DISTANT SPEECH RECOGNITION USING MICROPHONE ARRAYS

M.Tech. Dissertation

Final Stage

George Jose
153070011

Supervised By

Prof. Preeti Rao

Department of Electrical Engineering
Indian Institute of Technology, Bombay
Powai, Mumbai - 400 076.

2016-2017

Abstract

Speech is the most natural mode of communication and distant speech recognition enables us to communicate conveniently with other devices without any body or head mounted microphones. But the real world deployment of such systems comes with a lot of challenges. This work seeks to address the two major challenges in such a system namely noise and reverberation by using microphone arrays. In this regard, beamforming is most commonly used technique in multichannel signal processing to reduce noise and reverberation. A detailed analysis of the existing source localization and Automatic Speech Recognition (ASR) performances of beamforming techniques were performed using the Chime Challenge dataset. Based on these studies an improved steering vector was proposed to increase the performance of Mininimum Variance Distortionless Response (MVDR) beamformer in real data. The proposed model helped to reduce the Word Error Rate (WER) of MVDR beamformer from 17.12% to 12.75% in real data using an ASR system based on GMM-HMM acoustic model and trigram language model. Finally the WER was reduced to 5.52 % using a DNN-HMM acoustic model and lattice rescoring using RNN language model.

Contents

Chapter 1

INTRODUCTION

First speech recognition system was Audrey designed by Bell Labs in 1952 which could recognize digits from a single voice. From there speech recognition systems began to evolve continuously with vocabulary size increasing from vowels to digits and finally to words. The focus then shifted on to speaker independent connected word recognition and finally to large vocabulary continuous speech recognition. Speech recognition technologies have then entered the marketplace, benefiting the users in a variety of ways.

The integration of voice technology into Internet of Things (IoTs) has led to development of plethora of real world applications ranging from Smart Homes, voice controlled personal assistants like Apple's Siri or Amazon's Alexa, humanoid robots etc where the user can be few metres away from the device. Deployment of speech recognition systems into the real world also comes with a lot of challenges like contending with noise, reverberation and overlapping speakers. For example, an automobile speech recognition system must be robust to noise but only low reverberation [1]. On the other hand, a meeting room environment and home environment typically has a much higher SNR but has moderate to high amounts of reverberation and the additional challenge of overlapping talkers [2, 3]. Mobile devices can be used in highly variable environments. So distant speech recognition is a highly challenging problem. This work tries to overcome the two main challenges i.e. noise and reverberation commonly occurring in enclosed scenarios like home and meeting environments using an array of microphones.

1.1 Array Processing

Speech recognition techniques using a single channel microphone produced poor recognition rates when the speaker was distant from the microphone (say more than 50cm). Clearly the single channel techniques were not able to deal effectively with low SNR and high reverberant scenarios. This led to the use of multiple microphones known as microphone arrays.

For the distant speech recognition task, the microphone arrays offer several advantages over a single channel. First a microphone array can locate and then track the speaker positions which will be useful in meetings and teleconferencing to steer the camera towards the active speaker [3]. This is achieved on the basis that the signals coming from different locations reach the microphones with different delays. Exploiting this helps finding the location of the speaker. Secondly, multiple microphones can be used for source separation tasks where two speakers are talking simultaneously and we need to separate each speaker. This is difficult in the frequency domain since the frequency content of both speakers overlap each other. Microphone arrays help exploit the separation in the spatial domain, when the signals come from different directions. This process of steering the response of microphone array towards a required direction while attenuating the signals coming from other directions is known as spatial filtering or beamforming.

This concept of array signal processing began as early as 1970s, where it was employed in antennas, sound detection and ranging(sonars) and radio detection and ranging(radars) for localizing and processing narrowband signals. For example, radars and sonars are used to detect and localize moving targets like air-crafts, missiles, and ships. In antennas, array processing is used for directional reception as well as transmission of narrowband signals. So much of the theory behind the construction of spatial filters were derived from these narrowband processing techniques. Since speech is a wideband signal, most of the array processing algorithm works by considering each frequency bin as a narrow band signal and applying the narrowband algorithms to each bin.

1.2 System Overview

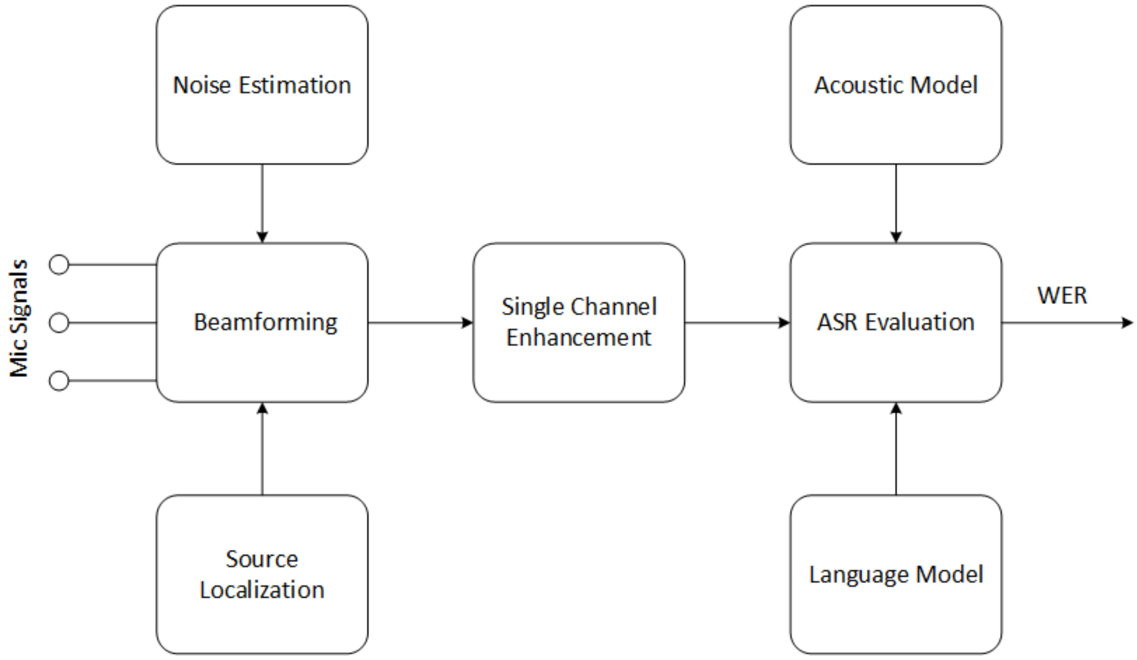


Figure 1.1: Block Diagram

The main components of our distant speech recognition system are shown in Fig 1.1 comprising of front end enhancement stage followed by a speech recognition system which takes microphone array signals as input and gives the recognition accuracy in terms of word error rate (WER). Following is a brief description of the different stages involved:

Source Localization : The process of finding the direction of the speaker using the information from the signals received at the microphone arrays. Various source localization algorithms are described in detail in Chapter 2

Beamforming : The process of steering the response of the microphone array towards the source direction thereby attenuating the undesired signals from other direction. The working of different beamforming techniques is explained in Chapter 3.

Noise Estimation : Most beamforming techniques are posed as a constrained optimization problem of minimizing the noise power at the output. For this an estimate of the correlation between the noise across channels is required. Some beamforming techniques works based on assumptions regarding the noise fields and donot estimate the noise.

Different noise field models and techniques for noise estimation are covered in Chapter 3

Single Channel Enhancement : After performing beamforming, single channel enhancement like Wiener filtering for noise reduction [4] or dereverberation algorithms like Non negative Matrix Factorization (NMF) [5] are performed to further enhance the signal.

ASR : Speech recognition engine generates a hypothesis regarding what the speaker said from the enhanced acoustic waveform with help of a trained acoustic and language models. These hypotheses are compared with reference text to compute accuracy in terms of WER. Chapter 7 presents the speech recognition accuracies of various beamforming methods under different conditions.

Chapter 2

Acoustic Source Localization

For the purpose of beamforming, it is necessary to estimate the location of speaker to apply spatial filtering techniques. This problem of finding the source location using sensor arrays has long been of great research interest given its practical importance in a great variety of applications, e.g.,radio detection and ranging (radar), underwater sound detection and ranging (sonar), and seismology. In these applications, source localization is more commonly referred to as direction of arrival (DOA) estimation. Following sections will discuss various source localization algorithms.

2.1 Classification of Source Localization Algorithms

. The various source localization algorithms can be broadly categorized into three as follows:

1. High resolution spectral based algorithms : These methods are based on the eigen decomposition of spatial correlation matrix between signals arriving at the microphones. Most often spatial correlation matrix is not known apriori and are estimated by taking the time averages from the observed data. These methods assume source signal to be narrowband, stationary and in the far field region of microphones . These algorithms are derived from high resolution spectral analysis based techniques. A major drawback is the associated computational complexity.

MUSIC (Multiple Signal Classification) [6, 7] and ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques) [8] are the two main algorithms under this category

2. Steered Response Power (SRP) based algorithms : These techniques involve evaluating a function at different hypothesis locations and then using a search algorithm to find the direction where the function attains it's maximum value [3, 9]. Typically the response of a beamformer is steered towards the hypothesis locations and the function which is evaluated is the received power at each direction [10]. When the source is in far field, in order to obtain higher spatial resolution, beam needs to be steered in a large range of discrete angles which increases the computational complexity leading to a poor response time.

3. Time Delay Of Arrival (TDOA) based algorithms : These are the simplest class of algorithms which involves estimating the time delay of arrival of the speech signal between a pair of microphones and then subsequently using this information to find the direction of source. The peaks in the cross correlation function between the signals is exploited to find the TDOA. Generalized Cross Correlation based methods which uses additional weighting functions to cross correlation are the commonly used algorithms in this category [11].

2.2 TDOA based algorithms

TDOA based algorithms are the most commonly used techniques for source localization due to it's computational simplicity, robustness, and lack of prior knowledge about microphone positions. These class of algorithms uses a reference channel and tries to find the relative delay of arrival of all the other channels with respect to this reference microphone. Following sections will explain different TDOA based methods in detail.

