ABSTRACT

Title of thesis: POSITION CALIBRATION OF ACOUSTIC
SENSORS AND ACTUATORS ON DISTRIBUTED
GENERAL PURPOSE COMPUTING PLATFORMS

Degree Candidate: Vikas Chandrakant Raykar

Degree and year: Master of Science, 2003

Thesis directed by: Professor Rama Chellappa
Department of Electrical Engineering

Dr. Ramani Duraiswami
Institute for Advanced Computer Studies

Arrays of audio/video sensors and actuators (such as microphones, cameras, speakers and displays) along with array processing algorithms offer a rich set of new features for emerging multimedia applications. Until now, array processing was mostly out of reach for consumer applications perhaps due to significant costs of dedicated hardware and complexity of processing algorithms. On the other hand, several mobile computing and communication devices like laptops, PDAs and tablets are equipped with multiple audio/video sensors and actuators. An ad-hoc network of such devices can be used to form a distributed sensor network. A prerequisite for using distributed audio-visual I/O capabilities is to put the sensors and actuators into a common time and space.

This thesis focuses on providing a common space by automatically determining the relative 3D positions of audio sensors and actuators. A closed form approximate solution is derived, which is further refined by minimizing a non-linear error function. The formulation and solution accounts for the lack of temporal synchronization among different platforms. An approximate expression for the mean and covariance of the implicitly defined estimator is derived using the implicit function theorem and approximate Taylors' series expansion. The theoretical performance limits for the sensor positions are derived via the Cramér-Rao bound and analyzed with respect to the number of sensors and actuators as well as their geometry. Extensive simulation results and the practical details of implementing our algorithms in a real-life system are discussed.

POSITION CALIBRATION OF ACOUSTIC SENSORS AND ACTUATORS ON DISTRIBUTED GENERAL PURPOSE COMPUTING PLATFORMS

by

Vikas Chandrakant Raykar

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2003

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Dr. Ramani Duraiswami, Co-Advisor
Professor Min Wu
Professor Shihab Shamma

DEDICATED TO

ACKNOWLEDGEMENTS

I am grateful to those people who have made this research and thesis possible and those who have made my experience in graduate school one that I will always remember fondly. First of all I would like to thank my research advisor Dr. Ramani Duraiswami who inspired me to do research in the area of microphone arrays. His passion and unparalleled enthusiasm for new ideas really put me on the track on how to think on my own to get new developments in the existing field. I am greatly indebted to Prof. B. Yegnanarayana with whom I had an opportunity to interact when he was a visiting professor at Maryland in summer for three months. I really learnt the right way of doing research from him. I would also like to thank my graduate advisor Prof. Rama Chellappa who has been a inspirational figure for me in terms of excellence a person can achieve in a field. I would like to thank Prof. Shihab Shamma and Prof. Min Wu who kindly agreed to be on my thesis committee. A large part of this work was done during my internship at Intel Labs, Intel corporation, Santa Clara. This work would be incomplete without the guidance of my mentors at Intel, Dr. Igor Kozintsev and Dr. Rainer Lienhart, who put on the right track and were helpful in criticizing and improving different aspects of the algorithms. I learnt a great deal during my internship at Intel as to how research can be directed to have significant impact in terms of applications. I would also like to thank Dr. Eric Debes who was fun to hang around with at Intel. Finally, I would also like to thank my co-interns Aaamer for those rollerblading trips, Brinda and Madhavi for the endless coffee hours and fundaes, my previous room mates Karthik, Saurabh and Devang, my room mates at Maryland Niranjan, Adarsh and Urvashi and my current room mates Arun and Punya. Last but not the least I am forever indebted for my parents, for I am what I am wholly due to their efforts and sacrifices.

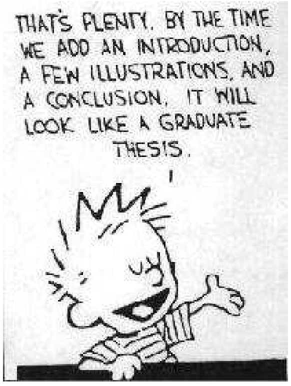

THAT'S PLENTY. BY THE TIME WE ADD AN INTRODUCTION, A FEW ILLUSTRATIONS, AND A CONCLUSION, IT WILL LOOK LIKE A GRADUATE THESIS.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

Arrays of audio/video sensors and actuators (such as microphones, cameras, loudspeakers and displays) along with array processing algorithms offer a rich set of new features for emerging multimedia applications. A typical setup as shown in Figure 1.1 would involve capturing the audio and video scene using multiple microphones and cameras. The captured multiple audio/video streams can be rendered on multiple loudspeakers/displays or used for different applications. A few such applications include multi-stream audio/video rendering, smart conference rooms [35, 34, 37] , meeting recording, hands free voice communication [23, 15], speech acquisition in automobile environments [16, 22], object localization and tracking, hearing-aid devices [14], speech enhancement [25, 24], speech dereverberation and acoustic surveillance (Refer Figure 1.2). In this thesis we are concerned only with acoustic sensors (microphones) and actuators (loudspeakers).

The two main applications for which we can use multiple microphones are for sound source

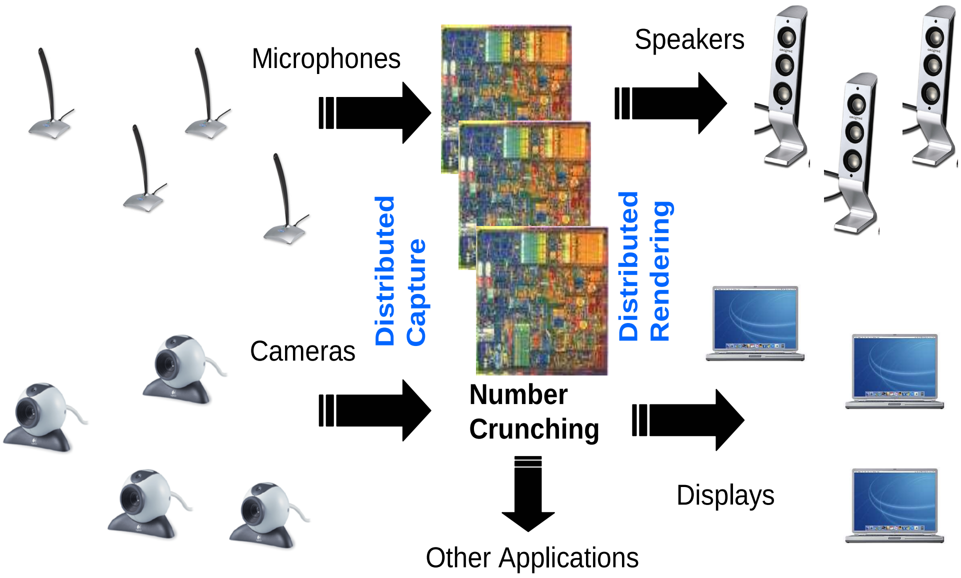


Figure 1.1: A general setup involving multiple microphones, loudspeakers, cameras and displays.

localization and beamforming. Using multiple microphones and knowing the locations of the microphones we can estimate the location of the speaker based on the waveform captured at each of the microphones. Once we know the location of the speaker we can track the moving speaker and beamform to his location. A beamformer does spatial filtering in the sense that it separates two signals with overlapping frequency content originating from different directions.

Consider a typical conference room scenario. The speech signal received from a speaker in such acoustical environments is corrupted both by additive noise and room reverberation. One effective way of dealing with such situations is to use a set of spatially distributed microphones for recording the speech. In order to keep the speaker in focus in videoconferencing, the speaker can be localized, and this information can be fed to a video system for actuating the pan-tilt operations of a camera. Once the actual position of the speaker is known, the microphone array can be steered electronically (beamformed) for high quality speech acquisition. Tracking a moving speaker is also useful in a multispeaker scenario in which speech from a particular speaker may need to be enhanced with respect to others, or with respect to noise sources.

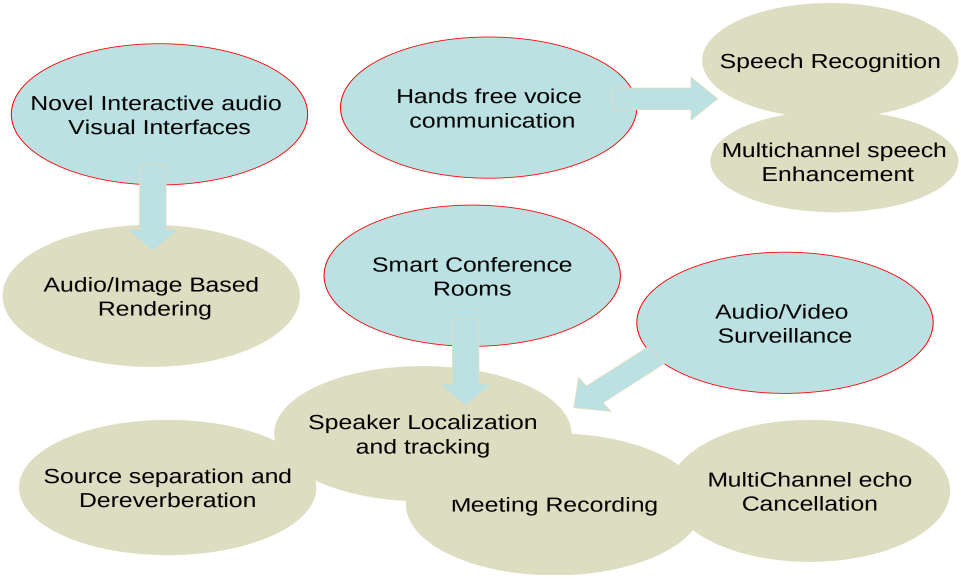

Novel Interactive audio Visual Interfaces
Hands free voice communication
Speech Recognition
Multichannel speech Enhancement
Audio/Image Based Rendering
Smart Conference Rooms
Audio/Video Surveillance
Speaker Localization and tracking
Source separation and Dereverberation
Meeting Recording
MultiChannel echo Cancellation

Figure 1.2: Some typical applications involving multiple audio/video sensors and actuators.

1.2 Distributed Array Processing

Much of the current work has focussed on setting up all the sensors and actuators on a single dedicated computing platform. Such a setup would require a lot of dedicated infrastructure in terms of the sensors, multi-channel interface cards and computing power. For example, to setup a microphone array on a single general purpose computer we need expensive multichannel sound cards and a CPU with huge computation power to process all the multiple streams. At the same time, recent advances in mobile computing and communication technologies suggest a very attractive platform for implementing these algorithms. Students in classrooms, co-workers at meetings, family members at home are nowadays accompanied by one or several mobile computing and communication devices like laptops, PDAs, tablets, with multiple audio and video I/O devices onboard. We collectively refer to such devices as General Purpose Computers (GPCs). In addition, high-speed wireless network connections, like IEEE 802.11a/b/g, are available to network those devices. If we manage to combine sensors/actuators with wireless connectivity and computational resources, we can potentially transform such a network into a complex array Digital Signal Processing system. The advantage of such an approach is that

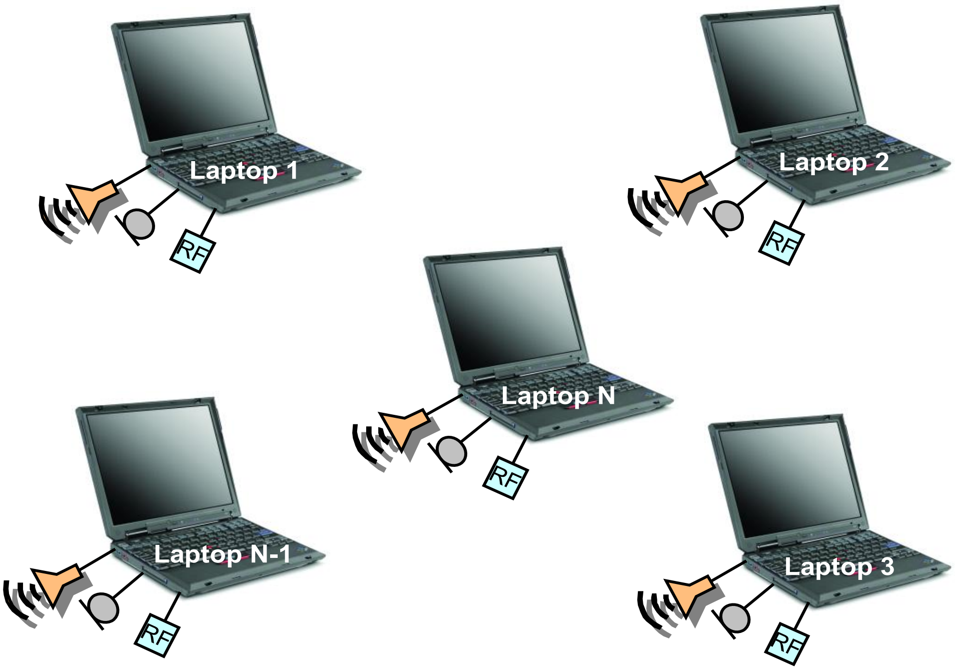


Figure 1.3: Distributed computing platform consisting of N general-purpose computers along with their onboard audio sensors, actuators and wireless communication capabilities.

multiple GPCs along with their sensors and actuators can be converted to a distributed sensor network in an ad-hoc fashion by just adding appropriate software layers. No dedicated infrastructure in terms of the sensors, actuators, multi-channel interface cards and computing power is required. However, there are several important technical and theoretical problems that need to be addressed before the idea of using GPCs for array signal processing algorithms can materialize in real-life applications. Figure 1.3 shows a schematic representation of our *distributed computing platform* consisting of N GPCs. Each GPC is assumed to be equipped with audio sensors (microphones), actuators (speakers) for performing audio I/O, and wireless communication capabilities for exchanging data between each other.

