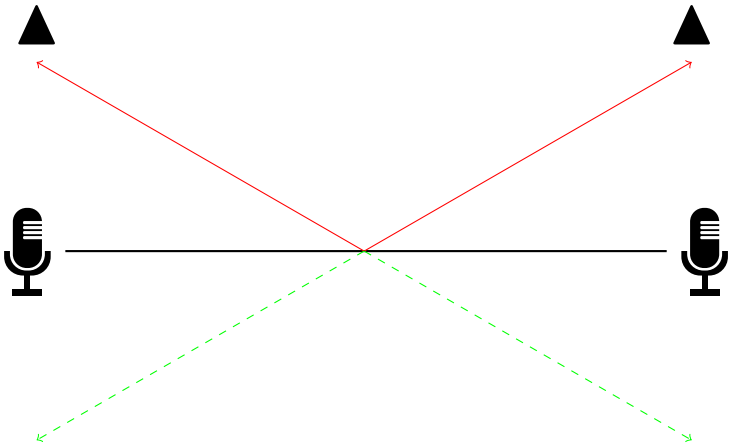


FIGURE 5.9: Illustration of real and phantom DoA. The real sources are indicated by ▲ and their corresponding real DoA lines are indicated by black arrows. The phantom DoA lines are drawn in green which are merely reflections of the real ones.

θ'_N can be real while the other is called phantom estimate. In this way, the total number of DoA estimates are doubled. For determined source localization using a microphone pair, we get $2 \times 2 = 4$ DoA estimates in total. Since we have three microphone pairs in the hexagonal array, the total number of estimates we get is $4 \times 3 = 12$. The identification scheme for real and phantom estimates is explained in later sections.

5.8.2 Localization of more than two speakers

The baseline BSS algorithm that we used is a determined BSS system. To localize more than two concurrent speakers using the BSS algorithm, we exploit the geometry of the microphone array. We tested our system on three concurrent speakers. When an under-determined audio recording is fed to a determined BSS algorithm, the algorithm correctly identifies any two of the three active speakers while discards the third one. The selection of the two sources that are correctly identified depends on the relative loudness of the sources as well as their direction of arrival at the microphone pair. If the loudness of the acoustic sources is comparable, the algorithm keeps on jumping between the three sources, picking any two of them at a time. Depending on the orientation of the microphone pair, if an acoustic source is outside its range of localization, it will be dropped throughout the audio recording.

Noise levels also play their part in contaminating the audio file. If the levels are too high, false positives are also detected and they appear in the DoA plot. Figures 5.3, 5.5 and 5.4 plots the direction of arrival identified by the determined-BSS algorithm for each channel of the three microphone pairs ([6-3], [5-2] and [4-1])

versus time. The jumpiness between three sources is visible while false positives can also be seen at some points.

5.8.3 Removal of phantom sources

As can be seen in Figure 5.7, there are more clusters identified in the GCM than there are speakers. The simplest approach to identify the most probable speaker locations is to pick the bulkiest clusters. But this approach does not always bring correct results. We devise a heuristic to find the correct clusters. The first step is to sort the clusters in descending order of their sizes. This gives us $\mathbf{\Theta}_{sorted}$.

$$\mathbf{\Theta}_{sorted} = \{\mathbf{\Theta}_1, \mathbf{\Theta}_2, \mathbf{\Theta}_3, ..., \mathbf{\Theta}_m\} \tag{5.10}$$

Next, we start picking up the bulkiest clusters one by one. As a cluster is picked, its phantom DoA's are removed from the GCM and the new GCM is clustered again. This process continues until we get an estimate for all of the speakers. This is explained through the Algorithm 1.