2.2.1 Cross Correlation (CC)

The simplest approach is to find the time shift where peaks appear in the cross correlation of signals between two channels. Let x$_1$(t) and x$_2$(t) be the signals received at two different channels, then the cross correlation can be expressed as :

$$R_{x_1x_2}(\tau) = E\{x_1(t)x_2(t-\tau)\} \tag{2.1}$$

where E{} is the expectation operator. TDOA is calculated as the time shift for which cross correlation is maximum.

$$D = \arg\max_{\tau}\ R_{x_1x_2}(\tau) \tag{2.2}$$

The working of above algorithm could be better explained using a delay only signal model in the presence of spatially uncorrelated noise. Let the speech signal received at each microphone be a delayed, attenuated version of the original speech signal with additive noise. So the signal received at the microphone can be represented as:

$$x(t) = \alpha s(t-\tau) + n(t) \tag{2.3}$$

Using the above signal model and with the assumption that noise and speech signal are uncorrelated,and also noise between the channels are uncorrelated ($R_{n_1n_2}(\tau)$=0) the cross correlation between the signals at microphones can be simplified as follows:

$$\begin{aligned} R_{x_1x_2}(\tau) &= E\{(\alpha_1 s(t-\tau_1) + n_1(t))(\alpha_2 s(t-\tau_2-\tau) + n_2(t-\tau))\} \\ &= \alpha_1\alpha_2 R_{ss}(\tau-(\tau_1-\tau_2)) + R_{n_1n_2}(\tau) \\ &= \alpha_1\alpha_2 R_{ss}(\tau-D) = \alpha_1\alpha_2 R_{ss}(\tau) * \delta(\tau-D) \end{aligned}$$

Cross correlation between the channels is the auto correlation of the speech signal convolved by an shifted impulse function. Since $R_{ss}(0) \geq R_{ss}(\tau)$, $R_{x_1x_2}(\tau)$ will have a peak at D.

The cross correlation is computed using cross power spectrum which are related ($G_{x_1x_2}(f)$) by the inverse Fourier Transform relationship :

$$R_{x_1x_2}(\tau) = \int G_{x_1x_2}(f)e^{j2\pi f\tau}df \tag{2.4}$$

Here also we get only an estimate of the cross power spectrum $G_{x_1x_2}(f)$. One common method of cross power spectrum estimation is by using Welch periodogram method [12].

2.2.2 Generalized Cross Correlation (GCC)

Cross correlation algorithms fail in the presence of reverberation when there are early reflections which are coming from different directions. Cross correlation between the signals in this case can be expressed as:

$$R_{x_1x_2}(\tau) = R_{ss}(\tau) * \sum \alpha_i \delta(\tau - D_i) \tag{2.5}$$

where impulses are due to the early reflections. Now the cross correlation function will contain scaled and shifted versions of $R_{ss}(\tau)$ corresponding to each impulse. Since $R_{ss}(\tau)$ is a smoothly decaying function, these shifted versions can overlap and produce new peaks leading to erroneous results.

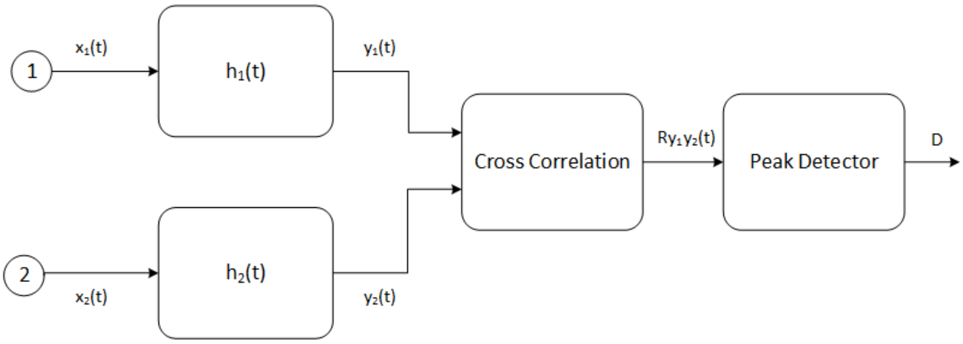


Figure 2.1: GCC Framework

GCC based algorithms were introduced to increase the robustness of CC method towards noise and reverberation by applying an additional weighing factor to each bin. Fig 2.1 shows the block diagram of GCC based algorithms based on which the generalized cross correlation function $R_{y_1y_2}(\tau)$ can be expressed as:

$$\begin{aligned}R_{y_1y_2}(\tau) &= \int G_{y_1y_2}(f)e^{j2\pi f\tau}df \\ &= \int H_1(f)H_2^*(f)G_{x_1x_2}(f)e^{j2\pi f\tau}df \\ &= \int \psi(f)G_{x_1x_2}(f)e^{j2\pi f\tau}df\end{aligned}$$

Here $\psi(f)$ represents the weighing factor applied to each frequency bin. The TDOA is estimated as the time shift for which the generalized cross correlation function attains maximum value.

$$D = \arg\max_{\tau}\ R_{y_1y_2}(\tau)$$

Following are the different weighing functions used [11]:

Roth Processor

The GCC function with Roth weighing factor is given by:

$$R_{y_1y_2}(\tau) = \int_{\tau} \frac{G_{y_1y_2}(f)}{G_{x_1x_1}(f)}e^{j2\pi f\tau}df \tag{2.6}$$

The working of the weighing function can understood by expanding cross power spectrum as follows:

$$\begin{aligned}R_{y_1y_2}(\tau) &= \int_{\tau} \frac{G_{ss}(f)e^{-j2\pi fD}}{G_{ss}(f)+G_{n_1n_1}(f)}e^{j2\pi f\tau}df \\ &= \delta(\tau - D) * \int_{\tau} \frac{G_{ss}(f)}{G_{ss}(f)+G_{n_1n_1}(f)}e^{j2\pi f\tau}df\end{aligned} \tag{2.7}$$

So it suppresses those frequency bins where SNR is lower.

Smoothed Coherence Transform (SCOT)

The GCC function with SCOT weighing factor is given by:

$$R_{y_1y_2}(\tau) = \int_{\tau} \frac{G_{x_1x_2}(f)}{\sqrt{G_{x_1x_1}(f)G_{x_2x_2}(f)}}e^{j2\pi f\tau}df \tag{2.8}$$

While Roth considers SNR of only one channel, SCOT considers SNR of both the channels. It also gives a sharper peak in the generalized cross correlation function.

GCC Phase Transform (GCC PHAT)

The GCC function with PHAT weighing factor is given by:

$$R_{y_1y_2}(\tau) = \int_{\tau} \frac{G_{y_1y_2}(f)}{|G_{x_1x_2}(f)|} e^{j2\pi f\tau} df \tag{2.9}$$

GCC PHAT uses only phase information by whitening the cross power spectrum and gives equal weightage to all the bins. GCC PHAT exhibits sharp peaks in the generalized cross correlation function and hence it works better in moderate reverberant conditions. Under low SNR and high reverberant conditions, the performance will degrade.

Hannan Thompson (HT)

The HT function is given by:

$$\psi(f) = \frac{1}{|G_{x_1x_2}(f)|} \frac{|\Gamma|^2(f)}{1 - |\Gamma|^2(f)} \tag{2.10}$$

Here $\Gamma(f)$ represents the coherence function given by :

$$\Gamma(f) = \frac{G_{x_1x_2}(f)}{\sqrt{G_{x_1x_1}(f)G_{x_2x_2}(f)}} \tag{2.11}$$

HT method adds an additional weighing factor based on the coherence between the channels to the GCC PHAT algorithm. Higher the coherence, more weightage will be given to that particular frequency bin.

2.3 Steered Response Power Phase Transform (SRP PHAT)

The TDOA based methods considers only a microphone pair at a time and does not make use of knowledge about microphone positions. SRP based algorithm tries to overcome these limitations at the cost of increased computational complexity. SRP PHAT in particular tries to combine the robustness of GCC PHAT with the above mentioned advantages of SRP based algorithms.

From the knowledge of array geometry, a set of TDOAs can be computed for each direction. Suppose an angular resolution of 1^o is required in the azimuth plane, then with respect to reference microphone a set of TDOAs can be computed for all the other microphones at each required angle. At each hypothesis location, the SRP PHAT function is computed by evaluating GCC PHAT for the estimated TDOA between each microphone pair and then summing over all the microphone pairs [3]. Suppose there are N microphones, then the SRP PHAT function at each hypothesis location θ can be evaluated as follows:

$$f(\theta) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} IFT\{\frac{G_{x_i x_j}(f)}{|G_{x_i x_j}(f)|} e^{-j2\pi f \tau_{ij}(\theta)}\} \tag{2.12}$$

Here $\tau_{ij}(\theta)$ represents the estimated TDOA between microphone pair i & j when the source is at an angle θ. [10] uses a non linear function of GCC PHAT function based on hyperbolic tangent to emphasize larger values. Finally the source direction can be computed as the θ which maximizes the above function.

2.4 Postprocessing

Most of the above mentioned algorithms works on the assumption that the noise across the microphones are weakly correlated. In the presence of strong directional noises like door shut, a strong peak will be present corresponding to this event leading to wrong results. To account for this some postprocessing is performed on estimated TDOAs.

One approach is to assume that these noises are present only for a shorter duration and perform some continuity filter in the time domain. BeamformIt [2] an open source software based on C++, uses Viterbi decoding to find the best path across time from a set of N-best TDOAs at each time frame. Here N-best TDOAs at each time frame are chosen by the taking time shifts corresponding to N highest peaks in the GCC PHAT.The transition probabilities between two points are defined based on the differences in the TDOA between that points and the emission probability is computed based on the logarithm of the GCC PHAT values. Now using these, Viterbi algorithm can be performed to find the best path. In the case of overlapping speaker scenario, the TDOAs corresponding to each speaker can be estimated by selecting 2-best paths across time.

2.5 Summary

This chapter gave a review of working of the different source localisation algorithms with the main focus given to TDOA based algorithms. In next chapter different spatial filtering techniques will be discussed

Chapter 3

Acoustic Beamforming

Beamforming is a popular technique used in antennas, radars and sonars for directional signal transmission or reception. Consider a scenario where two speakers are speaking simultaneously and we want to perform source separation. Clearly this could not be achieved in the time-frequency domain since the frequency components overlap each other. One possible solution is to exploit the spatial separation between the speakers. In this chapter, first a simple and intuitive way of how microphone arrays help in achieving spatial separation is explained and finally a more formal explanation from optimization view point is provided.