1.3 Common Time and Space

A prerequisite for using distributed audio-visual I/O capabilities is to put sensors and actuators into a common time and space. [19] proposes a way to provide a common time reference for multiple distributed GPCs with the precision of ten's of microseconds. This thesis is mainly concerned with providing a common space (relative coordinate system) by means of actively estimating the three dimensional positions of the sensors and actuators. Many multi-microphone array processing algorithms (like sound source localization or conventional beamforming) need to know the positions of the microphones very precisely. Even relatively small uncertainties in sensor location could make substantial, often dominant, contributions to overall localization error [27].

1.4 Previous work

Current audio array processing systems either rely on placing the microphones in known locations or manual calibration of their positions. There are some approaches which do position calibration using speakers in known locations. [28] describes an experimental setup for automatic calibration of a large-aperture microphone array using acoustic signals from transducers whose locations are known. We follow a more general approach where we assume that the speakers locations are also unknown. A lot of related theoretical work can be found in [27, 36, 21]. Most of the formulations assume that all the sensors and actuators are on a synchronized setup i.e capture and playback occur simultaneously. However in a typical distributed setup we start the audio capture and playback on each GPC one by one and the playback and the capture start time are generally unknown. Our solution explicitly accounts for the errors in localization due to lack of temporal synchronization among different platforms. A recent paper [20] accounts only for the unknown source emission time. The solution turns out to be a non-linear minimization problem which requires a good starting point to reach the global minimum. We derive a closed form approximate solution to be used as initial guess for the minimization routine.

The problem of self-localization for a network of nodes has also been dealt in the wireless network and robotics community . The problem is essentially the same as in our case but the ranging method differ depending on the sensors and actuators. The problem of self-localization of a network of nodes involves two steps: ranging and multilateration. Ranging involves the estimation of the distance between two nodes in the network. Multilateration refers to using the estimated ranges to find the position of different nodes. The ranging technology can be either based on the Time-Of-Arrival (TOA) or the Received Signal Strength (RSS) of acoustic, ultrasound or radio frequency (RF) signals. The choice of a particular technology depends on the environment and the range for which the sensor network is designed. The GPS system and long range wireless sensor networks use RF technology for range estimation. Localization using Global Positioning System (GPS) is not suitable for our applications since GPS systems do not work indoors and are very expensive. Also RSS based on RF is very unpredictable [29] and the RF TOA is very small to be used indoors. [29] discuss systems based on ultrasound TOA using specialized hardware (like motes) as the nodes. However, our goal is to use the already available sensors and actuators on the GPCs to estimate their positions. So our ranging technology is based on acoustic TOA as in [28, 20, 13]. Once we have the range estimates the Maximum Likelihood (ML) estimate can be used to get the positions.

1.5 Organization

The thesis is organized as follows. In Chapter 2, we formulate the problem and derive the Maximum Likelihood (ML) estimator. We derive two estimators, one based on TOF and the other based on TDOF. In Chapter 3, an approximate closed form solution is derived, which can be used as an initial guess for the non-linear minimization routine. In Chapter 4, we derive the theoretical mean and covariance of the estimated parameters. The Cramér-Rao bound is also derived and analyzed for its sensitivity with respect to the number of sensors and actuators as well as their geometry. Chapter 5 gives a discussion of the issues involved in designing a practical system. Chapter 6, concludes with a summary of the present work.

1.6 Novel Contributions

The following are the novel contributions of this thesis.

- We propose a novel setup for array processing algorithms with ad-hoc connected GPCs.

- The position estimation problem has been derived as a maximum likelihood in several papers [20, 36, 28]. The solution turns out to be the minimum of a nonlinear cost function. Iterative nonlinear least square optimization procedures require a very close initial guess to converge to a global maximum. We propose the technique of metric Multidimensional Scaling (MDS)[32] in order to get an initial guess for the nonlinear minimization problem. Using this technique, we get the approximate positions of GPCs.
- Most of the previous work on position calibration (except [13] which describes a setup based on Compaq iPAQs and motes) are formulated assuming time synchronized platforms. However in an ad-hoc distributed computing platform consisting of heterogeneous GPCs we need to explicitly account for errors due to lack of temporal synchronization. We perform an analysis of the localization errors due to lack of synchronization among multiple platforms and propose ways to account for the unknown emission start times and capture start times.
- Most of the existing localization methods use the Time Of Flight (TOF) approach for position calibration [20, 28, 13]. We show that for distributed computing platforms, the method based on Time Difference of Flight (TDOF) is better than the TOF method in many respects.
- We derive the approximate mean and covariance of the implicitly defined estimator using the implicit function theorem and Taylor series expansion as in [11]. We also derive the Cramèr-Rao bound and analyze the localization accuracy with respect to the number of sensors and sensor geometry.

The work presented in this paper resulted in two conference publications and two patents being filed.

Publications

- *Position Calibration of Audio sensors and actuators in a distributed computing platform* Vikas C. Raykar, Igor Kozintsev and Rainer Lienhart , ACM Multimedia 2003, Berkeley, CA, USA, November 2003.

- *Self Localization of acoustic sensors and actuators on distributed platforms* Vikas C. Raykar, Igor Kozintsev and Rainer Lienhart, ICCV 2003 International Workshop on Multimedia Technologies in E-Learning and Collaboration, Nice, France, October 2003.

Patents filed

- Three-Dimensional Position Calibration of Audio Sensors and Actuators on a Distributed Computing Platform. (filed on 05/09/2003 along with Igor Kozintsev and Rainer Lienhart)
- Method for 3-Dimensional position calibration of audio sensors and actuators on a distributed computing platform. (filed on 08/29/2003 along with Igor Kozintsev and Rainer Lienhart)

Chapter 2

Problem Formulation

2.1 Problem statement and notation

Given a set of M acoustic sensors (microphones) and S acoustic actuators (speakers) in unknown locations, our goal is to estimate their three dimensional coordinates. We assume that each of the GPCs has at least one microphone and one speaker. We also assume that at any given instant we know the number of sensors and actuators in the network. Any new node entering/departing the network announces its arrival/departure by some means, so that the network of sensors and actuators can be recalibrated.

Each of the speaker is excited using a known calibration signal such as maximum length sequence or chirp signal and the signal is captured by each of the acoustic sensors. The Time of Flight (TOF) is estimated from the captured audio signal. The TOF for a given pair of microphone and speaker is defined as the time taken by the acoustic signal to travel from the speaker to the microphone[1]. We assume that the signals emitted from each of the speakers do not interfere with

[1] In some papers, TOF is referred to as Time Of Arrival (TOA).

each other i.e. each signal can be associated with a particular speaker. This can be achieved by confining the signal at each speaker to disjoint frequency bands or time intervals. Alternately, we can use coded sequences such that the signal due to each speaker can be extracted at the microphones and correctly attributed to the corresponding speaker. The MS TOF measurements constitute our observations, based on which we have to estimate the microphone and speaker positions.

The approach we describe is a generalization of the *trilateration* and *multilateration* techniques used in GPS positioning and other localization systems. Such systems assume that the locations of four sources are known. Based on these sources the TOF to a sensor is estimated. By trilateration a sensor's position can be determined. At least four speakers are required to find the position of an omnidirectional microphone. Knowing the distance from one speaker, the microphone can lie anywhere on a sphere centered at the speaker. With two speakers the microphone can lie on a circle, since two spheres intersect at a circle. With three we can get two points and four speakers can give a unique location. Since the estimated distances are corrupted by noise, the intersection in general need not be a unique point. Therefore we solve the problem in a least square sense by adding more speakers. We formulate the problem for the general case where the positions of both the microphones and the speakers are unknown.

Let $\mathbf{m_i}$ for $i \in [1, M]$ and $\mathbf{s_j}$ for $j \in [1, S]$ be the three dimensional vectors representing the spatial coordinates of the i^{th} microphone and j^{th} speaker, respectively. We excite one of the S speakers at a time and measure the TOF at each of the M microphones. Let TOF_{ij}^{actual} be the actual TOF for the i^{th} microphone due to the j^{th} source. Based on geometry the actual TOF can be written as (assuming a direct path),

$$TOF_{ij}^{actual} = \frac{\parallel \mathbf{m_i} - \mathbf{s_j} \parallel}{c} \tag{2.1}$$

where c the speed of sound in the acoustical medium [2] and $\parallel \; \parallel$ is the Euclidean norm. The TOF, which we estimate based on the signal captured confirms to this model only when all the sensors

[2] The speed of sound in a given acoustical medium is assumed to be constant. In air it is given by $c = (331 + 0.6T)m/s$, where T is the temperature of the medium in Celsius degrees. For improved position calibration it is beneficial to integrate a temperature sensor into the system. It is also possible to include the speed of sound as a

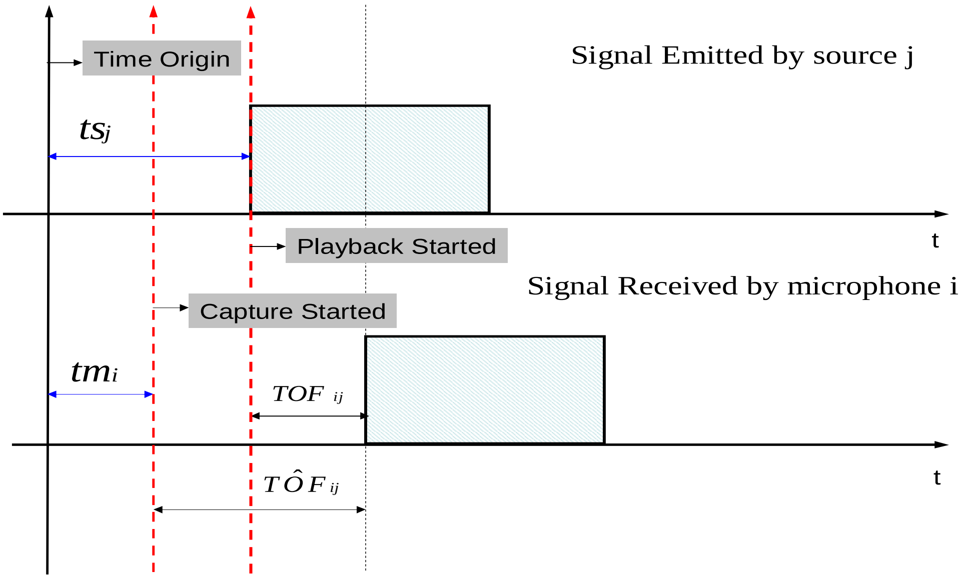


Figure 2.1: Schematic indicating the unknown emission and capture start time.

start capturing at the same instant and we know when the calibration signal was sent from the speaker. This is generally the case when we use multichannel sound cards to interface multiple microphones and speakers [3].

However in a typical distributed setup of GPCs as shown in Figure 1.3, the master starts the audio capture and playback on each of the GPCs one by one. As a result the capture starts at different instants on each GPC and also the time at which the calibration signal was emitted from each loud speaker are not known. In a distributed setting, the TOF which we measure includes both the speaker emission start time and the microphone capture start time (See Figure 2.1 where $T\hat{O}F_{ij}$ is what we measure and TOF_{ij} is what we require).

The speaker emission start time is defined as the time at which the sound is actually emitted from the speaker. This includes the time when the play back command was issued (with reference to some time origin), the network delay involved in starting the playback on a different machine (if the speaker is on a different GPC), the delay in setting up the audio buffers and also

parameter to be estimated, as in [28].

[3] For multichannel sound cards all the channels are synchronized and the time when the calibration signal was sent can be determined by doing a loop back from the output to the input. This loopback signal can be used as a reference to estimate the TOF.

the time required for the speaker diaphragm to start vibrating. The emission start time is generally unknown and depends on the particular sound card, speaker and the system state such as the processor workload, interrupts, and the processes scheduled at the given instant. The microphone capture start time is defined as the time instant at which capture is started. This includes the time when the capture command was issued, the network delay involved in starting the capture on a different machine and the delay in transferring the captured sample from the sound card to the buffers.