Algorithm 1 Removal of phantom sources

$\theta \leftarrow$ *observation vector*
$\epsilon \leftarrow$ *dbscan cluster radius*
$minPts \leftarrow$ *dbscan cluster size*
$N \leftarrow$ *number of speakers*
$i \leftarrow 0$
for $i < N$ **do**
    $\hat{\Theta} \leftarrow \text{dbscan}(\theta, \epsilon, minPts)$
    $\Theta_{sorted} \leftarrow \text{sort}(\hat{\Theta}, descending)$
    $\Theta_i \leftarrow \Theta_{sorted}[i]$
    $j \leftarrow 0$
    for $j < \text{size}(\Theta_i)$ **do** *remove phantom of* $\Theta_i[j]$ *from* θ

For the last speaker, if there is more than one cluster of the same size, we calculate which cluster is more concentrated towards its mean. This is found by summing the distance between each point and the cluster mean. The cluster with minimum distance is picked as the estimate. Algorithm 2 outlines this process.

$$\mathbf{\Theta}_{Nn} = \underset{\mathbf{\Theta}_k}{\text{argmin}} \sum_{\theta_{ik} \in \mathbf{\Theta}_k} \mid \theta_{ik} - \mu_k \mid \tag{5.11}$$

Algorithm 2 Density based cluster selection

$N \leftarrow$ *number of speakers*
$\theta \leftarrow$ *observation vector*
$\Theta_{sorted} \leftarrow$ *cluster vector*
$remainingClusters \leftarrow N$
$\Theta_{picked} \leftarrow [\,]$ ▷ Empty array
$i \leftarrow 0$
$j \leftarrow 0$
while $i < N$ **do**
    $\Theta \leftarrow \Theta_{sorted}[i]$
    $size \leftarrow \text{size}(\Theta)$
    $numEqualClusters \leftarrow \text{findClusters}(\Theta_{sorted},\ size)$
    if $numEqualClusters > remainingClusters$ **then**
        ▷ Calculate densities of all equal sized clusters
        while $j < numEqualClusters$ **do**
            $\sigma[j] \leftarrow \text{findDensity}(\Theta)$
        ▷ Retrieve the densest clusters.
        $temp[\,] \leftarrow \text{returnDenseClusters}(\Theta,\ \sigma,\ remainingClusters)$
        ▷ Add the retrieved clusters to Θ_{picked} array and discard rest.
        $\Theta_{picked} \leftarrow \{\Theta_{picked},\ temp[\,]\}$
        $i \leftarrow \text{size}(\Theta_{picked})$ ▷ Increment i.
    else
        ▷ Pick all clusters of given size and add them to Θ_{picked} array.
        $\Theta_{picked} \leftarrow \{\Theta_{picked},\ \text{returnClustersOfSize}(\Theta,\ size)\}$
        $i \leftarrow \text{size}(\Theta_{picked})$ ▷ Increment i.

Chapter 6

Results and Discussion

6.1 Overview

Performance evaluation is the most important step in the development of every algorithm. How accurate an algorithm produces the results defines its performance. Apart from the accuracy, the running time, memory footprint, future extendibility, and the tolerance to external factors are also important in determining the value of a contribution.

In this chapter, we measure the accuracy of our algorithm and present the results. We also compare the stability of the output estimates of the proposed algorithm with those of the Steered-Response Power Phase Transform (SRP-PHAT) algorithm and it is found that the proposed algorithm is relatively stable in offline processing mode for multiple concurrent speakers. The later part discusses what these results mean, the justification behind any shortcomings and the future prospects of this contribution.

6.2 Results and Discussion

We tested our algorithm on the collected corpus. Table 6.1 summarizes the underdetermined source localization results we obtained. Root Mean Square Error (RMSE) is used as a performance metric. (6.1) is used to calculate the RMSE.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(\theta_{estimated} - \theta_{actual}\right)^2}{N}} \tag{6.1}$$

Only one out of five different speaker configurations is misclassified. The RMSE is below 12 degrees in the rest of the cases.

TABLE 6.1: Results of the proposed algorithm on the corpus we collected. The first column shows the measured speaker locations. Each reading has a ±5° of margin added to account for the unintentional head movement during speech. 4^{th} column shows Root Mean Square Error in the readings.