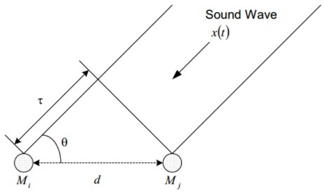


Figure 3.1: Linear Microphone Array [13]

Consider the scenario of M microphones separated by a distance 'd' with the source located at an angle θ with respect to the axis of a linear microphone array as shown in Fig 3.1 . The time delay of arrival at the ith with respect to first microphone is given by:

$$\tau_{1i} = (i-1)\frac{dcos\theta}{c}$$

So the signal received at each microphone is a delayed version of the original signal, with the delay depending on the source direction. Let x$_i$(t) be the signal received at the ith microphone and s(t) be the original speech signal, then x$_i$(t) can be expressed as:

$$x_i(t) = s(t - (i-1)\frac{dcos\theta}{c})$$

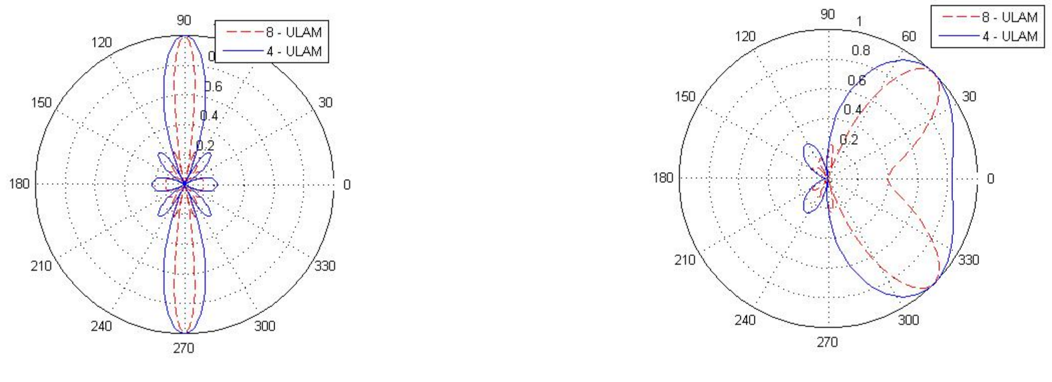
On simply averaging the signals received at different microphones,

$$y(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t) = \frac{1}{N}\sum_{i=1}^{N} s(t - (i-1)\frac{dcos\theta}{c})$$

Taking the discrete Fourier Transform gives,

$$Y(\omega) = S(\omega)\frac{1}{N}\sum_{i=1}^{N} e^{-j\omega((i-1)\frac{dcos\theta}{c})} = S(\omega)H(\omega)$$

Here H(ω) represents the response of the array to the speech signal. The frequency response is dependant on N, d, ω and θ. Plotting the magnitude response keeping N,d and ω fixed on polar coordinates gives the directivity pattern or beam pattern [14].



(a)

(b)

Figure 3.2: Beampattern for Uniform Linear Array Microphone(ULAM) with simple averaging (left) and after steering towards an angle of 45^o at 2000Hz frequency(right). Dotted line represents the magnitude response of 8-channel ULAM and solid line represents the magnitude response of 4-channel ULAM

Fig 3.2 (a) shows the beampattern pointing towards 0^o. But we need the beam to point towards the source direction, say 45^o. So instead of simply averaging, we need to first compensate for delays and then average across channels. Let θ_s be the estimated source direction. Then the output signal can be expressed as :

$$y(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t + (i-1)\frac{dcos\theta_s}{c})$$

Now the impulse response of the microphone array towards the speech signal will be :

$$H(\omega, \theta) = \frac{1}{N}\sum_{i=1}^{N} e^{-j\omega(i-1)(\frac{d(cos\theta - cos\theta_s)}{c})}$$

Fig 3.2 (b) shows the beampattern now steered towards an angle of 45^o. It can observed that the response of array is maximum in the direction of source while attenuating the signals from other directions. This principle of the algorithmically steering the beam towards the direction of source is known as beamforming.

3.1 Array Model

This section will give a more formal introduction about different beamforming techniques and regarding the different signal models. Let y$_i$(t) be the signal received at the ith microphone which is a delayed version of the speech signal s(t) in the presence of additive noise n$_i(t)$. So the received signal, y$_i$(t) $=$ s(t-τ_{1i}) $+$ n$_i(t)$, where τ_{1i} is the TDOA with respect to the first microphone. Without any loss of generality, the first microphone is selected as the reference microphone. Signal could be represented in the frequency domain as :

$$y_i(f) = x_i(f) + n_i(f) = s(f)e^{-j2\pi f\tau_{1i}} + n_i(f) \tag{3.1}$$

In vector notation, the received signal can be represented as:

$$\mathbf{y}(f) = \mathbf{x}(f) + \mathbf{n}(f) = \mathbf{d}(f)s(f) + \mathbf{n}(f) \tag{3.2}$$

where, $\mathbf{d}(f) = [1\ e^{-j\omega\tau_{12}}\ e^{-j\omega\tau_{13}}\ \ldots\ e^{-j\omega\tau_{1N}}]^T$ is known as the steering vector which is calculated at source localization stage from the TDOA estimates.

Originally proposed for narrowband signals, beamformer applies a filter to each channel and then sums the output of all the filters as shown in Fig 3.3. Filters are designed based on the optimization criterion and assumptions regarding noise. For a wideband signal like speech, each frequency bin is approximated as a narrow band signal and a set of filters is designed to each bin independently.

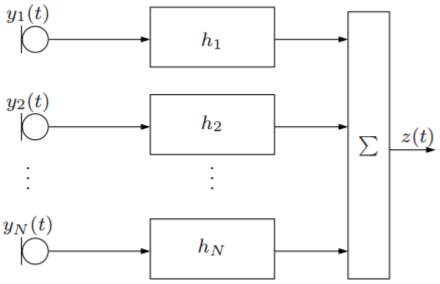


Figure 3.3: Beamformer Model [15]

Let **h**(f) be a coloumn vector with each element representing the transfer function of filter at the output of each microphone. Then the output at the beamformer is given by:

$$\begin{aligned} z(f) &= \mathbf{h}^H(f)\mathbf{y}(f) \\ &= \mathbf{h}^H(f)\mathbf{d}(f)s(f) + \mathbf{h}^H(f)\mathbf{n}(f) \end{aligned} \tag{3.3}$$

For no signal distortion at the beamformer output, $h^H(f)d(f)$ should be equal to one which is referred as the signal distortionless constraint. The power at the output of the beamformer is given by :

$$\begin{aligned} P = E[z(f)z^H(f)] &= E\{(\mathbf{h}^H(f)(\mathbf{x}(f)+\mathbf{n}(f))(\mathbf{h}^H(f)(\mathbf{x}(f)+\mathbf{n}(f))^H\} \\ &= \mathbf{h}^H(f)E\{\mathbf{x(f)x(f)}^H\}\mathbf{h}(f) + \mathbf{h}^H(f)E\{\mathbf{n}(f)\mathbf{n}(f)^H\}\mathbf{h}(f) \\ &= \mathbf{h}^H(f)\mathbf{R}_x(f)\mathbf{h}(f) + \mathbf{h}^H(f)\mathbf{R}_n(f)\mathbf{h}(f) \\ &= \sigma_s(f)|\mathbf{h}^H(f)\mathbf{d}(f)|^2 + \mathbf{h}^H(f)\mathbf{R}_n(f)\mathbf{h}(f) \end{aligned}$$

where σ_s is PSD of the speech signal, and $\mathbf{R}_n(f)$ is the spatial coherence matrix of the noise field. Let $\mathbf{R}_n(f) = \sigma_n(f)\mathbf{\Gamma}_n(f)$, where $\mathbf{\Gamma}_n(f)$ is known as the pseudo coherence matrix [15] and $\sigma_n(f)$ is the average PSD of the noise at input.

3.2 Noise Coherence Matrix Estimation

Many techniques based on single channel enhancement techniques require an estimate of PSD. In the case of multichannel algorithms, apart from PSD of the microphones an estimate of the cross power spectral densities between the microphones is also required which is represented in the form of a matrix known as noise coherence matrix ($\mathbf{R}_n(f)$). The main diagonal elements contain the PSD of each microphone while the cross terms represent the cross PSDs. It captures the information regarding noise field which encodes the spatial information regarding the noise sources. Hence an accurate estimation of noise coherence matrix is required to effectively suppress the noise sources.

A typical way to estimate the noise coherence matrix is to identify the regions where only noise exists and then perform an ensemble averaging. Some methods rely on the assumption that in initial part of the signal speech is absent and estimate noise from this region. Another popular method is to use a Voice Activity Detector (VAD) [16, 17] to find the regions where speech is absent and estimate noise coherence matrix using these frames. But VAD based methods donot perform updating of noise coherence matrix when speech is present which poses a problem in non stationary noise scenarios where noise statistics are changing.

One approach is to exploit the sparsity of the speech in the time frequency domain. Instead of performing voice activity detection at frame level, those time-frequency bins which contain only noise are identified and noise coherence matrix is updated. A spectral mask is used to estimate the posterior probability of each time frequency bin belonging to the noise class and then a weighted averaging is performed based on these posterior probabilities to estimate the coherence matrix. Yoshioka et. al. [18] uses a complex Gaussian Mixture Model (CGMM) [19] to estimate the spectral while Heymann et. al. [20] uses a bidirectional Long Short Term Memory (BLSTM) network to estimate the spectral masks.

Some beamforming techniques instead of estimating the noise fields, uses noise coherence matrix models based on some ideal assumptions regarding the noise fields. Two types of model are the diffuse field noise model and the spatially white noise model. Diffuse field model in turn can be classified into spherically isotropic [21] and cylindrical isotropic model [22]. Spherically isotropic model assumes the noise signal propagates as plane waves in all directions with equal power in the three dimensional space while cylindrically isotropic model assumes noise propagates only along two dimensions in the horizontal directions. Spatially white noise model assumes that noise signals across the channels are uncorrelated. A common property of above noise models is that every element in the noise coherence matrix is real.

3.3 Performance Metrics

Based on the above signal models following are some of the narrowband performance metrics that could be used to evaluate the performance of beamforming [15]:
iSNR - Defined as the ratio of the average desired signal power to average noise power at the input

$$iSNR(f) = \frac{\sigma_s(f)}{\sigma_n(f)}$$

oSNR - Defined as the ratio of the desired signal power to residual noise power at the output of beamformer

$$\begin{aligned} oSNR(h(f)) &= \frac{\sigma_s(f)|\mathbf{h}^H(f)\mathbf{d}(f)|^2}{\sigma_n(f)\mathbf{h}^H(f)\mathbf{\Gamma}_n(f)\mathbf{h}(f)} \\ &= \frac{|\mathbf{h}^H(f)\mathbf{d}(f)|^2}{\mathbf{h}^H(f)\mathbf{\Gamma}_n(f)\mathbf{h}(f)} iSNR(h(f)) \end{aligned}$$

Array Gain - Defined as the ratio of the output SNR (oSNR) to input SNR (iSNR).

$$A(h(f)) = \frac{oSNR}{iSNR} = \frac{|\mathbf{h}^H(f)\mathbf{d}(f)|^2}{\mathbf{h}^H(f)\mathbf{\Gamma}_n(f)\mathbf{h}(f)}$$

White Noise Gain - Defined as the array gain in a spatially white noise field. In a spatially white noise field, the noise present in the channels are uncorrelated with each other leading to pseudo coherence noise matrix being an identity matrix.

$$W(h(f)) = \frac{|\mathbf{h}^H(f)\mathbf{d}(f)|^2}{\mathbf{h}^H(f)\mathbf{h}(f)}$$

Directivity - Defined as the array gain in a spherically isotropic diffuse noise field. In a diffuse noise field, the sound pressure level is uniform at all points, with noise coming from all directions. Coherence between the channels decreases with increasing frequency as well as microphone distance.

$$D(h(f)) = \frac{|\mathbf{h}^H(f)\mathbf{d}(f)|^2}{\mathbf{h}^H(f)\mathbf{\Gamma}_{diff}(f)\mathbf{h}(f)}$$

where $\mathbf{\Gamma}_{diff}(f)$ represents the pseudo coherence matrix of the diffuse noise field whose elements are given by:

$$[\mathbf{\Gamma}_{diff}(f)]_{ij} = sinc(2fd_{ij}/c)$$

Here d_{ij} is the distance between the microphones i & j and c is the speed of sound.