Let ts_j be the emission start time for the j^{th} source and tm_i be the capture start time for the i^{th} microphone with respect to some origin (see Figure 2.1). Incorporating these two the actual TOF now becomes,

$$\begin{aligned} T\hat{O}F_{ij}^{actual} &= TOF_{ij}^{actual} + ts_j - tm_i \\ &= \frac{\parallel \mathbf{m_i} - \mathbf{s_j} \parallel}{c} + ts_j - tm_i \end{aligned} \tag{2.2}$$

The origin can be arbitrary since $T\hat{O}F_{ij}^{actual}$ depends on the difference of ts_j and tm_i. We start the audio capture on each GPC one by one. We define the microphone on which the audio capture was started first as our first microphone. In practice, we set $tm_1 = 0$ i.e. the time at which the first microphone started capturing is our origin. We define all other times with respect to this origin.

If two audio input and output channels are available on a single GPC then one of the output channels can be used to play a reference signal which is RF modulated and transmitted through the air [19]. This reference signal can be captured in one of the input channels, demodulated and used to estimate $ts_j - tm_i$, since the transmission time for RF waves can be considered almost zero. Note that this assumes that all audio channels on the same I/O device are synchronized, which is generally true. However this method requires more hardware in terms of RF modulators/demodulators. The other solution is to jointly estimate the unknown source emission and capture start time along with the microphone and source coordinates. However we can eliminate the source emission start time if we use Time Difference Of Flight instead of Time Of Flight.

2.1.1 Time Difference Of Flight

The TDOF for a given pair of microphones and a speaker is defined as the time difference between the signal received by the two microphones [4]. Let $TDOF_{ikj}^{estimated}$ be the estimated TDOF between the i^{th} and the k^{th} microphone when the j^{th} source is excited. Let $TDOF_{ikj}^{actual}$ be the actual TDOF. It is given by

$$TDOF_{ikj}^{actual} = \frac{\parallel \mathbf{m_i} - \mathbf{s_j} \parallel - \parallel \mathbf{m_k} - \mathbf{s_j} \parallel}{c} \tag{2.3}$$

Including the source emission and capture start times, it becomes

$$T\hat{DOF}_{ikj}^{actual} = \frac{\parallel \mathbf{m_i} - \mathbf{s_j} \parallel - \parallel \mathbf{m_k} - \mathbf{s_j} \parallel}{c} + tm_k - tm_i \tag{2.4}$$

In the case of TDOF the source emission time is the same for both microphones and thus gets cancelled out. Therefore, by using TDOF measurements instead of TOF we can reduce the number of parameters to be estimated.

2.2 Maximum Likelihood Estimate

Assuming an additive Gaussian[5] noise model for the TDOF observations we can derive the Maximum Likelihood estimate as follows. Let Θ, be a vector of length $P \times 1$, representing all the unknown non-random parameters to be estimated (microphone and speaker coordinates and microphone capture start times). Let Γ, be a vector of length $N \times 1$, representing noisy TDOF measurements. Let $T(\Theta)$, be a vector of length $N \times 1$, representing the actual value of the observations. Then our model for the observations is $\Gamma = T(\Theta) + \eta$ where η is the zero-mean additive white Gaussian noise vector of length $N \times 1$ where each element has the variance σ_j^2. Also let us define Σ to be the $N \times N$ covariance matrix of the noise vector η. The likelihood

[4] Given M microphones and S speakers we can have $MS(M-1)/2$ TDOF measurements as opposed to MS TOF measurements. Of these $MS(M-1)/2$ TDOF measurements only $(M-1)S$ are linearly independent.

[5] We estimate the TDOF or TOF using Generalized Cross Correlation (GCC)[17]. The estimated TDOF or TOF is corrupted due to ambient noise and room reverberation. For high SNR the delays estimated by the GCC can be shown to be normally distributed with zero mean. [17].

function of Γ in vector form can be written as:

$$p(\Gamma/\Theta) = (2\pi)^{-\frac{N}{2}} \mid \Sigma \mid^{-\frac{1}{2}} \exp -\frac{1}{2}(\Gamma - T)^T \Sigma^{-1} (\Gamma - T) \tag{2.5}$$

The log-likelihood function is given by

$$\ln p(\Gamma/\Theta) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2} ln \mid \Sigma \mid -\frac{1}{2}(\Gamma - T)^T \Sigma^{-1} (\Gamma - T) \tag{2.6}$$

The ML estimate of Θ is the one which maximizes the log likelihood ratio and is given by

$$\begin{gathered} \hat{\Theta}_{ML} = \arg_{\Theta} \max F(\Theta, \Gamma) \\ F(\Theta, \Gamma) = -\frac{1}{2}[\Gamma - T(\Theta)]^T \Sigma^{-1} [\Gamma - T(\Theta)] \end{gathered} \tag{2.7}$$

In our case, Θ represents a vectorized form of the following parameters.

$$\Theta = [\Theta_m, \Theta_s, \Theta_{tm}] \tag{2.8}$$

$$\begin{gathered} \Theta_m = [mx_1, my_1, mz_1, ..., mx_M, my_M, mz_M]^T \\ \Theta_s = [sx_1, sy_1, sz_1, ..., sx_S, sy_S, sz_S]^T \\ \Theta_{tm} = [tm_1, tm_2, ..., tm_M]^T \end{gathered} \tag{2.9}$$

where mx_i, my_i, and mz_i are the x, y and z coordinates of the i^{th} microphone and sx_i, sy_i, and sz_i are the x, y and z coordinates of the i^{th} speaker. tm_i is the microphone capture start time for the i^{th} microphone. Γ and T corresponds to the estimated

Assuming that each of the TDOFs are independently corrupted by zero-mean additive white Gaussian noise of variance σ^2_{ikj} the ML estimate becomes a nonlinear least squares problem (in this case Σ is a diagonal matrix), i.e.

$$\begin{gathered} \hat{\Theta}_{ML} = \arg_{\Theta} \min[\widetilde{F}_{TDOF}(\Theta, \Gamma)] \\ \widetilde{F}_{TDOF}(\Theta, \Gamma) = \sum_{j=1}^{S} \sum_{i=1}^{M-1} \sum_{k=i+1}^{M} \frac{(TDOF^{estimated}_{ikj} - T\hat{D}OF^{actual}_{ikj})^2}{\sigma^2_{ikj}} \end{gathered} \tag{2.10}$$

In the case of TOF measurements the ML estimate can be similarly derived as above and is given by,

$$\hat{\Theta}_{ML} = \arg_{\Theta} \min[\widetilde{F}_{TOF}(\Theta, \Gamma)]$$
$$\widetilde{F}_{TOF}(\Theta, \Gamma) = \sum_{j=1}^{S} \sum_{i=1}^{M} \frac{(TOF_{ij}^{estimated} - TOF_{ij}^{actual})^2}{\sigma_{ij}^2} \tag{2.11}$$

In this case Θ also includes the speaker emission start times.

2.3 Reference Coordinate System

Since the TOF and TDOF depends on pairwise distances, any translation and rotation of the coordinate system, will also be a global minimum. In order to eliminate multiple global minima we select three arbitrary nodes to lie in a plane such that the first is at $(0, 0, 0)$, the second at $(x_1, 0, 0)$, and the third at $(x_2, y_2, 0)$. Basically we are fixing a plane so that the sensor configuration cannot be translated or rotated. In two dimensions we select two nodes to lie on a line, the first at $(0, 0)$ and the second at $(x_1, 0)$. To eliminate the ambiguity due to reflection along the Z-axis (or Y-axis in 2D) we specify one more node to lie in the positive Z-axis (or positive Y-axis in 2D). Also the reflections along the X-axis and Y-axis (for 3D) can be eliminated by assuming the nodes, which we fix, to lie on the positive side of the respective axes, i.e. $x_1 > 0$ and $y_2 > 0$.

Since the TDOF and TOF depends on time differences (i.e. $ts_j - tm_i$ in case of TOF and $tm_k - tm_i$ in case of TDOF) there are multiple global minima due to shifts in the time axis. Similar to fixing a reference coordinate system in space we introduce a reference time line by setting $tm_1 = 0$. This is needed since we are estimating the absolute source emission and capture start times[6]. Note we are only interested in the positions of the microphones and speakers. The emission and capture times are just nuisance parameters.

[6] If we are estimating the difference then we do not need a time reference. However estimating the difference introduces a lot of unnecessary parameters($O(N^2)$ parameters instead of $O(N)$ parameters.

2.4 Non-Linear Least Squares

The ML estimate for the node coordinates of the microphones and speakers is implicitly defined as the minimum of the non-linear function defined in Equation 2.10. This function has to be minimized using numerical optimization methods. Least squares problems can be solved using a general unconstrained minimization. However there exist specialized methods like the Gauss-Newton and the Levenberg-Marquardt method which are often more efficient in practice. The Levenberg-Marquardt method [8] is a popular method for solving non-linear least squares problems. It is a compromise between steepest descent and Newton's methods. The steepest descent method potentially has a very slow convergence, but can converge from any starting point. Newton's method converges fast but requires a good initial guess and computation of the inverse of the Hessian matrix. For more details on nonlinear minimization refer to [12]. Appendix A gives the non zero partial derivatives needed for the minimization routines[7]. The common problem with minimization methods is that they often get stuck in a local minima. Good initial guesses of the node locations counteract the problem.

2.5 Minimum number of microphones and speakers required

Non-linear least squares optimization requires that the total number of observations is greater than or equal to the total number of parameters to be estimated. This imposes a minimum number of microphones and speakers required for the position estimation method to work. Assuming we have M microphones and S speakers Table 2.1 summarizes the number of independent observations (N) and the number of parameters to be estimated (P) in each of the estimation procedures. In case of the TDOF based method only $(M-1)S$ out of $MS(M-1)/2$ pair of TDOF measurements for each speaker are linearly independent. Assuming M=S=K, the

[7] Many commercial software solutions are available for the Levenberg-Marquardt method such as *lsqnonlin* in MATLAB, *mrqmin* provided by Numerical Recipes in C[26] , and the MINPACK-1 routines[3]

Table 2.2 lists the minimum K required for least squares fitting.

Table 2.1: Total Number of independent observations(N) and parameters to be estimated(P) for different estimation procedures: M = Number of Microphones, S = Number of Speakers, D = Dimension.

	N	P
TOF Position Estimation	MS	$DM + DS - \frac{D(D+1)}{2}$
TDOF Position Estimation	$(M-1)S$	$DM + DS - \frac{D(D+1)}{2}$
TOF Joint Estimation	MS	$(D+1)M + (D+1)S - \frac{D(D+1)}{2} - 1$
TDOF Joint Estimation	$(M-1)S$	$(D+1)M + DS - \frac{D(D+1)}{2} - 1$

Table 2.2: Minimum value of Microphone Speaker Pairs (K) required for different estimation procedures (D=Dimension).

$K \geq$	$D = 2$	$D = 3$
TOF Position Estimation	3	5
TDOF Position Estimation	5	6
TOF Joint Estimation	6	7
TDOF Joint Estimation	6	7

Chapter 3

Closed Form approximate Solution

The common problem with minimization methods is that they often get stuck in a local minima. They do not converge unless we have a very good starting point. In this chapter we make some approximations to get closed form solutions to the microphone and speaker positions and the capture start times which can be used as a initial guess for the nonlinear minimization routine.

3.1 Initial Guess for capture and emission start times

Consider two laptops i and j each having one microphone and one speaker. For these two laptops we can measure $T\hat{O}F_{ii}$, $T\hat{O}F_{jj}$, $T\hat{O}F_{ij}$ and $T\hat{O}F_{ji}$. Assuming no noise these are related to the actual TOF as follows:

$$\begin{aligned}
T\hat{O}F_{ii} &= TOF_{ii} + ts_i - tm_i \\
T\hat{O}F_{jj} &= TOF_{jj} + ts_j - tm_j \\
T\hat{O}F_{ij} &= TOF_{ij} + ts_j - tm_i \\
T\hat{O}F_{ji} &= TOF_{ji} + ts_i - tm_j
\end{aligned} \tag{3.1}$$

Assuming sufficient closeness between the microphone and speaker on the same laptop compared to the distance between two laptops, the following approximations can be made.

$$\begin{aligned} TOF_{ii} \approx TOF_{jj} &\approx 0 \\ TOF_{ij} &\approx TOF_{ji} \end{aligned} \tag{3.2}$$

Substituting we have the following equations:

$$\begin{aligned} T\hat{O}F_{ii} &\approx ts_i - tm_i \\ T\hat{O}F_{jj} &\approx ts_j - tm_j \\ T\hat{O}F_{ij} &\approx TOF_{ij} + ts_j - tm_i \\ T\hat{O}F_{ji} &\approx TOF_{ij} + ts_i - tm_j \end{aligned} \tag{3.3}$$

From the above equations we can solve for TOF_{ij} as:

$$TOF_{ij} \approx \frac{(T\hat{O}F_{ij} + T\hat{O}F_{ji}) - (T\hat{O}F_{ii} + T\hat{O}F_{jj})}{2} \tag{3.4}$$

Also we can solve for the microphone capture start time and the source emission start time as follows:

$$\begin{aligned} ts_i &\approx T\hat{O}F_{ii} + tm_i \\ tm_j &\approx \frac{(T\hat{O}F_{ij} - T\hat{O}F_{ji}) + (T\hat{O}F_{ii} - T\hat{O}F_{jj})}{2} + tm_i \end{aligned} \tag{3.5}$$

Considering the time when the capture on the first microphone is started as zero (i.e. $tm_1 = 0$), we can solve for all the other microphone capture start times and the speaker emission start times. Note that all the above equations are true only approximately. Their values have to be refined further using the ML estimation procedure.