Real location [deg.]	Estimated location [deg.]	Difference [deg.]	RMSE
30 ± 5 270 ± 5 330 ± 5	25.84 290.99 319.52	4.16 ± 5 20.99 ± 5 10.48 ± 5	14.01 ± 4.24
30 ± 5 90 ± 5 330 ± 5	24.69 99.93 349.17	5.31 ± 5 9.93 ± 5 19.17 ± 5	13.06 ± 4.39
90 ± 5 210 ± 5 330 ± 5	271.93 214.53 324.05	181.93 ± 5 4.53 ± 5 5.95 ± 5	105.20 ± 3.05
30 ± 5 195 ± 5 255 ± 5	37.24 191.55 260.41	7.24 ± 5 3.45 ± 5 5.4 ± 5	6.04 ± 4.45
150 ± 5 270 ± 5 350 ± 5	144.49 264.85 335.55	5.51 ± 5 5.15 ± 5 14.45 ± 5	9.75 ± 4.29

After successful results for three-speaker scenarios, we tested our algorithm on two-speaker scenarios as well. The speakers were arranged around the microphone array in a configuration similar to the three-speaker case. The algorithm successfully localized the speakers sitting around the microphone array and showed accurate results. The RMSE remained below 9 in all five speaker configuration. The results are summarized in Table 6.2.

To establish a basis for comparison, we ran SRP-PHAT algorithm on the same audio recordings and drew the GCM. Figure 6.1 shows the GCM obtained by running the SRP-PHAT algorithm on the same audio recording that we used to draw Figure 5.7 using the proposed algorithm. With no post-processing, it can be noted that the Figure 5.7 is less spread out than the Figure 6.1.

6.2.1 Speaker Tracking

We also used our algorithm for DoA based speaker tracking. The methodology was modified to run the algorithm on small overlapping frames, of duration 853.33 ms, of the audio recording. The overlap between the frames was fixed at 80% to ensure a smooth transition in the DoA estimates.

Another optimization that was made is that in the windowed operation of the

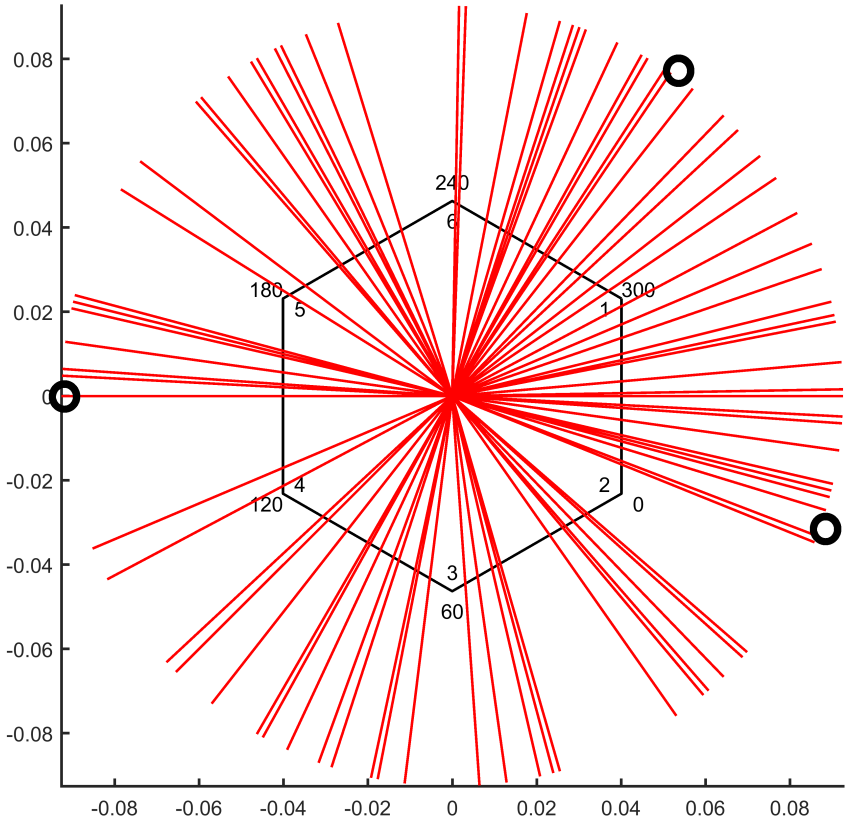


FIGURE 6.1: Global Coherence Map of three concurrent speakers obtained from the SRP-PHAT algorithm. The true speaker locations are marked by the black circles.