Beampattern - Represents the response of the beamformer as a function of the direction of the source. It is defined as the ratio of the output power of the desired signal having a steering vector **d**(f) to the input power.

$$B(\mathbf{d}(f)) = |\mathbf{h}^H(f)\mathbf{d}(f)|^2$$

Noise Reduction Factor - Defined as the ratio of noise power at the input to the residual noise power at the output of beamformer gives an indication of how much noise power the beamformer is able to reject.

$$\xi nr(h(f)) = \frac{1}{\mathbf{h}^H(f)\mathbf{\Gamma}_n(f)\mathbf{h}(f)}$$

Desired Signal Cancellation Factor - Defined as the ratio of the average power of the desired signal at the input to the desired signal power at the output of beamformer.

$$\xi_{dsc}(h(f)) = \frac{1}{|\mathbf{h}^H(f)\mathbf{d}(f)|^2}$$

This can take a value of 1 corresponding to no distortion when $|\mathbf{h}^H(f)\mathbf{d}(f)| = 1$

3.4 Beamforming Techniques

This section builds on top of the previous two sections to discuss the different beamforming techniques proposed in literature. The optimization criteria and assumptions each technique make regarding noise is discussed in detail.

3.4.1 Maximum SNR Beamforming

As the name suggests, maximum SNR beamformer tries to maximize the SNR at the output of the beamformer for each frequency bin. SNR at the output of the beamformer can be expressed as:

$$oSNR(h(f)) = \frac{\mathbf{h}^H(f)\mathbf{R}_x(f)\mathbf{h}(f)}{\mathbf{h}^H(f)\mathbf{R}_n(f)\mathbf{h}(f)}$$

where $\mathrm{R}_x(\mathrm{f}) = \sigma_s\mathbf{d}(f)\mathbf{d}^H(f)$ is a rank-1 matrix if the speaker is assumed to be stationary. Here optimization criteria is to find the filter weights which maximizes the SNR at the output of the beamformer. Above problem is termed as Generalized Eigen Value problem based on which the optimization criteria can be rewritten as:

$$h_{SNR}(f) = \arg\max_{h(f)} \frac{\mathbf{h}^H(f)\mathbf{R}_n^{-1}(f)\mathbf{R}_x(f)\mathbf{h}(f)}{\mathbf{h}^H(f)\mathbf{h}(f)} \tag{3.4}$$

Solution to this will be the eigen vector corresponding to maximum eigen value of $\mathbf{R}_n^{-1}(f)\mathbf{R}_x(f)$. Since $\mathbf{R}_x(f)$ is a rank-1 matrix, the product of the matrices will be rank-1 and hence it will have only one non zero positive (Hermitian matrix) eigen value which will also be the maximum value. So the solution to the eigen value problem $\sigma_s\mathbf{R}_n^{-1}(f)\mathbf{d}(f)\mathbf{d}^H(f)h_{SNR}(f) = \lambda h_{SNR}(f)$, where λ represents the eigen value is obtained as :

$$h_{SNR}(f) = \alpha \mathbf{R}_n^{-1}(f)\mathbf{d}(f) \tag{3.5}$$

where α is an arbitrary scaling factor which doesnot influences the subband SNR but can introduce distortions to the speech signal. [23] discusses two types of normalization: Blind Analytic Normalization (BAN) and Blind Statistical Normalization (BSN) to control the speech distortions by applying a single channel postfiltering. This technique is also known as Generalized Eigen Value (GEV) beamforming since it solves the generalized eigen value problem [20].

3.4.2 Minimum Variance Distortionless Response Beamforming

MVDR beamformer minimizes the noise power at the beamformer output with the constraint that there is no speech distortion [15, 24]. As explained in section 3.3, the signal distortionless constraint is given by $|\mathbf{h}^H(f)\mathbf{d}(f)|$=1. MVDR filter is obtained by solving the constrained optimization problem:

$$\mathbf{h}_{MVDR}(f) = \arg\min_{\mathbf{h}(f)} \mathbf{h}^H(f)\boldsymbol{R}_n(f)\mathbf{h}(f) \quad \text{subject to } \mathbf{h}^H(f)\mathbf{d}(f) = 1 \tag{3.6}$$

$$\mathbf{h}_{MVDR}(f) = \frac{\boldsymbol{R}_n^{-1}(f)\mathbf{d}(f)}{\mathbf{d}^H(f)\boldsymbol{R}_n^{-1}(f)\mathbf{d}(f)} \tag{3.7}$$

The denomoinator $\mathbf{d}^H(f)\boldsymbol{R}_n^{-1}(f)\mathbf{d}(f)$ is a gain factor. So MVDR beamforming can be expressed as $\alpha\mathbf{R}_n^{-1}(f)\mathbf{d}(f)$, where α is fixed to ensure that there is no speech distortion. Hence it also maximizes the subband SNR. The beamwidth of the main lobe of MVDR beamforming is very less making it susceptible to signal cancellation issues in the presence of source localisation errors. The white noise gain of MVDR beamformers decreases with increasing $|h_{MVDR}(f)|^2$ (from section 3.3). So inorder to make the MVDR beamformers more robust to white noise and source localization errors, an additional constraint was imposed to limit the norm of the weights. Solving the optimization problem in Eq 3.6 using both the above constraints we get Minimum Variance Distortionless Response

Diagonal Loading (MVDR DL) beamformer

$$\mathbf{h}_{MVDRDL}(f) = \frac{(\boldsymbol{R_n(f)} + \boldsymbol{\epsilon I})^{-1}\mathbf{d}(f)}{\mathbf{d}^H(f)(\boldsymbol{R_n(f)} + \boldsymbol{\epsilon I})^{-1}\mathbf{d}(f)} \tag{3.8}$$

3.4.3 Delay Sum Beamforming

Delay and Sum beamforming (DSB) solves the constrained optimization problem of maximizing the white noise gain at the output of the beamformer subject to signal distortionless constraint. The DSB filter is obtained as follows:

$$\mathbf{h}_{DSB}(f) = \arg\min_{\mathbf{h}(f)} \frac{|\mathbf{h}^H(f)\mathbf{d}(f)|^2}{\mathbf{h}^H(f)\mathbf{h}(f)} \quad \text{subject to } \mathbf{h}^H(f)\mathbf{d}(f) = 1 \tag{3.9}$$

$$\mathbf{h}_{DSB}(f) = \frac{\mathbf{d}(f)}{\mathbf{d}^H(f)\mathbf{d}(f)} = \frac{\mathbf{d}(f)}{N} \tag{3.10}$$

As the name suggests, it just compensates for the delay at each channel and adds them. This is same as the beamformer discussed in the beginning of this chapter. DSB is a data independent beamformer since the filter weights doesnot depend on the data received at the input. DSB beamformers have a narrow main lobe width in the beampattern at higher frequencies but wider width at lower frequencies which limits the ability to attenuate noise from other directions. Stolbov et. al. [25] proposes a modification by multpyling with an additional complex gain to each filter to account for fluctuations in microphone sensitivity and phase. This method referred to as Multi Channel Alignment (MCA) beamforming helps reduce the width of the main lobe and also reduces sidelobe levels.

3.4.4 Super Directive Beamforming

Super Directive beamforming (SDB) solves the optimization criteria of maximizing the directivity (see section 3.3) at the output of the beamformer subject to the distortionless constraint [26]. The SDB filter could be obtained as follows:

$$\mathbf{h}_{SDB}(f) = \frac{|\mathbf{h}^H(f)\mathbf{d}(f)|^2}{\mathbf{h}^H(f)\mathbf{\Gamma}_{diff}(f)\mathbf{h}(f)} \quad \text{subject to } \mathbf{h}^H(f)\mathbf{d}(f) = 1 \tag{3.11}$$

$$\mathbf{h}_{SDB}(f) = \frac{\mathbf{\Gamma}_{diff}^{-1}(f)\mathbf{d}(f)}{\mathbf{d}^H(f)\mathbf{\Gamma}_{diff}^{-1}(f)\mathbf{d}(f)} \tag{3.12}$$

Like in MVDR beamforming, an additional WNG constraint is imposed to the optimization problem to make it more robust to white noise and source localization errors. Compared to DSB, SDB has a narrow main lobe width at low frequencies.

3.4.5 Linear Constrained Minimum Variance Beamforming

Linear Constrained Minimum Variance beamforming (LCMV) is a generalized version of MVDR beamforming. MVDR beamformers imposes only a single constraint, which is the signal distortionless constraint. Like in MVDR, LCMV also minimizes noise power at the output but imposes multiple linear constraints [24, 27]. Suppose the direction of interfering point sources are known, then additional constraints could be imposed such that the beamformer also places a null in those desired directions. Let $\mathbf{C}^H\mathbf{h}(f) = \mathbf{u}$ be the set of linear constraints the beamformer has to satisfy, then LCMV filter can be obtained as follows:

$$\mathbf{h}_{LCMV}(f) = \underset{\mathbf{h}(f)}{argmin}\, \mathbf{h}^H(f)\boldsymbol{R}_n(f)\mathbf{h}(f) \quad \text{subject to } \mathbf{C}^H\mathbf{h}(f) = \mathbf{u} \tag{3.13}$$

$$\mathbf{h}_{LCMV}(f) = [\mathbf{C}^H(f)\boldsymbol{R}_n(f)^{-1}(f)\mathbf{C}(f)]^{-1}\boldsymbol{R}_n^{-1}(f)\mathbf{C}(f)\mathbf{u} \tag{3.14}$$

Generalised Sidelobe Canceller (GSC) is an alternate efficient implementation of LCMV by providing a mechanism for converting the constrained optimization to an unconstrained one. [28] gives a detailed description of the GSC along with various adaptive versions like least mean squares (LMS) and recursive least square (RLS) algorithms.

3.5 Summary

A detailed mathematical explanation regarding the theory behind working of different beamforming techniques was given in this chapter. Next chapter discusses about CHiME Challenge which is designed for multichannel distant speech recognition applications.

Chapter 4

CHiME Challenge

CHiME Challenge is a series of challenges targeting distant speech recognition in real world scenarios. First CHiME Challenge was introduced in 2011 and complexity of the tasks have evolved with every challenge. Over the years, participants from all over the world both from academia and industries have submitted to CHiME Challenge resulting in major breakthroughs in this area. Latest edition will be the fifth in series which will be starting on January 2018.

The work in this thesis uses the datasets and baselines provided by the CHiME 4 challenge. Following sections will give a brief description of CHiME 1 and CHiME 2 tasks followed by a detailed description of datasets used in CHiME 3 and CHiME 4.