3.2 Initial Guess for microphone and speaker positions

Given the pairwise Euclidean distances between N nodes their relative positions can be determined by means of metric Multidimensional Scaling (MDS) [32]. MDS is popular in psychology and denotes a set of data-analysis techniques for the analysis of proximity data on a set of stimuli for revealing the hidden structure underlying the data [31]. The proximity data refers to some measure of pairwise dissimilarity. Given a set of N stimuli along with their pairwise dissimilarities p_{ij}, MDS places the N stimuli as points in a multidimensional space, such that the distances between any two points are a monotonic function of the corresponding dissimilarity. MDS is widely used to visually study the structure in proximity data. [31] describes an experiment where MDS is used to reveal some of the perceptual dimensions that people might use for face similarity judgement tasks.

If proximity data are based on the Euclidean distances, then classical metric MDS [32] can exactly recreate the configuration. Given a set of N GPCs, let X be a $N \times 3$ matrix where each row represents the 3D coordinates of each GPC. Then the $N \times N$ matrix $B = XX^T$ is called the dot product matrix. By definition, B is a symmetric positive definite matrix, so the rank of B (i.e the number of positive eigen values) is equal to the dimension of the datapoints i.e. 3 in this case. Also based on the rank of B we can find whether the GPCs are on a plane or distributed in 3D. Starting with a matrix B (possibly corrupted by noise), it is possible to factor it to get the matrix of coordinates X. One method to factor B is to use singular value decomposition (SVD) [26], i.e., $B = U\Sigma U^T$ where Σ is a $N \times N$ diagonal matrix of singular values. The diagonal elements are arranged as $s_1 \geq s_2 \geq s_r > s_{r+1} = ... = s_N = 0$, where r is the rank of the matrix B. The columns of U are the corresponding singular vectors. We can write $X^{'} = U\Sigma^{1/2}$. From $X^{'}$ we can take the first three columns to get X. If the elements of B are exact (i.e., they are not corrupted by noise), then all the other columns are zero. It can be shown that SVD factorization minimizes the matrix norm $\parallel B - XX^T \parallel$.

In practice, we can estimate the distance matrix D, where the ij^{th} element is the Euclidean distance between the i^{th} and the j^{th} GPC. This distance matrix D must be converted into a dot product matrix B before MDS can be applied. We need to choose some point as the origin of our coordinate system in order to form the dot product matrix. Any point can be selected as the origin, but Togerson [32] recommends the centroid of all the points. If the distances have random errors then choosing the centroid as the origin will minimize the errors as they tend to cancel each other. We can obtain the dot product matrix using the cosine law which relates the distance between two vectors to their lengths and the cosine of the angle between them. Refer to Appendix B for a detailed derivation of how to convert the distance matrix to the scalar product matrix.

3.2.1 Multidimensional Scaling with clustering

In our case of M microphones and S speakers we cannot use MDS directly because we cannot measure all the pairwise distances. We can measure the distance between each speaker and all the microphones. However we cannot measure the distance between two microphones or two speakers. In order to apply MDS, we cluster microphones and speakers, which are close together. Based on the approximation discussed in the previous section, the distance d_{ij} between the i^{th} and j^{th} GPC is given by

$$d_{ij} \approx \frac{c\,(T\hat{O}F_{ij} + T\hat{O}F_{ji} - T\hat{O}F_{ii} - T\hat{O}F_{jj})}{2} \tag{3.6}$$

where c is the speed of the sound.

The position estimate from MDS is arbitrary with respect to the centroid and the orientation and is converted into the reference coordinate system described in Section 2.3. The approximate locations of the GPCs are slightly perturbed to get the initial guess for the microphone and speaker locations.

3.3 Final Algorithm

Figure 3.1 summarizes the algorithm.

Say we have M microphones and S speakers

- **STEP 1**: *Measure the $M \times S$ Time Of Flight ($T\hat{O}F$) matrix.*

- **STEP 2**:

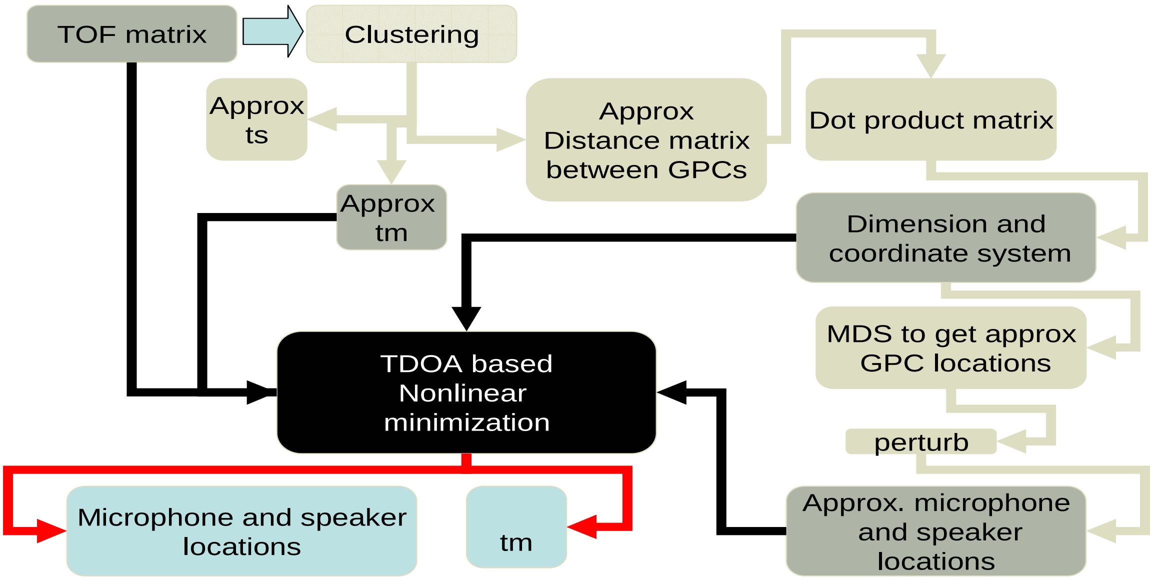

TOF matrix
Clustering
Approx ts
Approx Distance matrix between GPCs
Dot product matrix
Approx tm
Dimension and coordinate system
MDS to get approx GPC locations
TDOA based Nonlinear minimization
perturb
Microphone and speaker locations
tm
Approx. microphone and speaker locations

Figure 3.1: Flow chart of the complete algorithm.

- *Form the approximate distance matrix D. (Equation 3.6)*
- *Assume $tm_1 = 0$ (microphone on which capture was started first) and get the approximate microphone capture and speaker emission start times. (Equation 3.5)*
- *Convert the distance matrix D to the dot product matrix B (Appendix I). Find the rank of B to determine whether the GPCs are in 2D or 3D.*

- **STEP 3**: *Form a reference coordinate system*
  - *If 3D select three nodes: The first one as the origin, the second to define the x-axis and the third to form the xy-plane. Also select a fourth node to represent the positive z-axis.*
  - *If 2D select two nodes: The first one as the origin, the second to define the x-axis. Also select a third node to represent the positive y-axis.*

- **STEP 4**:
  - *Get the approximate positions of the GPCs using metric Multidimensional Scaling.*
  - *Translate, rotate and mirror to the coordinate system choosen.*

   - *Slightly perturb the coordinates to get approximate initial guess for the microphone and speaker coordinates.*

- **STEP 5**: *Minimize the TDOF based error function using the Levenberg-Marquardat method to get the final positions of the microphones and speakers. Use the approximate positions and the capture start times as the initial guess.*

Figure 3.2 shows an example with 10 laptops each having one microphone and one speaker. The actual locations of the sensors and actuators are shown as 'x'. The '*'s are the approximate GPC locations as determined by MDS. As can be seen the MDS results are very close to the microphone and speaker locations. The estimated locations are further improved in STEP 3 and marked as 'o's.

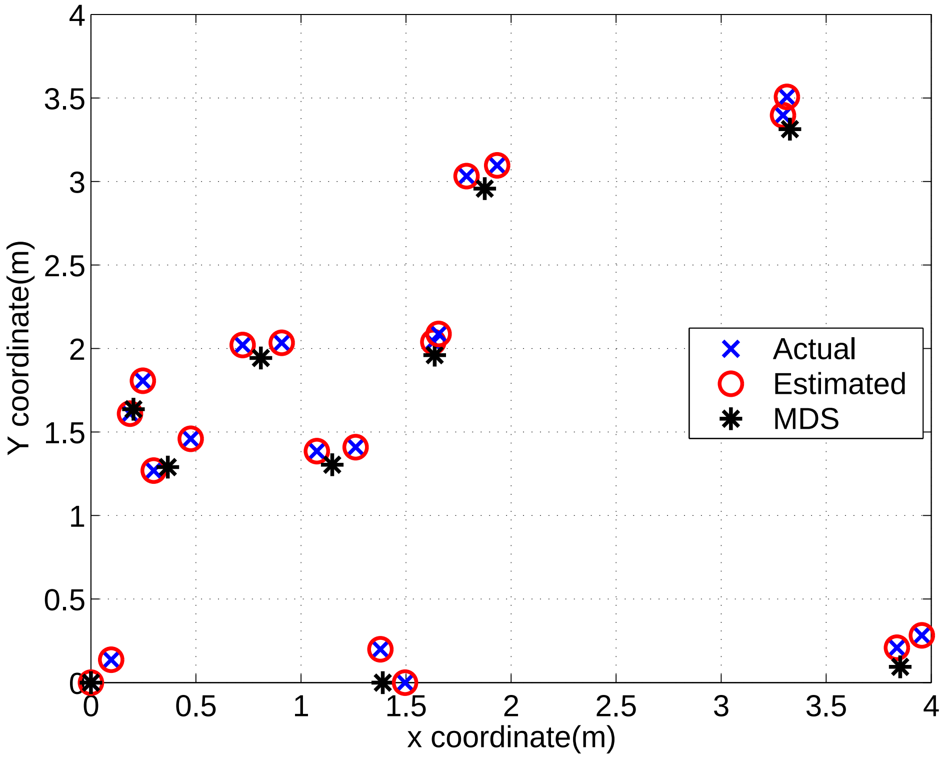


Figure 3.2: Results of Multidimensional Scaling for a network consisting of 10 GPCs each having one microphone and one speaker.

Chapter 4

Estimator Performance

The properties of the ML estimator can be studied in terms of the estimator bias and error covariance matrix. The bias and error variance depends on the noise variance, the number of microphones and speakers and the geometry of the setup. One way to study it is to do extensive Monte Carlo Simulations for various geometries and different number of nodes. However if we get an analytical expression for the bias and the variance of the estimator then these simulation studies can be carried out quickly and the estimator can be studied in depth.

The ML estimate for the microphone and speaker positions is defined implicitly as the minimum of a certain error function. Hence it is not possible to get exact analytical expressions for the mean and the variance. However, by using the implicit function theorem and the Taylor series it is possible to derive approximate expressions for the mean and variance of implicitly defined estimators [10, 11]. In this section we derive the approximate expressions for both the mean and variance of the estimators. We could have derived the Cramér-Rao bound which gives the lower bound on the error covariance matrix of any unbiased estimator. However since we cannot determine whether our estimator is unbiased, we cannot use the Cramér-Rao bound for unbiased estimators. However, we also derive the Cramér-Rao bound assuming our estimator is unbiased. It turns out be to same as our approximate covariance matrix expression.

4.1 Notation

Let Θ, be a vector of length $P \times 1$, representing all the unknown non-random parameters to be estimated. Let Γ, be a vector of length $N \times 1$, representing our noisy measurements. Let $T(\Theta)$, be a vector of length $N \times 1$, representing the actual value of the observations.

$$\begin{aligned} \Theta &= [\theta_1, \theta_2, ..., \theta_P]^T \\ \Gamma &= [\gamma_1, \gamma_2, ..., \gamma_N]^T \\ T(\Theta) &= [t_1, t_2, ..., t_N]^T \end{aligned} \tag{4.1}$$

Then our model for the observations was $\Gamma = T(\Theta) + \eta$ where η is the zero-mean additive white Gaussian noise vector of length $N \times 1$ where each element has the variance σ_j^2. Also let us define Σ to be the $N \times N$ covariance matrix of the noise vector η.