TABLE 6.2: Localization results of the proposed algorithm for two concurrently active speakers.

Real location [deg.]	Estimated location [deg.]	Difference [deg.]	RMSE
30 ± 5 90 ± 5	28.46 87.37	1.54 ± 5 2.63 ± 5	5.04 ± 2.07
30 ± 5 330 ± 5	25.04 335.19	4.96 ± 5 5.19 ± 5	5.11 ± 4.97
90 ± 5 150 ± 5	90.21 142.85	0.21 ± 5 7.15 ± 5	6.53 ± 2.82
100 ± 5 150 ± 5	107.62 141.15	7.62 ± 5 8.85 ± 5	8.27 ± 4.98
210 ± 5 270 ± 5	206.11 273.98	3.89 ± 5 3.98 ± 5	5.00 ± 3.94

algorithm, there is already very few DoA estimates available as compared to the case of static localization. We do not remove the phantom sources in this scenario, because, if we remove the phantom sources, the Algorithm 1 causes some of the real clusters to vanish for having lesser than the minimum required estimates for DoA clustering. The fake sources are rather suppressed by the size of the bulky clusters. With a lesser number of DoA sources available, the probability of phantom sources to gather and form a fake bulky cluster is also greatly reduced.

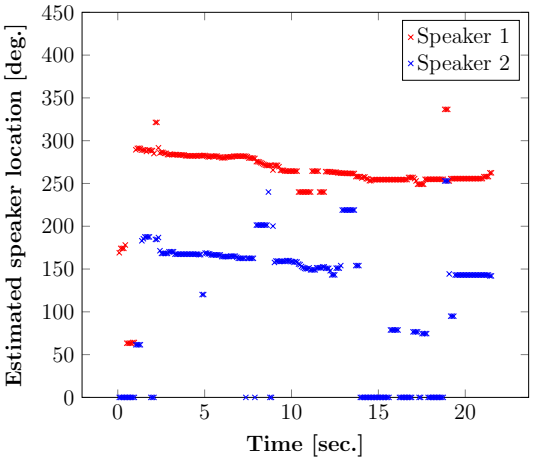
We tried our algorithm on two-speaker audio recordings with the speakers synchronously moving around the microphone array in one direction. The algorithm was able to capture the direction of movement of the speakers and the pattern of movement. The results for one of the cases is shown in Figure 6.2. These can be fed to a tracker algorithm for automated speaker tracking.

SRP-PHAT is well-known for its efficiency in speaker localization. For comparison, we ran the SRP-PHAT algorithm on the same two-speaker audio recording. The obtained results are presented in Figure 6.2c. It can be seen that in comparison to the results obtained by the proposed algorithm, the results of the SRP-PHAT algorithm are relatively unstable.

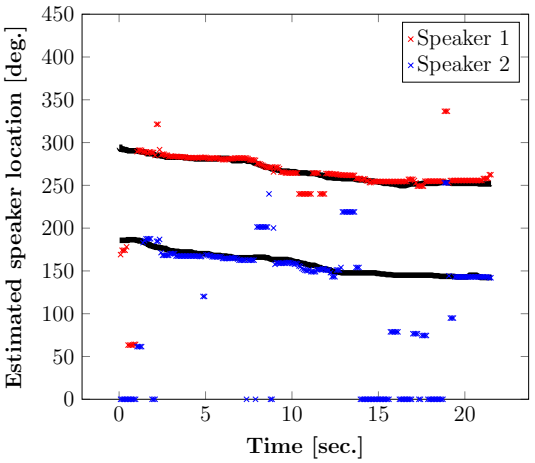
6.3 Conclusion and Future work

Speaker localization has been an area of research these days. In this research work, we proposed a microphone array-based source localization algorithm by using the BSS algorithm as the baseline. The proposed algorithm successfully localized three concurrently active speakers by exploiting the redundancy of the microphone array. The algorithm showed promising results on real-life audio recordings. The algorithm is also tried on two-speaker audio recordings and it showed accurate results without any modification in the algorithm. The algorithm was further extended for dynamic speaker tracking. The proposed algorithm showed the capability for tracking the moving speakers. Speaker localization and tracking using condensed microphone arrays are particularly useful in robotics [26, 27].