4.1 CHiME 1 & CHiME 2

The first and second editions was introduced focussing on distant speech recognition in domestic environments. The aim of the CHiME-1 challenge was to recognize keywords within noisy and reverberant utterances spoken in a living room. The data required for the challenge was simulated by convolving GRID utterances with the binaural room impulse responses (BRIR) and then mixing with the CHiME background audio. The BRIR was recorded using a mannequin from a distance of 2m directly infront. CHiME background audio consists of 20 hours of non stationary noise data recorded using binaural mannequin from the living room of a family comprising of 2 adults and 2 children. The other major noise sources included TV, outdoor noises, toys, footsteps and other electronic gadgets. The reverberated utterances was placed in the CHiME background data in such a manner to produce mixtures at 6 different SNRs. So no scaling of the speech or noise amplitudes was required.

CHiME 2 was introduced to address some of the limitations of CHiME in emulating the real world scenarios namely the stationary speaker scenario and smaller vocabulary. Two separate tracks were present to evaluate both separately. For Track1, time varying BRIRs to account for small head movements was simulated by first recording BRIRs at different places and then interpolating it. The data was simulated such that the speaker was static at the beginning and end while making small head motions in between with each movement at most 5cm and a speed of atmost 5cm/s. Track 2 uses a larger vocabulary by adopting Wall Sreet Journal (WSJ0) dataset instead of the GRID utterances. The submitted systems to the Challenge was evaluated based on the Word Error Rates (WERs) obtained on the test data.

4.2 CHiME 3 & CHiME 4

The third and fourth editions were aimed at addressing distant speech recognition in real life noisy environments recorded using a 6 channel microphone array embedded on the frame of a tablet. Fig 4.1 shows the array configuration with Mic2 facing backside and all the others towards the speaker. The data was recorded in four different environments: cafe, street, bus and pedestrian environments. The utterances were based on the WSJ0 corpus which was also used in the previous edition. Two types of data were available: Real and Simulated data. Simulated data consists of artificially generated data in which the clean speech data was mixed with recorded noise while real data consists of recordings which were collected from speakers in the four noisy environments.

CHiME 4 is an extension of CHiME 3 by making the task more challenging by reducing the number of microphones. Three separate tracks were present consisting of

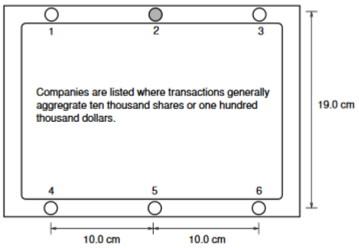

1
2
3
Companies are listed where transactions generally aggregrate ten thousand shares or one hundred thousand dollars.
19.0 cm
4
5
6
10.0 cm
10.0 cm

Figure 4.1: Microphone array geometry [29]

1 mic, 2 mics and 6 mics. CHiME 4 also provided better acoustic and language model baselines.

4.2.1 Data Collection

Data was collected from 12 US English talkers consisting of 6 males and 6 females whose ages were between 20 to 50 years old. For each talker, the data was first collected in an acoustically isolated booth chamber which was not anechoic and then in the four noisy environments. In addition to array microphones, the data was also collected using a close talking microphone (CTM). Each talker had about 100 sentences in each environment which was displayed on the tablet. They were allowed to keep the tablet in whichever way they feel comfortable like holding in front, resting on lap or putting it on a table. The distance from the speaker to the tablet was around 40 cm and all the utterances were based on the WSJ0 prompts. The data was collected originally at 48kHz and then down-sampled to 16kHz and 16 bits.

The talkers were allowed to repeat each sentence as many times until they got it correct. For the purpose of annotation, the annotators chose that sentence which was read correctly. An annotation file was created to record the start and end times of each correct utterance. A padding of 300ms of context was included prior to the start time. Incase there were any errors, the transcriptions were changed to match the best utterance. Apart from the continuous audio stream, isolated audio containing each utterances based on the above annotation was also made available.

4.2.2 Data Simulation

The simulated data for the training set was derived from the clean speech present in WSJ0 training set while development and test set was derived from the CTM data recorded in booth environment. For each WSJ0 utterance in the training set, first a random environment was chosen and then an utterance with duration closest to the current WSJ0 utterance was selected from real recordings which was also from the same environment. Then an impulse response of duration 88ms was estimated for each of the tablet microphones at each time frequency bin using CTM and degraded microphone array data cite. This was done to estimate the SNR of the real recordings [30].

In the second stage, the time delay of arrival (TDOA) at each microphone for the real recordings was estimated using SRP PHAT algorithm. Then a filter was applied to model the direct path delay from speaker to each tablet microphones. Noise was chosen from a random portion of the background noise audio stream belonging to the same environment. Same SNR as that of the real recordings was maintained by adding noise to appropriately scaled version of the obtained speech data.

In the case of development and test set,corresponding real recordings are available for each utterance to be simulated from the booth data. The only difference from the training set simulation is that noise estimated from the corresponding real recordings was added instead from a background audio stream. Noise was estimated by the subtracting real recordings at each channel with signal obtained by convolving the CTM signal with the estimated impulse response.

A major drawback of the simulated data compared to the real data is that, it does not account for the echoes, reverberation, microphone mismatches and microphone failures.

4.2.3 Dataset Description

The dataset was split into training, development and evaluation sets with each containing simulated and real data. The details regarding each set are as follows:

1. Training set: Consists of 1600 utterances in real environments which was spoken

by 4 speakers (2 male and 2 female) with each reading 100 utterances in four environments. Simulated data consists of artificially degraded utterances with the clean speech used for mixing obtained from randomly chosen 7138 utterances of the WSJ0 SI-84 training set comprising of 83 speakers. So the training set consists of a total of 8738 (7138 + 400x4) utterances with a total duration of around 18 hours.

2. Development Set: Consists of 410 real and simulated utterances from each of the 4 environments collected from a total of 4 speakers. The development set consists of 3280 (410x4 + 410x4) utterances. The utterances are based on the "no verbal punctuation" (NVP) part of the WSJ0 speaker-independent 5k vocabulary development set.

3. Test Set: Consists of 330 real and simulated utterances from each of the 4 environments collected from a total of 4 speakers. The development set consists of 2640 (330x4 + 330x4) utterances. As in the development set, the utterances are also based on the "no verbal punctuation" (NVP) part of the WSJ0 speaker-independent 5k vocabulary evaluation set.

4.2.4 Baselines

For the speech enhancement part, a MATLAB code was provided which performs MVDR beamforming with diagonal loading. Non linear SRP PHAT along with Viterbi decoding was used to estimate the location of the speaker. Noise coherence matrix was estimated from 500ms context prior to utterance. The ASR baselines provided were based on the GMM-HMM and DNN-HMM models trained on the noisy data. A detailed description of the ASR models is present in section 5.3

4.3 Summary

A detailed description of Chime Challenge was given in this chapter. The next chapter discusses the proposed approach for the Chime Challenge.

Chapter 5

Proposed Approach

This Chapter gives a complete description of the system proposed for the Chime Challenge and the improvements over the current methods. Most of the beamforming techniques derived in Chapter 3 was based on the assumption that signal received at the microphone is only a delayed version of the speech signal in the presence of additive noise. Frequency domain representation of the received signal is (see Eq 3.1):

$$y_i(f) = s(f)e^{-j2\pi f\tau_{1i}} + n_i(f) \tag{5.1}$$

But this assumption is not valid in real world scenarios where there is reverberation. Let r$_i$(f) be a complex valued function denoting the acoustic transfer function from source to the microphone, then a more appropriate model for received signal will be:

$$y_i(f) = r_i(f)s(f) + n_i(f) \tag{5.2}$$

Now deriving beamformers based on this general signal model will lead to elements of steering vector being replaced by acoustic transfer function from the source to corresponding microphone i.e $\mathbf{d}(f) = [r_1(f)\; r_2(f)\; r_3(f)\; \ldots\; r_N(f)]^T$. Speech distortion will be absent only when the steering vector takes the above form.

One way of finding this steering vector is to take the eigen vector corresponding to maximum eigen value of the source coherence matrix. From Eq 5.2, the coherence matrix for the observed signal can be represented as:

$$\begin{aligned}\mathbf{R}_y(f) = E\{\mathbf{y}(f)\mathbf{y}^H(f)\} &= E\{(\mathbf{d}(f)s(f) + \mathbf{n}(f))(\mathbf{g}(f)s(f) + \mathbf{n}(f))^H\}\\ &= \mathbf{d}(f)\mathbf{d}^H(f)\sigma_s(f) + E\{\mathbf{n}(f)\mathbf{n}^H(f)\}\\ &= \mathbf{R}_s(f) + \mathbf{R}_n(f)\end{aligned}$$

Here $\mathbf{R}_s(f)$ is a rank-1 matrix and the steering vector could be obtained by finding the principal eigen vector of $\mathbf{R}_s(f)$. Zhao et. al [31] uses a simplified model by assuming speech signal undergoes a delay and a frequency dependant attenuation. The model is given by:

$$y_i(f) = g_i(f)s(f)e^{-j2\pi f\tau_i} + n_i(f) \tag{5.3}$$

where $g_i(f)$ is real valued gain factor to account for the effects of the propagation energy decay and the amplification gain of the ithmicrophone. The steering vector based on this model is given by $\mathbf{d}(f) = [g_1(f)e^{-j2\pi f\tau_1}\ g_2(f)e^{-j2\pi f\tau_2}\ g_3(f)e^{-j2\pi f\tau_3} \ldots g_N(f)e^{-j2\pi f\tau_N}]^T$.

5.1 Steering Vector Estimation

This section discusses the proposed approach to estimate the frequency dependent gain to obtain an improved steering vector model. The steering vector involves estimation of two parameters : the gain factor and TDOA. TDOA is computed using SRP PHAT localization method discussed in section 2.3. Method is a slight modification of method discussed in [31], where it find the relative gains with respect to a reference microphone. Signal received at the microphone in a noise free scenario can be represented as:

$$y_i(f) = g_i(f)s(f)e^{-j2\pi f\tau_i}$$

The relative gain at the ith microphone is computed by finding the ratio of cross correlation between signals at ith microphone and reference microphone to the auto correlation of the signal at the reference microphone

$$\frac{|E\{y_i(f)y_r^*(f)\}|}{E\{y_r^*(f)y_r^*(f)\}} = \frac{g_i(f)g_r(f)\sigma_s(f)}{g_r(f)g_r(f)\sigma_s(f)} = \frac{g_i(f)}{g_r(f)} \tag{5.4}$$

Inorder to calculate the above expectation, [31] uses only those bins which are dominated by speech. Speech dominant bins was found using a combination of noise floor tracking, onset detection and coherence test. Now suppose the reference channel is noise free, then the absolute value of cross correlation between the noise free reference channel and noisy input signal can be expressed as:

$$\begin{aligned} |E(y_i(f)y_r^*(f)| &= |E\{(g_i(f)s(f)e^{-j2\pi\tau_i} + n_i(f))(g_r(f)s(f)e^{-j2\pi f\tau_r})^*\}| \\ &= g_i(f)g_r(f)\sigma_s(f) \end{aligned} \tag{5.5}$$

which is same as the numerator in Eq 5.4. In this work, the reference channel was obtained by applying DSB to the input signals. Fig 5.1 shows the block diagram for estimating the gain. Delay block phase aligns the speech signals in all the channels using TDOAs estimated from SRP PHAT algorithm. Normalized Cross Correlation blocks computes the expectation of each input channel with reference channel y$_{DSB}$(f) as in Eq 5.4 to produce the respective gains of each channel.