The ML estimate of Θ is the one which maximizes the log likelihood ratio and is given by

$$\begin{aligned} \hat{\Theta_{ML}} &= \arg_{\Theta} \max F(\Theta, \Gamma) \\ F(\Theta, \Gamma) &= -\frac{1}{2}[\Gamma - T(\Theta)]^T \Sigma^{-1} [\Gamma - T(\Theta)] \end{aligned} \tag{4.2}$$

4.2 Vector Derivatives

In further derivations we need the first and second derivatives of Equation 4.2 with respect to Θ and Γ. In this section we specify the vector derivative notation we use and the corresponding derivatives of $F(\Theta, \Gamma)$.

The $P \times 1$ column gradient operator $\bigtriangledown_{\Theta}$ is defined as

$$\nabla_{\Theta} F(\Theta, \Gamma) = [\frac{\partial F(\Theta, \Gamma)}{\partial \theta_1}, \frac{\partial F(\Theta, \Gamma)}{\partial \theta_2}, ..., \frac{\partial F(\Theta, \Gamma)}{\partial \theta_P}]^T \tag{4.3}$$

Similarly the $N \times 1$ column gradient operator ∇_Γ with respect to Γ is defined as

$$\nabla_\Gamma F(\Theta,\Gamma) = [\frac{\partial F(\Theta,\Gamma)}{\partial \gamma_1}, \frac{\partial F(\Theta,\Gamma)}{\partial \gamma_2}, ..., \frac{\partial F(\Theta,\Gamma)}{\partial \gamma_N}]^T \tag{4.4}$$

We also define the following four second derivative operators: the $P \times P$ operator $\nabla_\Theta\nabla_\Theta$, $N \times N$ operator $\nabla_\Gamma\nabla_\Gamma$, $N \times P$ operator $\nabla_\Gamma\nabla_\Theta$ and $P \times N$ operator $\nabla_\Theta\nabla_\Gamma$, which are defined as below

$$\begin{aligned}
\nabla_\Theta\nabla_\Theta F(\Theta,\Gamma) &= \nabla_\Theta[\{\nabla_\Theta F(\Theta,\Gamma)\}^T] \\
\nabla_\Gamma\nabla_\Gamma F(\Theta,\Gamma) &= \nabla_\Gamma[\{\nabla_\Gamma F(\Theta,\Gamma)\}^T] \\
\nabla_\Gamma\nabla_\Theta F(\Theta,\Gamma) &= \nabla_\Gamma[\{\nabla_\Theta F(\Theta,\Gamma)\}^T] \\
\nabla_\Theta\nabla_\Gamma F(\Theta,\Gamma) &= \nabla_\Theta[\{\nabla_\Gamma F(\Theta,\Gamma)\}^T]
\end{aligned} \tag{4.5}$$

Using the generalized chain rule it can be shown that the vector derivatives are as follows

$$\begin{aligned}
\nabla_\Theta F(\Theta,\Gamma) &= J^T\Sigma^{-1}(\Gamma - T(\Theta)) \\
\nabla_\Gamma F(\Theta,\Gamma) &= -\Sigma^{-1}(\Gamma - T(\Theta)) \\
\nabla_\Theta\nabla_\Theta F(\Theta,\Gamma) &= -J^T\Sigma^{-1}J \\
\nabla_\Gamma\nabla_\Gamma F(\Theta,\Gamma) &= -\Sigma^{-1} \\
\nabla_\Gamma\nabla_\Theta F(\Theta,\Gamma) &= \Sigma^{-1}J \\
\nabla_\Theta\nabla_\Gamma F(\Theta,\Gamma) &= J^T\Sigma^{-1}
\end{aligned} \tag{4.6}$$

where J is a $N \times P$ matrix of partial derivatives of $T(\Theta)$ called the *Jacobian* of $T(\Theta)$.

$$[J]_{ij} = \frac{\partial t_i(\Theta)}{\partial \theta_j} \tag{4.7}$$

Refer to Appendix A for the individual derivatives of the *Jacobian* matrix.

4.3 Estimator Covariance

In this section we use the Taylor series expansion and the implicit function theorem to derive an approximate expression for the covariance of the implicity defined estimator. The ML estimate of Θ is the one which maximizes the log likelihood ratio defined in Equation 4.2. The maximum can be found by setting the first derivative to zero i.e.

$$\nabla_\Theta F(\Theta, \Gamma) \mid_{\Theta=\hat{\Theta}} = \mathbf{0} \tag{4.8}$$

where $\mathbf{0}$ is a zero column vector of length P. The implicit function theorem guarantees that Equation 4.8 implicitly defines a vector valued function $\hat{\Theta} = h(\Gamma) = [h_1(\Gamma), h_1(\Gamma), ..., h_P(\Gamma)]^T$ that maps the observation vector Γ to the parameter vector $\hat{\Theta}$. Equation 4.8 can be written as

$$\nabla_\Theta F(\Theta, \Gamma) \mid_{\Theta=h(\Gamma)} = \mathbf{0} \tag{4.9}$$

$$\nabla_\Theta F(h(\Gamma), \Gamma) = \mathbf{0} \tag{4.10}$$

However it is not possible to find an analytical expression for $h(\Gamma)$. But we can approximate the covariance using the first-order Taylor series expansion for $h(\Gamma)$. Let Γ_m be the mean of Γ. Then expanding $h(\Gamma)$ around Γ_m we get

$$h(\Gamma) \approx h(\Gamma_m) + [\nabla_\Gamma h(\Gamma)^T \mid_{\Gamma=\Gamma_m}]^T(\Gamma - \Gamma_m) \tag{4.11}$$

where $\nabla_\Gamma = [\frac{\partial}{\partial\gamma_1}, \frac{\partial}{\partial\gamma_2}, ..., \frac{\partial}{\partial\gamma_N}]^T$ is a $N \times 1$ column gradient operator. Taking the covariance on both sides yields

$$Cov(h(\Gamma)) \approx [\nabla_\Gamma h(\Gamma)^T \mid_{\Gamma=\Gamma_m}]^T Cov(\Gamma)[\nabla_\Gamma h(\Gamma)^T \mid_{\Gamma=\Gamma_m}] \tag{4.12}$$

Note we do not know $h(\Gamma)$. Differentiating Equation 4.10 with respect to Γ and evaluating at Γ_m yields

$$\nabla_\Theta\nabla_\Theta F(h(\Gamma), \Gamma)[\nabla_\Gamma h(\Gamma)^T]^T + \nabla_\Theta\nabla_\Gamma F(h(\Gamma), \Gamma) \mid_{\Gamma=\Gamma_m} = \mathbf{0}$$

$$\nabla_\Theta\nabla_\Theta F(h(\Gamma_m), \Gamma_m)[\nabla_\Gamma h(\Gamma_m)^T]^T + \nabla_\Theta\nabla_\Gamma F(h(\Gamma_m), \Gamma_m) = \mathbf{0} \tag{4.13}$$

Assuming $\nabla_\Theta\nabla_\Theta F(h(\Gamma_m), \Gamma_m)$ is invertible we can write

$$[\nabla_\Gamma h(\Gamma_m)^T]^T = -[\nabla_\Theta\nabla_\Theta F(h(\Gamma_m), \Gamma_m)]^{-1}\nabla_\Theta\nabla_\Gamma F(h(\Gamma_m), \Gamma_m) \tag{4.14}$$

Substituting from Equation 4.6 we get

$$[\nabla_\Gamma h(\Gamma_m)^T]^T = -[-J^T\Sigma^{-1}J]^{-1}J^T\Sigma^{-1} \tag{4.15}$$

Using this in the covariance expression in Equation 4.12, we arrive at

$$
\begin{aligned}
Cov\hat{\Theta} = Cov(h(\Gamma)) &= [J^T\Sigma^{-1}J]^{-1}J^T\Sigma^{-1}\Sigma[J^T\Sigma^{-1}]^T\{[J^T\Sigma^{-1}J]^T\}^{-1} \\
&= [J^T\Sigma^{-1}J]^{-1}J^T\Sigma^{-1}\Sigma\Sigma^{-1}J\{[J^T\Sigma^{-1}J]^T\}^{-1} \\
&= [J^T\Sigma^{-1}J]^{-1}[J^T\Sigma^{-1}J][J^T\Sigma^{-1}J]^{-1} \\
&= [J^T\Sigma^{-1}J]^{-1} \qquad (4.16)
\end{aligned}
$$

$$
Cov\hat{\Theta} = [J^T\Sigma^{-1}J]^{-1} \qquad (4.17)
$$

4.4 Estimator Mean

Taking the expectation of the first order Taylor series expansion in Equation 4.11

$$
E(h(\Gamma)) \approx h(\Gamma_m) = h(T(\Theta)) \qquad (4.18)
$$

We have made use of the fact that $\Gamma_m = T(\Theta)$. We see that the mean is the value given by the estimation procedure when applied to the actual noise free measurements T. It is also possible to get the mean using the second order Taylor series expansion, but it involves third order derivatives and generally we cannot get simple form as in Equation 4.17.

4.5 Cramér-Rao Bound

The Cramér-Rao bound gives a lower bound on the variance of *any* unbiased estimate [33]. It does not depend on the particular estimation method used. In this section, we derive the Cramér-Rao bound (CRB) assuming our estimator is unbiased. The variance of any unbiased estimator $\hat{\Theta}$ of Θ is bounded as [33]

$$
E\left[(\hat{\Theta}-\Theta)(\hat{\Theta}-\Theta)^T\right] \geq F^{-1}(\Theta) \qquad (4.19)
$$

where $F(\Theta)$ is called the Fischer's Information matrix and is given by

$$
F(\Theta) = E\left\{[\nabla_\Theta \ln p(\Gamma/\Theta)]\,[\nabla_\Theta \ln p(\Gamma/\Theta)]^T\right\} \qquad (4.20)
$$

The derivative of the log-likelihood function can be found using the generalized chain rule and is given by

$$\nabla_\Theta \ln p(\Gamma/\Theta) = J^T \Sigma^{-1}(\Gamma - T) \tag{4.21}$$

where J is the *Jacobian*. Substituting this in Equation 4.20 and taking the expectation the Fishers Information matrix is,

$$F = J^T \Sigma^{-1} J \tag{4.22}$$

$$Cov\hat{\Theta} \geq [J^T \Sigma^{-1} J]^{-1} \tag{4.23}$$

Note that this expression is the same as the approximate covariance of the estimator derived in the previous section.

4.6 Rank of the Fischer Information Matrix

If we assume $\Sigma = \sigma^2 I$, i.e. the noise components are independent, then the covariance matrix can be simplified as

$$Cov[\hat{\Theta}] = \frac{1}{\sigma^2}[J^T J]^{-1} = F^{-1} \tag{4.24}$$

where $F = \frac{1}{\sigma^2} J^T J$. If we assume that all the microphone and source locations are unknown, F is rank deficient and hence not invertible. This is because the solution to the ML estimation problem as formulated is not invariant to rotation and translation. In order to make the Fisher Information matrix invertible we remove the rows and columns corresponding to the known parameters.

Theorem : $rank(F) < P$

Proof : $rank(F) = rank(J^T J) = rank(J) \leq min(N, P)$. In our case we have always sufficient number of microphones and speakers such that $P < N$ i.e. the number of parameters to be estimated is always less than the number of observations. Hence $rank(F) \leq P$. Since rank of F is equal to rank of J rank of F will be strictly less than P only when the columns of J are linearly dependent. J is a $N \times P$ matrix of partial derivatives of $T(\Theta)$ called the *Jacobian* of $T(\Theta)$. Each row of J corresponds to the derivatives of TOF_{ij} with respect to all the unknown parameters. From Appendix A it can be seen that for TOF_{ij} the only non-zero derivatives are with respect to mx_i, my_i, mz_i, sx_j, sy_j and sz_j. Also all these derivatives sum up to zero. Therefore each row of J sums up to zero. Hence the columns of J are linearly dependent.