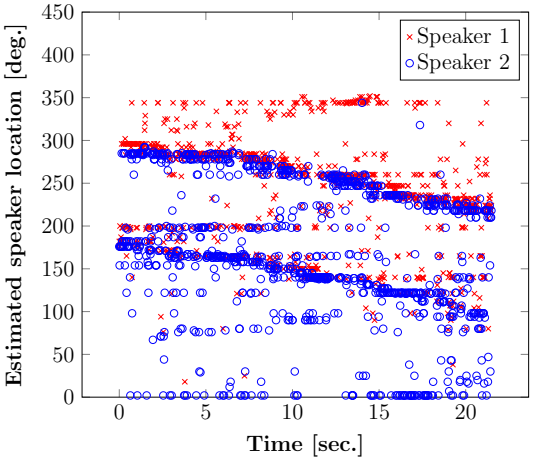
To further access the capability of the motion tracking version of the algorithm, we trial ran it for three speaker tracking. As a first step, we used a static speaker recording for this purpose. The results are presented in Figure 6.3. The algorithm was able to successfully localize and track two of the three concurrent speakers. This can be attributed to the inability of the baseline BSS algorithm to identify three speakers when the window size is small. Future research work can be devoted to three speaker tracking.



(A)

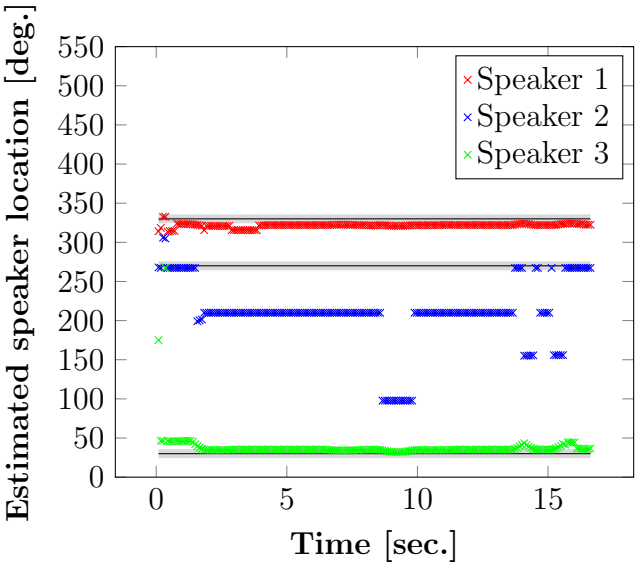


(B)

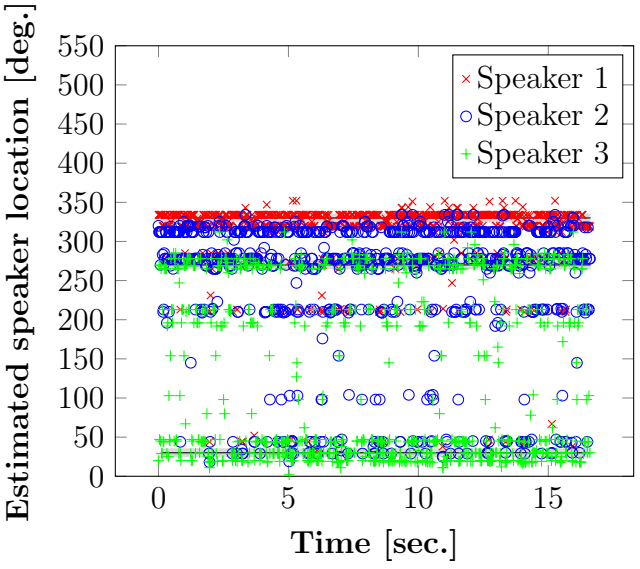


(C)