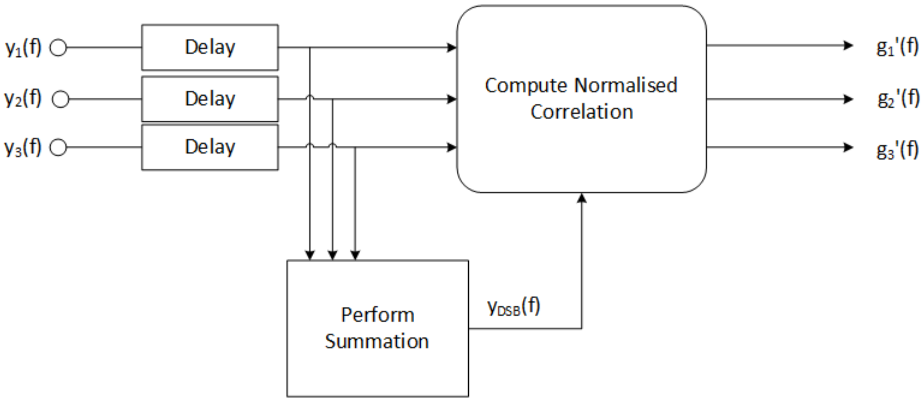


Figure 5.1: Gain Computation

5.2 Beamforming

MVDR beamforming was performed with an improved model of the steering vector estimated from the previous section. Noise coherence matrix was estimated from silence region 500ms prior to each utterance. Diagonal loading factor of 10^{-3} was used.

5.3 Automatic Speech Recognition

For the purpose of ASR, the WERs were evaluated using two acoustic models: GMM-HMM system and a DNN-HMM system . Models trained on noisy training data of the Chime Challenge were provided as baselines. For language modelling , trigram model was available along with 5-gram and recurrent neural networks (RNN) language models used for lattice rescoring. Following sections give a brief description regarding the acoustic models:

5.3.1 GMM-HMM Model

The acoustic features were based on the Mel Frequency Cepstral Coefficients (MFCC). A 91-dimensional feature vector was formed after extracting 13 MFCC features from the current frame and appending it with MFCC vectors from the 3 left and 3 right frames. After performing linear discriminative analysis (LDA), the feature vector was reduced to a 40 dimensional vector. Each feature vector was used to model one of the 2500 tied tri-phone HMM states or senone using a multivariate Gaussian mixture model (GMM) comprising of 6 Gaussians. Maximum likelihood linear transformation (MLLT), and feature-space maximum likelihood linear regression (fMLLR) with speaker adaptive training (SAT) are also applied.

5.3.2 DNN-HMM Model

The deep neural network comprised of 7 hidden layers with each layer containing 2048 neurons. The input to the network consists of 440-dimensional vector obtained by appending 40-dimensional log Mel filter bank features of the current frame with the features from 5 left and 5 right frames. Generative pre-training with restricted Boltzmann machines (RBMs) was performed to initialize the layers of the network. DNN was trained with cost function based on sequence discriminative criterion using the state-level minimum Bayes risk (sMBR) criterion.

Chapter 6

Source Localization Experiments

This Chapter presents a detailed study on the experiments conducted to test the performances of different TDOA based source localization algorithms under various conditions. First a brief description about database used for the experiments and how the multichannel data was simulated is given.

6.1 TIDigits Database

TIDigits corpus containing connected digits sequences was chosen for the experiments in this work. Entire database consisted of a total of 326 speakers comprising of 111 men, 114 women, 50 boys and 51 girls with each pronouncing 77 digit sequences. The adults portion of the database containing men and women were selected for simulation purposes.The sequences were made up of : "zero", "oh", "one", "two","three", "four", "five", "six", "seven", "eight", and "nine". For the purpose of ASR, adult database is further split into train and test set. Training set comprised of 55 men and 57 women speakers, while the test consisted of 56 men and 57 women speakers. For all the experimented in this work, a subset of TIDigit testing set was chosen by randomly selecting 5 male and 5 female speakers containing a total of 770 utterances.

6.2 Room Impulse Response (RIR)

The reverberant speech captured by a microphone from a source located at a fixed position in an enclosed room can be modelled as the output of an LTI system. The impulse response of this LTI system is referred to as RIR. Typically an RIR is separated into two parts: early reflections and late reflections (reverberant tail). The early part comprises of part of RIR upto 50ms after the arrival of direct path signal. Early part help improve the intelligibility of the speech by reinforcing the sound. Signals which arrive after larger delays (>50ms) called the late reflections severely affect the ASR performances. Following are the different parameters which characterizes reverberation:

1. Reverberation Time (T_{60}): Time taken for the energy of the sound to decay by 60dB once the sound source has been turned off.

2. Direct to Reverberation Ratio (DRR): Ratio of the energy of the the direct path to the energy of the reverberant part

3. Early-to-Reverberation Ratio (ELR): Ratio of the energy of the early part of the RIR to the energy of the late part of the RIR.

6.3 Multichannel Data Simulation

For testing the performances of the source localization and beamforming algorithms under different noise and reverberant conditions, multichannel data was simulated from single channel TIDigits corpus. Habets RIR generator, a room impulse generator based on the image method was used to obtain the required impulse responses [32]. RIRs corresponding to two different T_{60} timings of 750ms and 500ms were generated for a source located at an angle of 45^o and a distance of 2m from the centre of an 8-channel microphone array of radius 10cm. RIRs were generated at a sampling frequency of 16kHz. Following are the parameters involved in generating the RIRs using the simulator :

Parameter	**Value**
Room Dimensions	6.67m x 6.14m x 6.57m
T60	750ms
Mic Centre	(3.27,3.50,0.72)
Source Distance	200cm
Source Angle	45^o
Source Location	(4.684,4.9142,0.72)

Table 6.1: Room1 Parameters

Parameter	**Value**
Room Dimensions	4.89m x 6.27m x 2.57m
T60	500ms
Mic Centre	(2.73,2.76,0.72)
Source Distance	200cm
Source Angle	45^o
Source Location	(4.1442,4.174,0.72)

Table 6.2: Room2 Parameters

Eq 6.1 represents signal model used to generate reverberant noisy data :

$$x_i(t) = h_i * s(t) + n_i(t) \tag{6.1}$$

where $h_i(t)$ and $n_i(t)$ represents the RIR and the noise at the i^{th} microphone respectively. Following are the steps adopted to simulate data create such a signal model :

1. The clean speech data was first convolved with the RIRs generated by the Habets RIR generator. Microphone locations, source location, T_{60} parameter and room dimensions were given as input to the RIR generator.

2. From the 8-channel background noise data provided by the REVERB challenge, a random portion was selected with length same as that of reverberant speech data.

3. A Voice Activity Detector (VAD) was used to find the regions where only speech was present and energy of that portion was taken as the signal energy.

4. Noise energy was calculated by finding the energy of the noise samples corresponding to speech regions mentioned in the previous step.

5. The noise was added to the reverberant speech data, after multiplying the noise by a scaling factor to get the desired SNR

$$ScalingFactor = \sqrt{\frac{SignalEnergy}{NoiseEnergy} * 10^{-SNR/10}}$$

The Voice Activity Detector (VAD) was used so as to ensure that addition of silence to the speech doesn't change the measured SNR .

6.4 Source Localization

The performances of different source localization algorithms were compared under varying SNR and reverberant conditions. To evaluate the performances, DOA was estimated for each of 770 audio files in the test set and the following the measures were computed

- Mean : Average of the DOA taken across all the DOA estimates.
- Standard Deviation (Std Dev) : A measure of how much degree the estimates vary about the mean value
- Mode : DOA estimate which occurs the maximum number of times or in other words with the largest frequency.
- Frequency : Ratio of the count of the maximally occurring DOA estimate to the total number of estimates

A good source localization algorithm should have mean value and mode value close to the actual DOA, low standard deviation and frequency near to 1. Following sections shows the performances for various source localization algorithms under different conditions for a source located at an angle of 45^o and a distance of 200cm from the microphone array.

6.4.1 In Presence of Noise

First different source localization algorithms were evaluated in noise only scenario by varying the SNR. Signals received at the each microphone were modelled using the equation:

$$y_i(t) = s(t - \tau_i) + n_i(t)$$

Tables 6.3 and 6.4 shows the performances of Cross Correlation (CC), Smoothed Coherence Transform (SCOT), Hannan Tranform (HT), Generalized Cross Correlation Phase Transform(GCC PHAT) at 20dB and 10dB noise respectively.

Method	Mean	Std Dev	Mode	Frequency
CC	44.99	0.11	45	0.99
SCOT	45.25	7.72	45	0.99
HT	44.99	0.14	45	0.99
GCC PHAT	45.26	8.27	45	0.99

Table 6.3: Source Localization Performance at 20dB for a source located at 45^o

Method	Mean	Std Dev	Mode	Frequency
CC	46.16	16.07	45	0.95
SCOT	45.24	7.77	45	0.96
HT	45.04	0.64	45	0.96
GCC PHAT	46.70	20.47	45	0.95

Table 6.4: Source Localization Performance at 10dB for a source located at 45^o

At 20dB, all the source localization algorithms were able to estimate the direction of source with a high accuracy which is evident from mean value close to true angle, low standard deviation and high frequency associated mode value of 45^o. As the SNR is decreased to 10dB, there is a decline in the performance of the algorithms especially in CC and GCC PHAT algorithm. HT and SCOT algorithms doesn't show much degradation when the SNR is lowered. This is because of the weighing factor used in both the algorithms which helps in giving more weight to those frequency bins with high SNR for reliable estimation of TDOAs. HT algorithm in particular was observed to be more robust to noise with a mean value very close to 45^o and standard deviation less than 1^o.

6.4.2 In Presence of Reverberation

Performance of source localization algorithms were studied in a reverberant scenario by varying T_{60} parameter. The impulses responses were simulated for a point source located at an angle of 45^o and 2m away from the microphone array with T_{60} parameter of 750ms and 500ms. Signals received at the each microphone were modelled using the equation:

$$y_i(t) = h_i(t) * s(t)$$

Tables 6.5 and 6.6 shows the performances of the various source localization algorithms for T$_{60}$ parameter of 750ms and 500ms.