4.7 Effect of the Nuisance parameters

The speaker emission start time, the microphone capture start time and the loudspeaker coordinates can be considered as the nuisance parameters since we are interested only in the microphone coordinates. We can split J the Jacobian into two blocks, one involving the parameters which we are interested i.e the microphone coordinates and the other nuisance parameters. Let Θ_m represent the parameters of interest and let Θ_s be the nuisance parameters.

$$J = [J_m J_s] \; where \; J_m = \left[\frac{\partial T(\Theta)}{\partial \Theta_m}\right] \; J_s = \left[\frac{\partial T(\Theta)}{\partial \Theta_s}\right] \tag{4.25}$$

So now,

$$F = \frac{1}{\sigma^2}\begin{bmatrix} J_m^T J_m & J_m^T J_s \\ J_s^T J_m & J_s^T J_s \end{bmatrix} \tag{4.26}$$

Using the formula for the inverse of a block matrix we can write

$$F^{-1} = \sigma^2 \begin{bmatrix} F_{mm}^{-1} & F_{ms} \\ F_{ms} & F_{ss}^{-1} \end{bmatrix} \tag{4.27}$$

where

$$\begin{aligned} F_{mm} &= J_m^T \left[I - J_s(J_s^T J_s)^{-1} J_s^T\right] J_m \\ F_{ss} &= J_s^T \left[I - J_m(J_m^T J_m)^{-1} J_m^T\right] J_s \\ F_{ms} &= -(J_m^T J_m)^{-1}(J_m^T J_s) F_{ss}^{-1} \\ F_{sm} &= -F_{ss}^{-1}(J_s^T J_m)(J_m^T J_m)^{-1} \end{aligned} \tag{4.28}$$

where I is the Identity matrix of appropriate dimensions. So the first term of the block matrix which gives a bound on the parameters of interest (i.e. microphone coordinates) can be written as

$$F_{mm}^{-1} = \sigma^2 \left\{J_m^T \left[I - J_s(J_s^T J_s)^{-1} J_s^T\right] J_m\right\}^{-1} \tag{4.29}$$

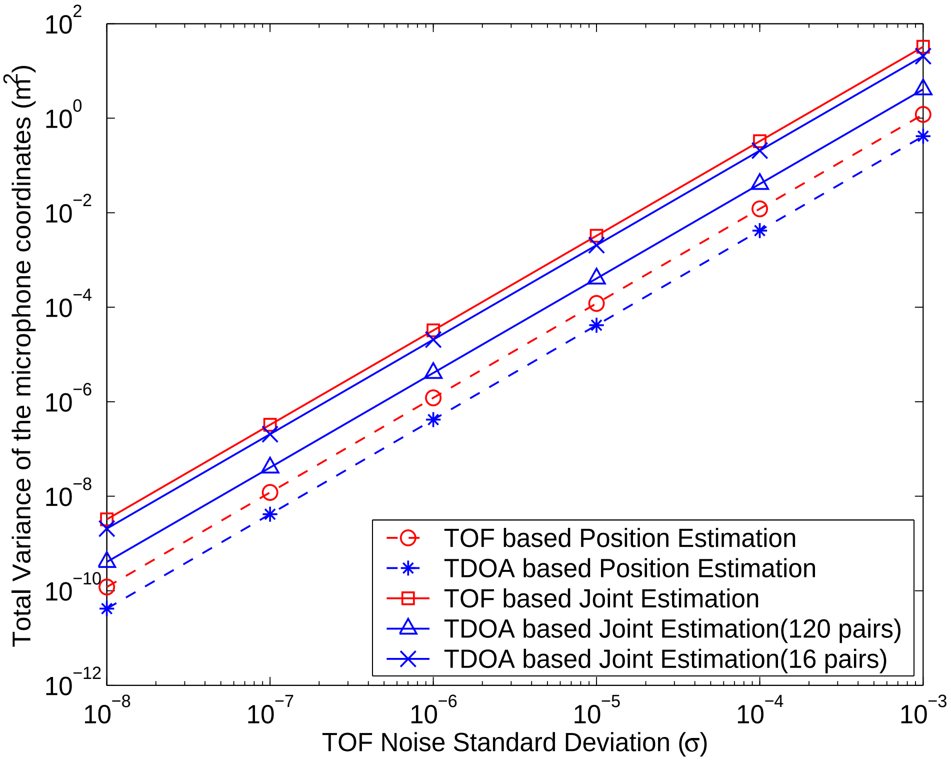


Figure 4.1: Cramér-Rao bound on the total variance of the unknown microphone coordinates as a function of TOF noise standard deviation σ for different estimation procedures. For the TDOF-based method the noise variance was taken as twice that of the TOF variance. The network had a total of 16 microphones and 16 speakers.

The diagonal terms of F_{mm}^{-1} represents the error variance for estimating each of the parameters in Θ_m. In the next few sections we explore the dependency of the error variance on different parameters.

Figure 4.1 shows Cramér-Rao bound on the total variance of the unknown microphone coordinates as a function of TOF noise standard deviation σ for a sensor network consisting of 16 microphones and 16 speakers, for different estimation procedures. In order to do a fair comparison, the corresponding TDOF noise variance was approximated to be twice the corresponding TOF noise variance. In the TOF case only one signal was degraded due to noise and reverberation while the other was the reference signal. In case of TDOF both the signals are degraded.

The effect of the nuisance parameters on the Cramér Rao bound can be seen from Figure 4.1, where the total error variance in the microphone coordinates is plotted against the noise standard deviation σ for both normal position estimation and joint position estimation. For both the TOF and TDOF approaches the joint estimation results in a higher variance which is due to the extra nuisance parameters. Among TOF and TDOF approaches TOF has more number of nuisance parameters and hence it has a higher variance than the TDOF approach. Another point to be noted is that in the TDOF approach we need not use all the $M(M-1)/2$ pairwise TDOF measurements. However as we use more and more TDOF measurements the variance decreases as can be seen in Figure 4.1.

4.8 Increasing the number of GPCs

As the number of nodes increases in the network, the CRB on the covariance matrix decreases. The more microphones and speakers in the network, the smaller the error in estimating their positions. Figure 4.2(a) shows the 95% uncertainty ellipses for a regular two dimensional array consisting of 9 microphones and 9 speakers, for both the TOF and the TDOF-based joint estimation procedures. We fixed the position of one microphone and the x coordinate of one speaker. For the fixed speaker only the variance in y direction is shown since the x coordinate is fixed. For TOF-based method the noise variance was assumed to be 10^{-9} in order to properly visualize the uncertainty ellipses. In order to give a fair comparison, a noise variance of 2×10^{-9} was assumed for the TDOF-based method. Figure 4.2(b) shows the corresponding 95% uncertainty ellipses for a two dimensional array consisting of 25 microphones and 25 speakers. It can be seen that as the number of sensors in the network increases the size of the uncertainty ellipses decreases.

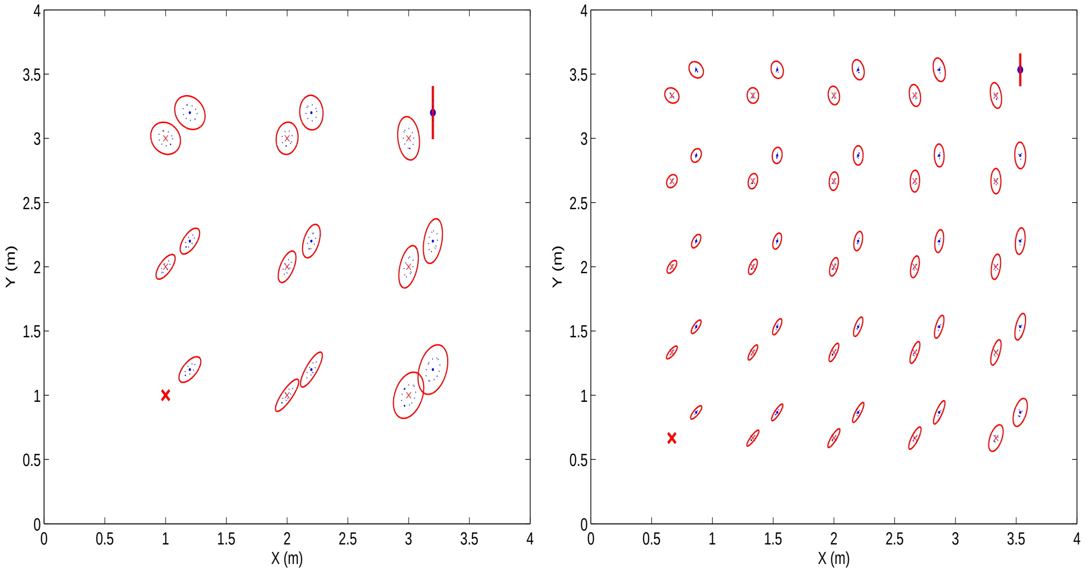
Intuitively this can be explained as follows: Let there be a total of n nodes in the network whose coordinates are unknown. Then we have to estimate a total of $3n$ parameters. The total number of TOF measurements available is however $n^2/4$ (assuming that there are $n/2$ microphones and $n/2$ speakers). So if the number of unknown parameters increases as $O(n)$, the number of available measurements increases as $O(n^2)$. The linear increase in the number of unknown parameters, is compensated by the quadratic increase in the available measurements, which suggests that the uncertainty per unknown variable will decrease.

4.9 How to select a coordinate system?

The geometry of the network plays an important role in CRB. It is possible to analyze how to place the sensors in order to achieve a lower CRB. In an ad-hoc network, however, such analysis is of little benefit. In our formulation we assumed that we know the positions of a certain number of nodes, i.e we fix three of the nodes to lie in the x-y plane. The CRB depends on which of the sensor nodes are assumed to have known positions. Figure 4.3 shows the 95% uncertainty ellipses for a regular two dimensional array containing 25 microphones and 25 speakers for different positions of the known nodes. In Figure 4.3(a) the two known nodes are at one corner of the grid. It can be seen that the uncertainty ellipse becomes wider as you move away form the known nodes. The uncertainty in the direction tangential to the line joining the sensor node and the center of the known nodes is much larger than along the line. The same can be seen in Figure 4.3(b) where the known nodes are at the center of the grid. The reason for this can be explained for a simple case where we know the locations of two speakers as shown in Figure 4.3(d). Each circular band represents the uncertainty in the distance estimation. The intersection of the two annuli corresponding to the two speakers gives the uncertainty region for the position of the sensor. As can be seen for nodes far away from the two speakers the region widens because of the decrease in the curvature. It is beneficial if the known nodes are on the edges of the network and as far away from each other as possible. In Figure 4.3(c) the known sensor nodes are on the edges of the network. As can be seen there is a substantial reduction in the dimensions of the uncertainty ellipses. In order to minimize the error due to Gaussian noise we should choose the three reference nodes (in 3D) as far as possible. In practice, using the TOF matrix we can choose three nodes such that the area of the triangle formed by these three nodes is maximum. In this way we can dynamically adapt our coordinate system to minimize the error even though the array geometry may change drastically.

4.10 Monte Carlo Simulation Results

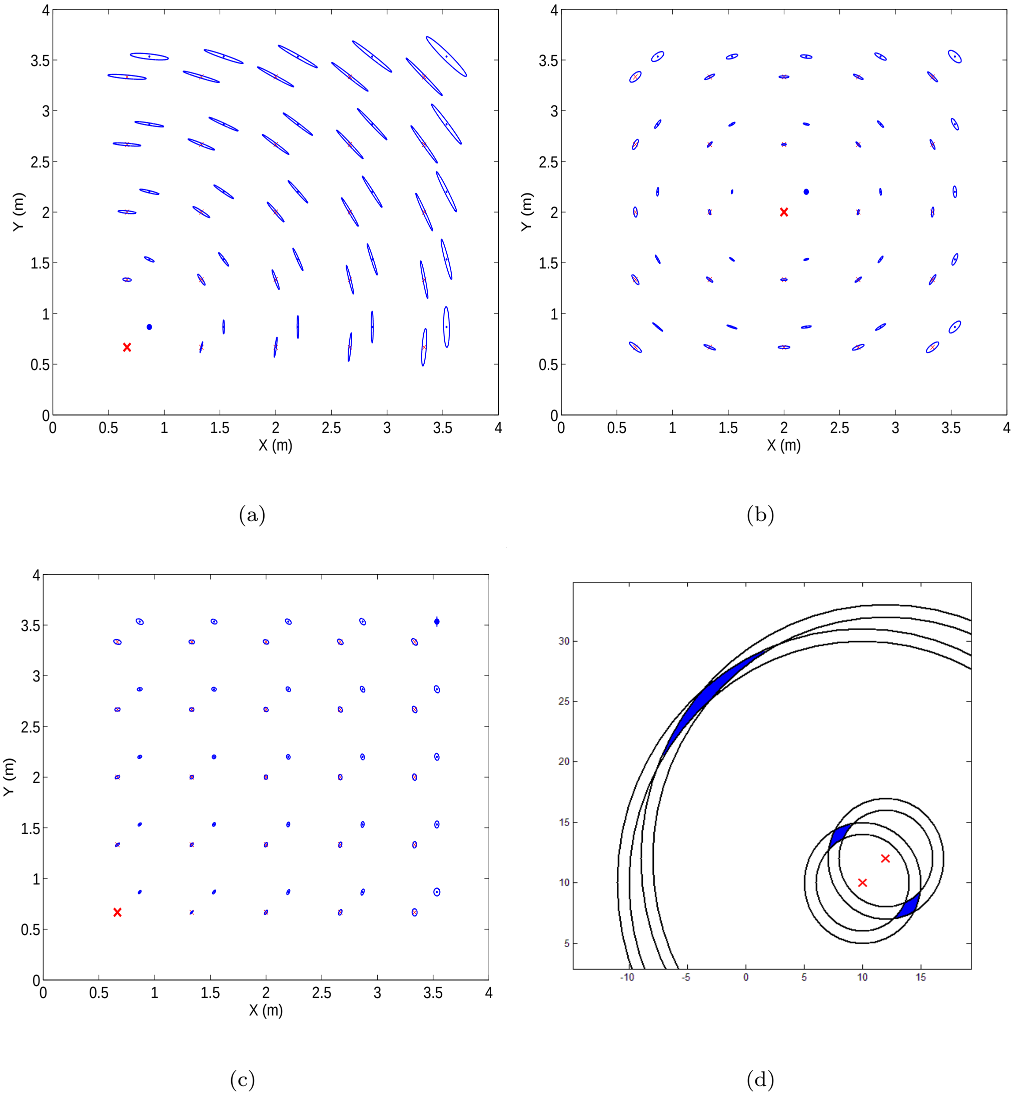
We performed a series of Monte Carlo simulations to compare the performance of the different estimation procedures. 16 microphones and 16 speakers were randomly selected to lie in a room of dimensions $4.0m \times 4.0m \times 4.0m$. The speaker was chosen to be close to the microphone in order to simulate a typical laptop. Based on the geometry of the setup the actual TOF between each speaker and microphones was calculated and then corrupted with zero mean additive white Gaussian noise of variance σ^2 in order to model the room ambient noise and reverberation. The TOF matrix was also corrupted by known systematic errors, i.e. a known microphone emission capture start time and speaker emission start time was added. The Levenberg-Marquardt method was used as the minimization routine. For each noise variance σ^2, the results were averaged over 2000 trials. Figure 4.4(a) and Figure 4.4(b) show the total variance and the total bias (sum of all the biases in each parameter) of all the unknown microphone coordinates plotted against the noise standard deviation σ for both the TOF and the TDOF-based approach. The results are shown both for position estimation and the Joint position and start times estimation procedures. The Cramér Rao bound for the TDOF-based joint estimation procedure is also shown. Since we corrupted the TOF with a systematic errors, the position estimation procedure shows a very high variance and a correspondingly high bias. Hence when the TOFs are corrupted by systematic errors we need to do joint estimation of the positions as well as the nuisance parameters. Even though theoretically the TDOF-based joint estimation procedure has the least variance, experimentally all the joint estimation procedures showed the same variance. The estimator is unbiased for low noise variances.