FIGURE 6.2: Motion tracking of two concurrent speakers synchronously moving counter clock-wise. The speakers started at $270° \pm 5°$ and $210° \pm 5°$ respectively and moved counter clock-wise by an angle of 30°. 6.2a plots the raw estimates found by the algorithm. The black tracks in 6.2b show the hand-drawn visible trend of the speaker movement. 6.2c plots the results of the SRP-PHAT algorithm when applied to the same audio file.



(A)



(B)

FIGURE 6.3: Position tracking of three concurrent speakers. The speakers were static in their place at $30° \pm 5°$, $270° \pm 5°$ and $330° \pm 5°$ shown by the black horizontal lines. The grey region is the $\pm 5°$ margin to account for the unintentional head movement during speech. Figure 6.3a are the results of the proposed algorithm while Figure 6.3b shows the results of SRP-PHAT algorithm on the same audio file. The proposed algorithm shows relatively stable estimates and is able to successfully track the position of two speakers. It, though, fails for the third one.

Appendix A

Probability Theory

Probability is common sense reduced to calculation — Laplace

A.1 Definitions

Sample Space

Every probabilistic model involves an *experiment*. The experiment can have several possible *outcomes*. The set of all possible outcomes of an event is called *Sample Space*.

$$\Omega = \{x_1, x_2, x_3, ..., x_n\} \tag{A.1}$$

where, x_1, x_2, x_3 etc. are the events and Ω is the sample space.

Probability

The likelihood of the occurrence of an event is called probability. If every event in the sample space A.1 is equally likely, then, the probability of event x_1 is given by:

$$P(x_1) = \frac{1}{No.\ of\ elements\ in\ \Omega} \tag{A.2}$$

Conditional Probability and Bayes' Rule

It is the probability of an event based on partial information. The probability of an event A given that and event B has already taken place is calculated as:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{A.3}$$

Equation A.3 is also known as Bayes' rule.

Independence

Conditional probability is calculated using the partial information that a prior occurring event provides about another event. Two events are said to be independent if they provide no information about each other. In other words, the partial probability is equal to the conditional probability.

$$P(A|B) = P(A) \tag{A.4}$$

Conditional Independence

Given and event C, the events A and B are said to be conditionally independent, if the following equation is satisfied:

$$P(A \cap B|C) = P(A|B)P(B|C) \tag{A.5}$$

Random Variable

A random variable is a real-valued function of the outcome of an experiment. A *function* of a random variable defines another random variable. It is of two types:

1. *Discrete* random variable has its range either finite or countably infinite.
2. *Continuous* random variable can take uncountably infinite number of values.

Probability Mass Function (PMF)

For all values x of a random variable X, PMF collects all possible outcomes that gives rise to the condition $X = x$. It is denoted as $p_X(x)$.

$$p_X(x) = P(X = x) \tag{A.6}$$

PMF is always calculated for discrete random variables. It is also called as *Marginal PMF*.

Expectation

Expectation or the expected value of a random variable with PMF p_X is given by,

$$E[X] = \sum_x x p_X(x) \tag{A.7}$$

It is also defined as the *mean* of the random variable.