Method	**Mean**	**Std Dev**	**Mode**	**Frequency**
CC	57.79	20.63	48	0.11
SCOT	46.72	5.21	45	0.54
HT	47.95	1.12	48	0.95
GCC PHAT	47.21	3.90	45	0.56

Table 6.5: Source Localization Performance at $T_{60} =$ 750ms for a source located at 45^o

Method	**Mean**	**Std Dev**	**Mode**	**Frequency**
CC	58.97	28.14	50	0.05
SCOT	43.52	12.55	45	0.29
HT	48.66	2.18	48	0.48
GCC PHAT	45	3.44	45	0.59

Table 6.6: Source Localization Performance at $T_{60} =$ 500ms for a source located at 45^o

In presence of reverberation, the generalised cross algorithms performed better than cross correlation algorithm. This is because GCC algorithms exhibits sharp peaks leading to less overlap among adjacent components as explained in section 2.2.2 making it more robust to reverberation.

GCC PHAT which uses only phase information of the cross power spectrum has the best performance in both the scenarios. GCC PHAT gives equal weightage to all bins while HT method gives an additional weighing factor based on the coherence between the channels. Even though this additional factor helped in the presence of noise, it degraded the performance in case of reverberation.

6.4.3 Under Both Noise and Reverberation

Finally the performance of source localization algorithms were tested in the presence of both noise and reverberation. Signals received at the each microphone were modelled using the equation:

$$y_i(t) = h_i(t) * s(t) + n_i(t)$$

Tables 6.5 and 6.6 shows the performances of the various source localization algorithms for SNRs 20dB and 10dB with a fixed T_{60} parameter of 750ms.

Method	**Mean**	**Std Dev**	**Mode**	**Frequency**
CC	56.32	15.71	48	0.09
SCOT	46.74	8.78	45	0.60
HT	51.91	7.50	45	0.29
GCC PHAT	47.33	15.90	45	0.65

Table 6.7: Source Localization Performance at 20dB SNR and $T_{60} = 750$ms for a source located at 45^o

Method	**Mean**	**Std Dev**	**Mode**	**Frequency**
CC	63.21	34.44	48	0.07
SCOT	54.87	38.97	45	0.42
HT	78.54	57.32	59	0.11
GCC PHAT	70.33	64.68	45	0.46

Table 6.8: Source Localization Performance at 10dB SNR and $T_{60} = 750$ms for a source located at 45^o

In the presence of both noise and reverberation, the SCOT algorithms performed much better than other algorithms. This is because SCOT algorithm uses a weighing factor based on SNR making it robust to noise, while it also exhibits narrow peaks compared to CC enabling it perform better in reverberant scenarios. On lowering the SNR while keeping the T_{60} parameter same, a severe degradation in performance was a observed with larger deviation of the estimated angle from the true angle and the standard deviation increasing by 30%.

Chapter 7

ASR Experiments

This Chapter presents various results on Automatic Speech Recognition based on two different datasets. A preliminary set of experiments were performed on the simulated multichannel data obtained from the TIDigits database. This simulated data help provided a flexible framework for evaluating the performances of various beamforming algorithms present in the literature. Second set of experiments were based on the Chime Challenge database which provided both real and simulated data. This facilitated the study of how the techniques perform on real data and compare the performances with simulated data.

7.1 Experiments on TIDigits

This section continues the work done in previous chapter by comparing the performance of different beamforming algorithms on the same database which is TIDigits. Since the aim of this work is to improve speech recognition accuracies, the metric used for comparing different beamforming algrorithms was WER evaluation on the enhanced data. Following studies were performed on the simulated multichannel data obtained from TIDigits:

1. Evaluation of source localization algorithms based on WER

2. ASR performances of various beamforming algorithms

3. Robustness to source localization errors by WER comparison

The next section gives an overview of the speech recognition system used for evaluating the WERs

7.1.1 Speech Recognition Engine

Speech recognition was performed using the Kaldi open source toolkit based on C++ [33]. Kaldi performs the complete speech recognition task involving feature extraction, acoustic modelling using GMM-HMM models, language modelling, creating graphs and decoding. The widely used 39 dimensional MFCC vectors were used as the features to represent various phones. Phones represent the smallest unit of sound while triphones are used to model phones based on their left and right context. But for 50 phones, the number of possible triphone combinations will be 50^3 which lead to a huge increase in the number of parameters to be learnt and consequently data required for training. So for computational purposes, similar triphones were clustered to form senones.

The acoustic properties of a spectrum corresponding to a phone can vary greatly depending on many factors - phone context, speaker, style of speech and so on. Hence a Gaussian Mixture Model(GMM) was used to model the statistical distribution of both the monophones and the senones. Parameters of the GMM were estimated using the Maximum Likelihood(ML) estimation on clean speech training data. In this work, a the triphone model was trained with 500 senones and 8 Gaussians were used to model each senone. Final speech recognition output was based on the combination of GMM-HMM acoustic model and a bigram language model.

7.1.2 ASR Performances of Beamforming Algorithms

This section compares the performances of different beamforming algorithms on noisy reverberant data simulated from TIDigits database. The test data had SNR of 20dB and impulse response was simulated for a source located 2m away at an angle 45^o with T_{60} parameter 750ms. Table 7.1 shows the WERs obtained for different beamforming algorithms using SCOT source localization algorithms.

Method	WER (%)
Clean Speech	3.48
Degraded Speech	33.68
BeamformIt	19.17
DSB	17.87
MVDR	17.64

Table 7.1: WERs of different beamforming algorithms at 20dB SNR and $T_{60} = 750$ms for a source located at 45^o

Clearly an improvement is observed, with WERs reducing from 33.68 % for degraded data to around 18% after applying multichannel enhancement techniques. Both MVDR and DSB have similar performances, and also showed an improvement over the reference method BeamformIt. These beamforming techniques are not able to completely suppress the noise and reverberation. Clearly some residual noise and reverberation are present in enhanced data creating mismatch with training data leading to higher WER when compared with clean speech.

7.1.3 Robustness to Source Localization Errors

Inorder to test the robustness of different beamforming algorithms to errors at the source localization stage, steering vectors were computed using two following methods :

1. Actual TDOA computed using the known geometry (True Delays)
2. Actual TDOA rounded off to the nearest integer (Rounded Delays)

Table 7.2 shows the ASR performances of DSB and MVDR beamforming methods when the steering vectors were obtained using Oracle Delays (ground truth) and when the oracle delays were rounded off to the nearest integer.

Method	Oracle Delay	Rounded Delay
DSB	17.47	18.14
MVDR	17.17	20.91

Table 7.2: WER(%) of MVDR and DSB at 20dB SNR and $T_{60} = 750$ms for a source located at 45^o with steering vector estimated using oracle and rounded delays

The results in Table 7.2 shows that there is a decrease in ASR performance of MVDR by 3.76% when the sample delays were rounded off to the nearest integer. This shows that MVDR is less robust to errors at the source localization stage. On the other hand, DSB had a decrease in WER by about 1% showing that it is more robust to source localization errors.

7.1.4 Multicondition Training

One of the major problems of poor ASR performance in noisy and reverberant conditions is the train test mismatch. As discussed in previous section there exist some residual noise and reverberation after enhancement. Inorder to reduce this mismatch, the training set was replaced by two types of data.

For creating the training set, clean speech of TIDigits training corpus was convolved with RIR simulated for Room2 with $T_{60} = 500$ms and speaker at a distance of 2m making an angle 45^o with respect to the microphone array (Reverb Train). Then noise was added to above data at 20dB SNR and the second model (MVDR Train) was trained using data obtained after performing MVDR to this noisy reverberant data.

For testing, a reduced version of clean speech data of TIDigits testing corpus comprising 10 speakers with RIR simulated for Room1 with $T_{60} = 750$ms and speaker at a distance of 2m making an angle 45^o with respect to the microphone array.

Table 7.3 shows the WERs on test data after performing MVDR beamforming models described above.

Model	**WER (%)**
Clean Speech Model	17.64
Reverb Train	6.76
MVDR Train	2.8

Table 7.3: WER (%) of MVDR beamforming after decoding using acoustic models trained on clean speech, reverberant speech and enhanced speech

Significant improvements are observed after performing multi condition training. Using the acoustic models trained on the reverberant help improve the WER to 6.76% compared to acoustic model trained using clean speech data. This clearly shows that there is residual reverberation present after enhancement using MVDR beamforming.

Finally when using acoustic model trained on MVDR beamforming is applied, the WER has reduced to 2.8%. The results using the simulated might be over optimistic and same performance cannot be guaranteed to real data due to two reasons. First the source was stationary and was located at same angle of 45^o for the training data and test data. This is not practical in real world scenarios. Other reason is that noise added is stationary background noise in a room environment. In real world, ASR system could encounter non stationary noises which are difficult to handle. So inorder to overcome the limitations of current experiments, the Chime Challenge data was used to evaluate performances of algorithms in real data and compare the performances with simulated data.

7.2 Chime Challenge Results

This section describes the ASR perfromances of the beamforming algorithms on the Chime Challenge data explained in chapter 4. Proposed approach described in chapter 5 was evaluated in real and simulated conditions and compared with performances of existing algorithms.

7.2.1 ASR WERs using GMM-HMM trigram model

Various multichannel enhancement techniques were performed on the Chime Challenge data and WERs were obtained using GMM-HMM system trained on noisy data with a trigram language model. Without applying any enhancements, the average WER obtained was 23.3%, with 22.16% for real data and 24.44% for simulated data. Table 7.4 shows the WERs obtained for different beamforming techniques. In all the beamforming algorithms except BeamformIt, the TDOAs were estimated using SRP PHAT localization algorithm. Some of the important observations are :

- Comparing Delay + DSB and Delay + Gain + DSB algorithms, incorporating gain into the steering vector has helped in decreasing the WER. The average WER has decreased from 23.3% to 12.04% after applying the latter algorithm to degraded multichannel data.

Method	**Real**	**Simu**	**Average**
Delay + DSB	12.71	13.73	13.22
Delay + Gain + DSB	**12.04**	12.05	12.04
MCA	12.77	11.84	12.30
BeamformIt	12.99	14.30	13.64
Delay + MVDR	17.12	10.67	13.92
Delay + Gain + MVDR	12.75	**10.48**	**11.62**

Table 7.4: WER (%) obtained on Chime Challenge development set using a GMM-HMM model trained on noisy data with a trigram language model

- The DSB algorithm with gain based steering vector have similar performance both in real and simulated environments indicating it is a more robust algorithm.

- The MCA algorithm in comparison to Delay + Gain + DSB have a higher WER in real data and a lower WER in simulated data. Only difference between both algorithms is that MCA tries to align the phase of each channel to the reference channel. In real data, the phase of the reference channel cannot be treated as the phase of required speech signal due to presence of reverberation. Since simulated data doesnot contain any reverberation, phase of signal will be close to the phase of original speech signal and hence improvements are obtained in this case.

- BeamformIt performs a weighted delay and sum beamforming where the weights depend on the cross correlation across channels. The superior performance of DSB is due to presence of better SRP PHAT localization algorithm while BeamformIt uses GCC PHAT.

- MVDR algorithm performs best for simulated data with around 1 % absolute WER improvement over the other methods. Unlike other techniques, MVDR is a data dependant beamforming technique, where the filter weights are designed according to second order statistics of the estimated noise. Due to this MVDR has a superior performance especially when directional noises are present.