(a)

(b)

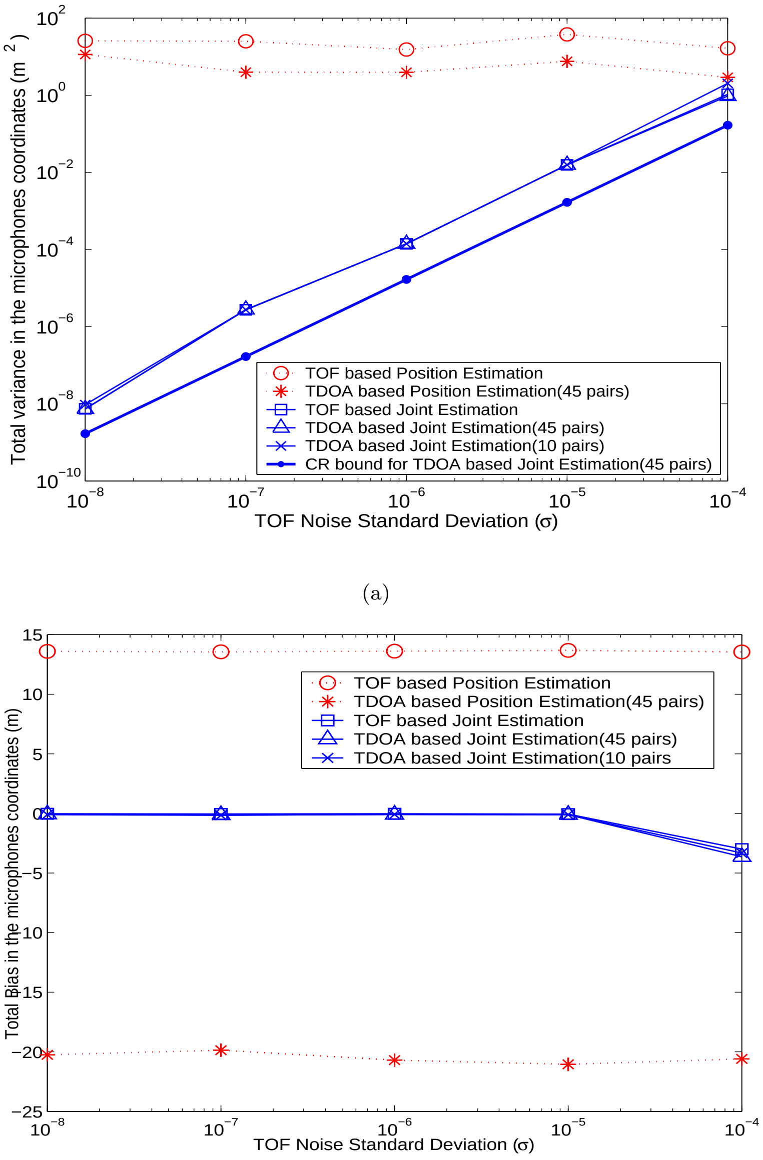
Figure 4.2: 95% uncertainty ellipses for a regular 2 dimensional array of (a) 9 speakers and 9 microphones, (b) 25 speakers and 25 microphones. Noise variance in both cases is $\sigma^2 = 10^{-9}$ for the TOF-based method and $\sigma^2 = 2 \times 10^{-9}$ for the TDOF-based method. The microphones are represented as crosses $(\times)$ and the speakers as dots (.). The position of one microphone and the x coordinate of one speaker is assumed to be known (shown in bold). The solid and dotted ellipses are the uncertainty ellipses for the estimation procedure using the TOF and TDOF-based method respectively.



(a) (b)

(c) (d)

Figure 4.3: 95% uncertainty ellipses for a regular 2 dimensional array of 25 microphones and 25 speakers for different positions of the known microphone and for different x coordinates of the known speaker. In (a) and (b) the known nodes are close to each other and in (c) they are spread out one at each corner of the grid. The microphones are represented as crosses (×) and the speakers as dots (.). Noise variance in all cases was $\sigma^2 = 10^{-9}$. (d) Schematic to explain the shape of uncertainty ellipses. 50 TDOF pairs were used for the estimation procedure.



(a)

(b)

Figure 4.4: (a) The total variance and (b) total bias of all the microphone coordinates for increasing TOF noise standard deviation σ. The sensor network consisted of 16 microphones and 16 speakers. The results are shown for both the TOF and TDOF-based Position and Joint Estimation. The Cramér Rao bound for the TDOF based Joint Estimation is also plotted. For the TDOF-based method the noise variance was taken as twice that of the TOF variance.

Chapter 5

Implementation Details

In this section we discuss some of the practical issues of our real-time implementation such as the type of calibration signal and the TOF estimation procedure used as well as other design choices.

5.1 Calibration Signals

In order to measure the TOF accurately the calibration signal has to be appropriately selected and the parameters properly tuned. Chirp signals and Maximum Length sequences are the two most popular sequences for this task. A linear chirp signal is a short pulse in which the frequency of the signal varies linearly between two preset frequencies. The cosine linear chirp signal of duration T with the instantaneous frequency varying linearly between f_0 and f_1 is given by

$$s(t) = Acos(2\pi(f_0 + (\frac{f_1 - f_0}{T})t)) \quad 0 \leq t \leq T \tag{5.1}$$

In our system, we used a chirp signal of 512 samples at 44.1kHz (11.61 ms) as our calibration signal. The instantaneous frequency varied linearly from 5 kHz to 8 kHz. The initial and the final frequency was chosen to lie in the common pass band of the microphone and the speaker

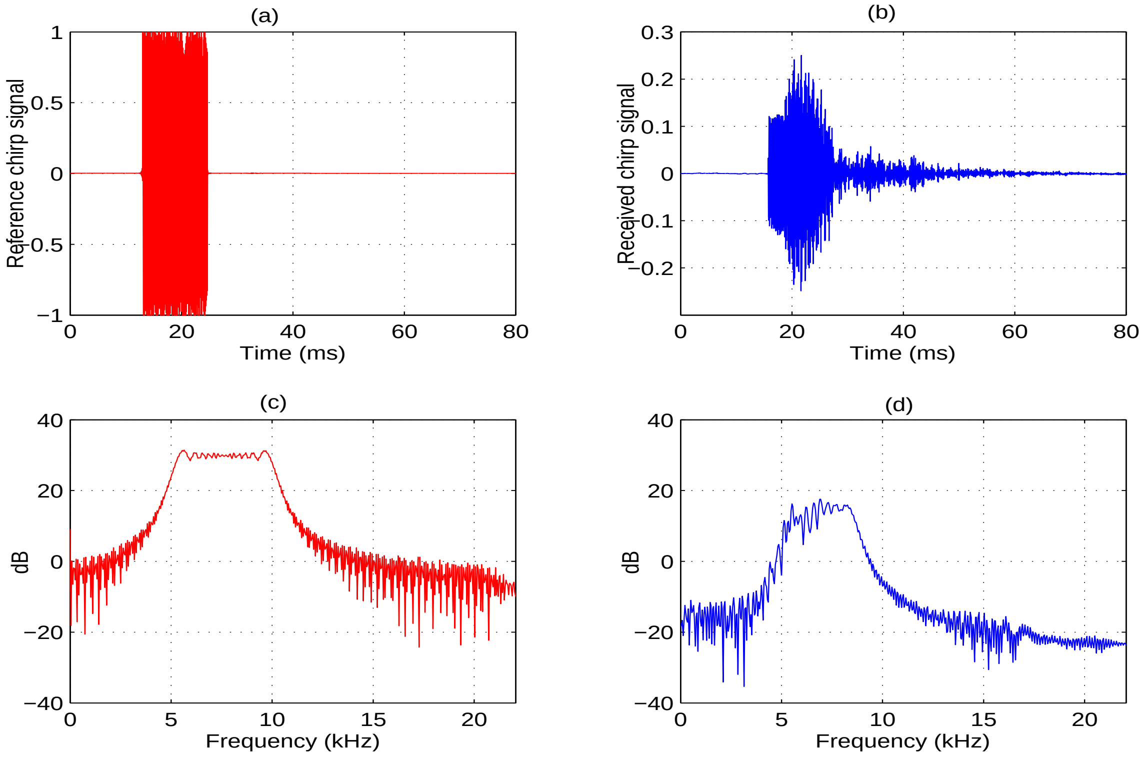


Figure 5.1: (a) The loopback reference chirp signal (b) the chirp signal received by one of the microphones (c) the magnitude of the spectrum of the reference signal and (d) the received chirp signal

frequency response. The chirp signal sent by the speaker is convolved with the room impulse response resulting in the spreading of the chirp signal. Figure 5.1(a) shows the chirp signal as sent out by the soundcard to the speaker. This signal is recorded by looping the output channels directly back to an input channel. The initial delay corresponds to the source emission time. Figure 5.1(b) shows the corresponding chirp signal received by a microphone. The chirp signal is delayed by a certain amount due to the propagation path. The distortion and the spreadout is due to the speaker, microphone and room response. Figure 5.1(c) and Figure 5.1(d) show the magnitude of the frequency response of the transmitted chirp signal and the received chirp signal, respectively.

One of the problems in accurately estimating the TOF is due to the multipath propagation caused by room reflections. This can be seen in the received chirp signal where the initial part corresponds to the direct signal and the rest are the room reflections. We use the Time Division Multiplexing scheme to send the calibration signal to different speakers. To avoid interference between the different calibration signals we zeropaded the calibration signal appropriately in dependence of the room reverberation level and the maximum delay. Alternatively, we could also use Frequency Division Multiplexing by allocating a frequency band at each channel or spread spectrum techniques by using different Maximum Length sequences for each channel. The advantage would be that all the output channels can be played simultaneously. However extra processing is needed at the input to separate the signals.

5.2 Time Delay Estimation

This is the most crucial part of the algorithm and also a potential source of error. Hence lot of care has to be taken to get the TOF measurements accurately in noisy and reverberant environments. The time-delay may be obtained by locating the peak in the cross-correlation function of the signals received by a pair of microphones. But this method is not robust to degradations in the signals. Knapp and Carter [18] developed the Maximum Likelihood (ML) estimator for determining the time-delay between signals received at two spatially separated microphones when the noise is uncorrelated. In this method, the estimated delay is the time lag which maximizes the cross-correlation between filtered versions of the received signals [18]. The cross-correlation of the filtered versions of the signals is called the Generalized Cross Correlation (GCC) function. The GCC function $R_{x_1x_2}(\tau)$ is given by [18]

$$R_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} W(\omega)X_1(\omega)X_2^*(\omega)e^{j\omega\tau}d\omega \tag{5.2}$$

where $X_1(\omega)$ and $X_2(\omega)$ are the Fourier transforms of the microphone signals $x_1(t)$ and $x_2(t)$, respectively, and $W(\omega)$ is the weight function. The effect of five different weight functions, namely, the Roth Impulse Response, the Smoothed Coherence Transform (SCOT), the Phase Transform (PHAT), the Eckart filter and the Maximum Likelihood (ML) weighting were studied in [18].