Variance

Variance of a random variable X is defined as the expected value of the random variable $(X - E[X])^2$.

$$\begin{aligned} var(X) &= E[(X - E[X])^2] && \text{(A.8)} \\ &= \sum_x (x - E[X])^2 p_X(x) && \text{(A.9)} \end{aligned}$$

Joint PMF

Joint PMF of two random variables X and Y that are associated with the same experiment is defined as the joint probability of values that the two random variables can take. It is defined as:

$$p_{X,Y}(x, y) = P(\{X = x, Y = y\}) \tag{A.10}$$

Marginal PMF of a random variable can be calculated from the joint PMF by summing the joint PMF over all values of all other random variables. Mathematically,

$$p_X(x) = \sum_y p_{X,Y}(x, y) \tag{A.11}$$

Similarly,

$$p_Y(y) = \sum_x p_{X,Y}(x, y) \tag{A.12}$$

Conditional PMF

The conditional PMF of a random variable X conditioned on an event A is defined as:

$$p_{X|A}(x) = P(X = x|A) = \frac{P(X = x \cap A)}{P(A)} \tag{A.13}$$

Entropy

Entropy of a random variable is defined as:

$$H(X) = -\sum_{i=1}^{n} p_i \; log(p_i) \tag{A.14}$$

Mutual Information

Mutual information between m random variables y_i, where $i = 1 \ldots m$, is defined as:

$$I(y_1, y_2, \ldots, y_m) = \sum_{i=1}^{m} H(y_i) - H(y) \tag{A.15}$$

Probability Density Function (PDF)

For a *continuous* random variable X, the PDF is defined as:

$$P(X \in B) = \int_B f_X(x) \; dx \tag{A.16}$$

for every subset B of the real line. The normalization property of the PDF is defined as:

$$\int_{\infty}^{\infty} f_X(x) \; dx = P(-\infty < X < \infty) = 1 \tag{A.17}$$

Gaussian Random Variable

A *continuous* random variable is said to be **Gaussian** or **Normal**, if its PDF is defined by the relationship:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \; e^{-(x-\mu)^2/2\sigma^2} \tag{A.18}$$

where,

$$\begin{aligned} \sigma^2 &= \text{Variance of random variable } x. \\ \mu &= \text{Mean of random variable } x. \end{aligned}$$

Joint PDF

The joint PDF of two random variables X and Y for every subset B of the 2D plane is defined as:

$$P((X,Y) \in B) = \iint_{(x,y)\in B} f_{X,Y}(x,y)\ dx\ dy \tag{A.19}$$

The marginal PDF's can be calculated as:

$$f_X(x) = \int_{-\infty}^{+\infty} f_{X,Y}(x,y)\ dy \tag{A.20}$$

Similarly,

$$f_Y(y) = \int_{-\infty}^{+\infty} f_{X,Y}(x,y)\ dx \tag{A.21}$$

Covariance

The covariance of two random variables X and Y, denoted by $cov(X,Y)$ is defined by:

$$cov(X,Y) = E\Big[(X - E[X])(Y - E[Y])\Big] \tag{A.22}$$

Alternatively, it can also be expressed as,

$$cov(X,Y) = E[XY]\ -\ E[X]E[Y] \tag{A.23}$$

Correlation

When $cov(X,Y)$ of the two random variables X and Y is zero, the variables are said to be **uncorrelated**.

Independent random variables are also uncorrelated.

The correlation coefficient is defined as:

$$\rho(X,Y) = \frac{cov(X,Y)}{\sqrt{var(X)\ var(Y)}} \tag{A.24}$$

Kurtosis

The classical measure of non-gaussianity is kurtosis or the fourth-order cumulant. The kurtosis of y is classically defined by:

$$kurt(Y) = E[y^4] - 3(E[y^2])^2 \tag{A.25}$$

For Gaussian random variables, the kurtosis is zero.

References

[1] Jinyu Han, Zafar Rafii, and Bryan Pardo. *Audio Source Separation*. http://music.cs.northwestern.edu/research.php.

[2] Aapo Hyvärinen and Erkki Oja. Independent component analysis: A tutorial. 1999.

[3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[4] John Walter McDonough Jr, Volker Sebastian Leutnant, Sri Venkata Surya Siva Rama Krishna, Spyridon Matsoukas, et al. Determining speaker direction using a spherical microphone array, January 31 2017. US Patent 9,560,441.

[5] Riccardo Levorato and Enrico Pagello. Probabilistic 2d acoustic source localization using direction of arrivals in robot sensor networks. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pages 474–485. Springer, 2014.