- MVDR has a comparatively poor performance in real data because it is sensitive to microphone gains and the errors in the source localization stage. The width of the main lobe in case of a MVDR beamformer is very narrow leading to signal cancellation problems in the presence of source localization errors. Simulated data doesnot account for varying microphone gains and reverberation.

- Adding gains to the steering vector help improve the performance of the MVDR beamforming in real data by bringing about 4% absolute WER improvement over the baseline MVDR. Significant improvements is observed in real but not in simulated compared to baseline MVDR because of absence of reverberation in simulated data.

7.2.2 Effect of Single Channel Enhancement

To the output of best performing system in the previous section based on MVDR with gain based steering vector, single channel enhancement techniques based on Non-negative Matrix Factorization (NMF) were applied. Two types of NMF techniques were applied: one using a convolutive NMF (CNMF) model and the other was convolutive NMF with speech model (CNMF+NMF).

Post processing	**Real**	**Simu**	**Average**
None	12.75	10.48	11.62
CNMF	15.25	12.87	14.06
CNMF + NMF	14.26	12.08	13.17

Table 7.5: WER (%) obtained with NMF based post processing methods to Delay + Gain + MVDR

As observed in Table 7.5, use of NMF based single channel enhancements have increased the WERs. The degradation in performance is due to the presence of residual noise after MVDR beamforming. NMF methods implemented are designed to deal with reverberation, but not robust to noise. Also reverberation is not present in the case of simulated data. Due to these reasons, NMF based single channel enhancement techniques didn't give any improvement in ASR WERs.

7.2.3 Effect of DNN-HMM Model

The GMM-HMM acoustic model used was changed to a DNN-HMM model and WERs of the different beamforming algorithms were evaluated. The architecture of The DNN-HMM model trained on the noisy training data was explained in section 5.3.2. Table 7.6 shows the WER obtained using a DNN-HMM model with a trigram language model.

Method	**Real**	**Simu**	**Average**
BeamformIt	8.14	9.03	8.59
Delay + DSB	8.08	8.29	8.18
Delay + Gain + DSB	7.87	7.73	7.80
Delay + MVDR	12.38	6.25	9.31
Delay + Gain + MVDR	8.71	6.60	7.66

Table 7.6: WER (%) obtained on Chime Challenge development set using a DNN-HMM model trained on noisy data with a trigram language model

Around 4% absolute improvement in WER is obtained when the acoustic model is changed from GMM-HMM to DNN-HMM model. This shows that DNN is better able to model the acoustic features compared to DNN. This could be attributed to hidden layers present in the network which is capable of effectively learning the non-linear relationship between the features and corresponding senone class. Delay + Gain + MVDR model produced the best results having WER of 7.66% using a DNN-HMM model compared to 11.62% obtained using a GMM-HMM model. WER of Delay + MVDR algorithm for real data is almost double that of simulated data. Adding the gain factor showed improvements for real data by reducing the WER from 12.38% to 8.71%, but a minor increase in WER was observed for simulated data.

7.2.4 Effect of Lattice Rescoring

This section shows the effect of better language model on ASR WERs. Decoding was performed after rescoring the lattices obtained from DNN-HMM trigram model using 5-gram model and RNN language model. Delay + Gain + DSB had the best average WER of 5.52% while Delay + Gain + MVDR performed the second best with WER of 5.66%.

Method	**Real**	**Simu**	**Average**
BeamformIt	6.85	7.75	7.30
Delay + DSB	6.59	7.29	6.94
Delay + Gain + DSB	6.39	6.66	6.52
Delay + MVDR	10.93	5.29	8.11
Delay + Gain + MVDR	7.39	5.50	6.44

Table 7.7: WER obtained on Chime Challenge development set using a DNN-HMM model trained on noisy data after lattice rescoring with 5-gram language model

Method	**Real**	**Simu**	**Average**
BeamformIt	5.76	6.77	6.27
Delay + DSB	5.55	6.27	5.90
Delay + Gain + DSB	5.35	5.69	5.52
Delay + MVDR	9.85	4.51	7.18
Delay + Gain + MVDR	6.57	4.75	5.66

Table 7.8: WER obtained on Chime Challenge development set using a DNN-HMM model trained on noisy data after lattice rescoring with RNN language model

As observed from table 7.7, while performing lattice rescoring using a 5-gram language model, there is an absolute improvement of around 1% over trigram model. Further improvement of around 1% was observed when lattice rescoring was performed using a superior language model based on RNN.

Chapter 8

Conclusion

As a part of this thesis, performances of various source localization and beamforming algorithms were evaluated under different environment conditions using data simulated based on the TIDIgits database. To test the performances on real data and compare with performances on simulated data, the algorithms were evaluated based on the Chime Challenge data. The performance of DSB was observed to be similar in both real and simulated data. In simulated data, the MVDR beamformer in combination with SRP PHAT localization performed the best among other evaluated algorithms while the performance was poor in real data. Inorder to improve the performance of MVDR beamformer in real data, an improved steering vector was proposed to model the frequency dependant gains due to reverberation and microphone characteristics. The proposed model reduced the WER from 17.12% to 12.75% in real data using an ASR system based on GMM-HMM acoustic model and trigram language model. Improving the acoustic model from GMM-HMM to DNN-HMM increased the recognition accuracy by around 4%. Further 2% absolute WER improvement was obtained when lattice rescoring was performed using RNN language model.

The best performing system was DSB with gain based steering vector had an average WER of 5.52% while the secod best system was MVDR beamformer with gain based steering vector having average WER of 5.66 %. The back end for both systems were based on DNN-HMM model with lattice rescoring using RNN language model.

References

[1] B. Lee, M. Hasegawa-Johnson, C. Goudeseune, S. Kamdar, S. Borys, M. Liu, and T. S. Huang, "Avicar: audio-visual speech corpus in a car environment." in *INTERSPEECH*, 2004.

[2] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[3] C. Zhang, D. Florêncio, D. E. Ba, and Z. Zhang, "Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings," *IEEE Transactions on Multimedia*, vol. 10, no. 3, pp. 538–548, 2008.

[4] K. Kumatani, T. Arakawa, K. Yamamoto, J. McDonough, B. Raj, R. Singh, and I. Tashev, "Microphone array processing for distant speech recognition: Towards real-world deployment," in *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*. IEEE, 2012, pp. 1–10.

[5] T. T. Vu, B. Bigot, and E. S. Chng, "Speech enhancement using beamforming and non negative matrix factorization for robust speech recognition in the chime-3 challenge," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 423–429.

[6] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[7] Y. Hioka, Y. Koizumi, and N. Hamada, “Improvement of doa estimation using virtually generated multichannel data from two-channel microphone array,” *Journal of Signal Processing*, vol. 7, no. 1, pp. 105–109, 2003.

[8] R. Roy and T. Kailath, “Esprit-estimation of signal parameters via rotational invariance techniques,” *IEEE Transactions on acoustics, speech, and signal processing*, vol. 37, no. 7, pp. 984–995, 1989.

[9] M. S. Brandstein and H. F. Silverman, “A robust method for speech signal time-delay estimation in reverberant rooms,” in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 375–378.

[10] C. Blandin, A. Ozerov, and E. Vincent, “Multi-source tdoa estimation in reverberant audio using angular spectra and clustering,” *Signal Processing*, vol. 92, no. 8, pp. 1950–1960, 2012.

[11] C. Knapp and G. Carter, “The generalized correlation method for estimation of time delay,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[12] P. D. Welch, “The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Transactions on audio and electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[13] J. Faneuff, “Spatial, spectral, and perceptual nonlinear noise reduction for hands-free microphones in a car,” Ph.D. dissertation, Worcester Polytechnic Institute, 2002.

[14] M. L. Seltzer, “Microphone array processing for robust speech recognition,” Ph.D. dissertation, Carnegie Mellon University Pittsburgh, PA, 2003.

[15] I. Cohen, J. Benesty, and S. Gannot, *Speech processing in modern communication: challenges and perspectives.* Springer Science & Business Media, 2009, vol. 3.

[16] J. Ramırez, J. C. Segura, C. Benıtez, A. De La Torre, and A. Rubio, “Efficient voice activity detection algorithms using long-term speech information,” *Speech communication*, vol. 42, no. 3, pp. 271–287, 2004.

[17] J. Sohn, N. S. Kim, and W. Sung, “A statistical model-based voice activity detection,” *IEEE signal processing letters*, vol. 6, no. 1, pp. 1–3, 1999.

[18] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi *et al.*, “The ntt chime-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices,” in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on.* IEEE, 2015, pp. 436–443.

[19] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, “Robust mvdr beamforming using time-frequency masks for online/offline asr in noise,” in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on.* IEEE, 2016, pp. 5210–5214.

[20] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, “Blstm supported gev beamformer front-end for the 3rd chime challenge,” in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on.* IEEE, 2015, pp. 444–451.

[21] R. K. Cook, R. Waterhouse, R. Berendt, S. Edelman, and M. Thompson Jr, “Measurement of correlation coefficients in reverberant sound fields,” *The Journal of the Acoustical Society of America*, vol. 27, no. 6, pp. 1072–1077, 1955.

[22] G. W. Elko, “Microphone array systems for hands-free telecommunication,” *Speech communication*, vol. 20, no. 3-4, pp. 229–240, 1996.

[23] E. Warsitz and R. Haeb-Umbach, “Blind acoustic beamforming based on generalized eigenvalue decomposition,” *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 5, pp. 1529–1539, 2007.

[24] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing.* Springer Science & Business Media, 2008, vol. 1.

[25] M. B. Stolbov and S. V. Aleinik, "Improvement of microphone array characteristics for speech capturing," *Modern Applied Science*, vol. 9, no. 6, p. 310, 2015.

[26] J. Bitzer and K. U. Simmer, "Superdirective microphone arrays," in *Microphone arrays.* Springer, 2001, pp. 19–38.

[27] D. H. Johnson and D. E. Dudgeon, *Array signal processing: concepts and techniques.* Simon & Schuster, 1992.

[28] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE assp magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[29] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'chime'speech separation and recognition challenge: Dataset, task and baselines," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on.* IEEE, 2015, pp. 504–511.

[30] E. Vincent, R. Gribonval, and M. D. Plumbley, "Oracle estimators for the benchmarking of source separation algorithms," *Signal Processing*, vol. 87, no. 8, pp. 1933–1950, 2007.

[31] S. Zhao, X. Xiao, Z. Zhang, T. N. T. Nguyen, X. Zhong, B. Ren, L. Wang, D. L. Jones, E. S. Chng, and H. Li, "Robust speech recognition using beamforming with adaptive microphone gains and multichannel noise reduction," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on.* IEEE, 2015, pp. 460–467.

[32] E. Habets, "Room impulse response (rir) generator," https://www.audiolabs-erlangen.de/fau/professor/habets/software/rir-generator, 2008.

[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition

toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.