[6] Huiqin Yan and H Howard Fan. Signal-selective doa tracking for wideband cyclostationary sources. *IEEE Transactions on Signal Processing*, 55(5), 2007.

[7] N Strobel, S Spors, and R Rabenstein. Joint audio-video object localization and tracking. *IEEE signal processing magazine*, 18(1):22–31, 2001.

[8] Yan-Chen Lu and Martin Cooke. Motion strategies for binaural localisation of speech sources in azimuth and distance by artificial listeners. *Speech Communication*, 53(5):622–642, 2011.

[9] R Venkatesan and A Balaji Ganesh. Full sound source localization of binaural signals. In *Wireless Communications, Signal Processing and Networking (WiSPNET), 2017 International Conference on*, pages 213–217. IEEE, 2017.

[10] Tobias May, Steven van de Par, and Armin Kohlrausch. A binaural scene analyzer for joint localization and recognition of speakers in the presence of interfering noise sources and reverberation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(7):2016–2030, 2012.

[11] Shoji Makino, Te-Won Lee, and Hiroshi Sawada. *Blind speech separation*, volume 615. Springer, 2007.

[12] Lang Tong, Yujiro Inouye, and R-W Liu. Waveform-preserving blind estimation of multiple independent sources. *IEEE Transactions on Signal Processing*, 41(7):2461–2470, 1993.

[13] Ivan Marković and Ivan Petrović. Speaker localization and tracking with a microphone array on a mobile robot using von mises distribution and particle filtering. *Robotics and Autonomous Systems*, 58(11):1185–1196, 2010.

[14] Ming Jian, Alex C Kot, and MH Er. Doa estimation of speech source with microphone arrays. In *ISCAS'98. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (Cat. No. 98CH36187)*, volume 5, pages 293–296. IEEE, 1998.

[15] Andreas Brendel, Sharon Gannot, and Walter Kellermann. Localization of multiple simultaneously active speakers in an acoustic sensor network. In *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 450–454. IEEE, 2018.

[16] Scott Rickard. Sparse sources are separated sources. In *2006 14th European Signal Processing Conference*, pages 1–5. IEEE, 2006.

[17] JH DiBiase, HF Silverman, and MS Brandstein. Microphone arrays: signal processing techniques and applications. In *chapter Robust localization in reverberant rooms*, pages 157–180. Springer Verlag, 2001.

[18] Luiz CF Nogueira and Mariane R Petraglia. Robust localization of multiple sound sources based on bss algorithms. In *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*, pages 579–583. IEEE, 2015.

[19] Lin Wang, Joshua D Reiss, and Andrea Cavallaro. Over-determined source separation and localization using distributed microphones. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(9):1573–1588, 2016.

[20] Andreas Brendel and Walter Kellermann. Localization of multiple simultaneously active sources in acoustic sensor networks using adp. In *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 1–5. IEEE, 2017.

[21] Ofer Schwartz and Sharon Gannot. Speaker tracking using recursive em algorithms. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(2):392–402, 2013.

[22] Despoina Pavlidi, Anthony Griffin, Matthieu Puigt, and Athanasios Mouchtaris. Real-time multiple sound source localization and counting using a circular microphone array. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(10):2193–2206, 2013.

[23] H. Buchner, R. Aichner, and W. Kellermann. Trinicon: a versatile framework for multichannel blind signal processing. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii–889–92 vol.3, May 2004.

[24] scikit-learn. http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/modules/clustering.html.

[25] Ui-Hyun Kim, Kazuhiro Nakadai, and Hiroshi G Okuno. Improved sound source localization and front-back disambiguation for humanoid robots with two ears. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 282–291. Springer, 2013.

[26] Christine Evers Yuval Dorfan, Sharon Gannot, and Patrick A Naylor. Speaker localization with moving microphone arrays. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 1003–1007. IEEE, 2016.

[27] Christine Evers, Yuval Dorfan, Sharon Gannot, and Patrick A Naylor. Source tracking using moving microphone arrays for robot audition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6145–6149. IEEE, 2017.