The two most commonly used weight functions are the ML and PHAT. The ML weight function accentuates the signal passed to the correlator at frequencies where the Signal-to-Noise Ratio (SNR) is high [18]. Brandstein, Adcock and Silverman [9] proposed an approximate ML type weighting for speech applications. The approximate weight function is given by

$$\widehat{W}_{ML}(\omega) = \frac{|X_1(\omega)||X_2(\omega)|}{|N_1(\omega)|^2|X_2(\omega)|^2 + |N_2(\omega)|^2|X_1(\omega)|^2} \tag{5.3}$$

where $|N_1(\omega)|$ and $|N_2(\omega)|$ are the noise power spectra at the two microphones, and are assumed to be known during the silence interval [9]. We use this weight function in our simulation studies. This ML weight function performs well when the effect of room reverberation is low. As the room reverberation increases, this method shows degradations in performance [6]. Since the spectral characteristics of the received signal are affected by the multipath propagation or reverberation in a room, the GCC function is made more robust by deemphasizing the frequency dependent weighting. The Phase Transform is one extreme case where the magnitude spectrum is flattened. The PHAT weight function $W_{PT}(\omega)$ is given by

$$W_{PT}(\omega) = \frac{1}{|X_1(\omega)X_2^*(\omega)|} \tag{5.4}$$

By flattening the magnitude spectrum the resulting peak in the GCC function corresponds to the dominant delay. However, the disadvantage of the PHAT weighting is that it places equal emphasis on both the low and high SNR regions, and hence it works well only when the overall noise level is low. Stéphanne and Champagne [30] proposed cepstral prefiltering to reduce the effects of reverberation. Benesty [7] proposed a novel method for time-delay estimation based on eigenvalue decomposition of the covariance matrix.

Our current implementation has the option of selecting between the simple Cross Correlation and GCC with PHAT weighting. We plan to appropriately combine the ML and the PHAT technique based on the environment. A more accurate estimate of the peak can be found by doing a parabolic fit around the peak. Figure 5.2 shows a sample GCC-PHAT function. The TOF is the position of the peak in the correlation function.

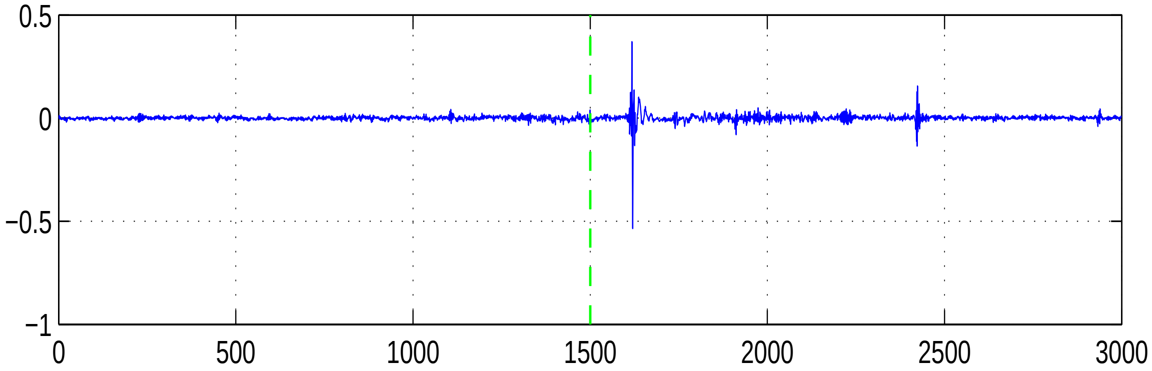


Figure 5.2: GCC function with PHAT weighting.

5.3 Testbed Setup

The real-time setup has been tested in a synchronized as well as a distributed setup using laptops. Figure 5.3(a) shows the top view of our experimental synchronized setup. Four omnidirectional microphones (RadioShack) and four loudspeakers (Mackie HR624) were setup in a room with low reverberation and low ambient noise. The microphones and loudspeakers were interfaced using an RME DIGI9652 card. For the distributed setup we used 5 laptops (IBM T-series Thinkpads with Intel Pentium series processors) as shown in Figure 5.4(a). For our experiments we used the internal microphones and speakers in the laptop. The room also had multiple PCs which acted as a noise sources. All the five laptops were placed on a flat table so that we can form a 2D coordinate system [1]. The ground truth was measured manually to validate the results from the position calibration methods.

5.4 Software details

Capture and play back was done using the free, cross platform, open-source, audio I/O library Portaudio [4]. PortAudio provides a very simple API for recording and/or playing sound using a simple callback function [4]. Most of the signal processing tasks were implemented using the Intel

[1] As discussed earlier we need minimum six laptops for the minimization routine. With 5 laptops we need to know the actual x-coordinate of one of the laptops.

Integrated Performance Primitives (IPP). IPP is a cross-platform low-level software layer that abstracts multimedia functionality from the processor underneath and provides highly optimized code [2]. For the non-linear minimization we used the *mrqmin* routine from Numerical Recipes in C [26]. For displaying the calibrated microphones and speakers we used the OpenGL Utility Toolkit (GLUT) ported to Win32 [5]. For the distributed platform we used the Universal Plug and Play (UPnP) [1] technology to form an adhoc network and control the audio devices on different platforms. UPnP technology is a distributed, open networking architecture that employs TCP/IP and other Internet technologies to enable seamless proximity networking [1]. Each of the laptops has an UPnP service running for playing the chirp signal and capturing the audio stream. A program on the master scans the network for all the available UPnP players. First the master starts the audio capture on each of the laptops one by one. Then the chirp signal is played on each of the devices one after the other and the signal is captured. The TOF computation is distributed among all the laptops, in that each laptop computes its own TOF and reports it back to the master. The master performs the minimization routine once it has the TOF matrix. As regards to CPU utilization the TOF estimation consumes negligible resources. If we use a good initial guess via the Multidimensional Scaling technique then the minimization routine converges within 10 to 30 iterations. For the setup consisting of 5 microphones and 5 speakers, Figure 5.4(b) shows the actual('X') and the estimated('o') positions of the microphones and speakers. The locations as got from the closed form approximate solution are shown as '*'. The localization error for each microphone or speaker is defined as the euclidean distance between the actual and the estimated positions. For our setup the average localization error was 8.2 cm. For the synchronized setup consisting of 4 speakers and 4 microphones, the sensors' and actuators' three dimensional locations could be estimated with an average bias of 0.08 cm and average standard deviation of 3 cm (results averaged over 100 trials). Figure 5.3(b) shows a snapshot of the OpenGL display, showing the estimated locations of the speakers and microphones.

Our algorithm assumed that the sampling rate was known for each laptop and the clock does not drift. However in practice the sampling rate is not as specified and the clock can also drift.

Hence our real time setup integrates the distributed synchronization scheme using ML sequence as proposed in [19]. This scheme essentially gives the exact sampling rate on each of the GPCs. Figure 5.5 shows a schematic of the TOF computation protocol.

5.5 Dealing with Partial Information

In some cases all TOF measurements may not be available. This could be either due to the presence of a large obstacle in between a microphone and a speaker, or an available TOF measurement may be an outlier. In such cases we can formulate the ML estimation procedure by ignoring the unavailable measurement. We can define a weighting function w, which takes the value 1, if the corresponding measurement is available, and else 0. For example for TOF- based position estimation, the ML estimate now becomes

$$\hat{\xi_{ML}} = \arg_{\xi} \min[F_{ML}(\xi)] \tag{5.5}$$

$$F_{ML}(\xi) = \sum_{j=1}^{S}\sum_{i=1}^{M} w_{ij} \frac{(TOF_{ij}^{estimated} - TOF_{ij}^{actual})^2}{\sigma_{ij}^2} \tag{5.6}$$

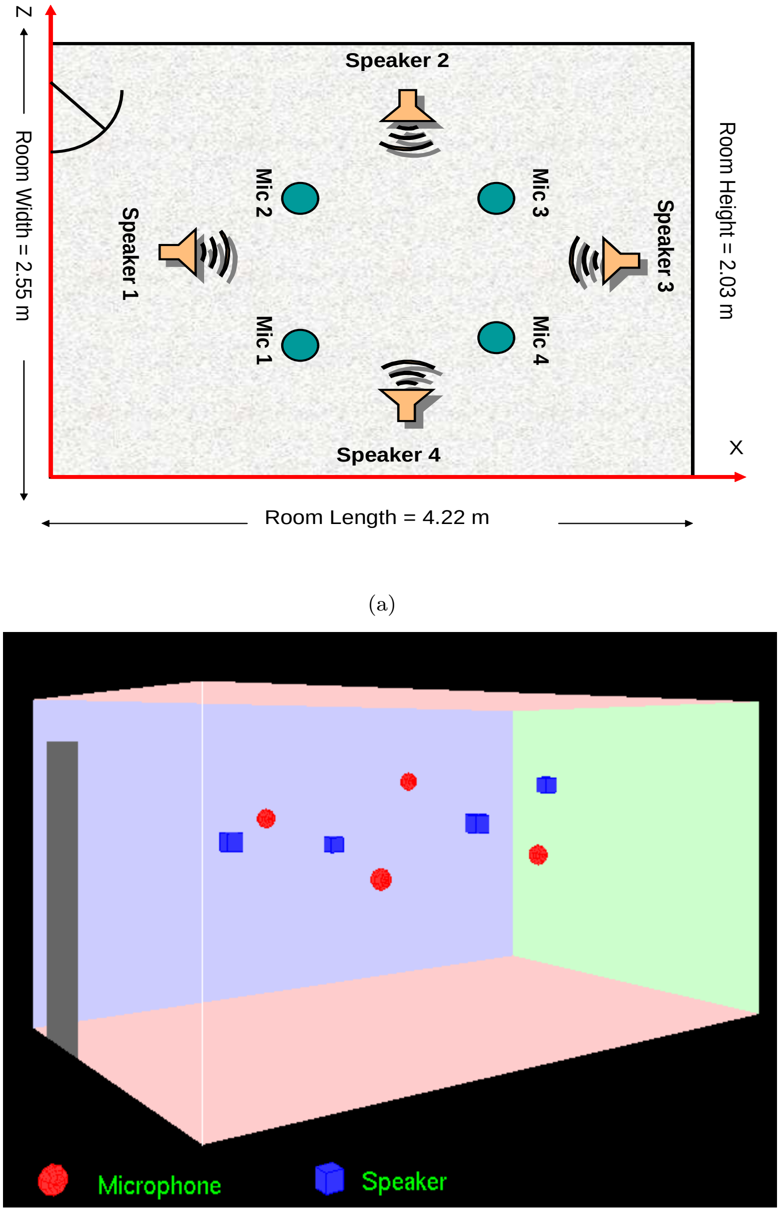
In the case of Multidimensional Scaling it is possible to fill in the missing data when all the pairwise distances are not available. If we know the distance between one node and at least four other nodes (three in case of 2 dimensions), then it is possible to find the distance between that node and all other nodes. First using the available pairwise distances between a subset of nodes, we can form a coordinate system for the subset of nodes and hence it is possible to know the locations of the subset of nodes. If the location of four nodes are known then by trilateration the node's position can be determined analytically. Knowing the distance from one known node, the unknown node can lie anywhere on a sphere centered at the first known node. With two nodes the unknown node can lie on a circle, since two spheres intersect at a circle. With three we can get two points, and with four nodes we can give a unique location. Since the estimated distances are corrupted by noise, the intersection in general needs not to be a unique point. If the distance to more than four nodes are known then we solve the problem in a least square sense. Once the node locations are known then the pairwise distances can be computed.

5.6 Robust ML estimation

In practice the TOF measurements may contain outliers (measurements which are in a gross disagreement with the underlying model). Outlier can have strong influence on the solution, and completely distort the nonlinear fitting function. In this situation we need to use robust methods for ML estimation. One method is to discard iteratively the measurement with the largest residual after the nonlinear least squares fitting. However this method not necessarily removes the actual outlier. The two other robust methods are the M estimators and the RANSAC method. In the case of M estimators we use some robust error metric in place of the squared error. One example is the Lorentzian function which gives less penalty to large errors as compared to the squared error function. The Lorentzian error function is given by

$$d(e_n) = \ln(1 + (\frac{e_n}{\sigma})^2) \tag{5.7}$$

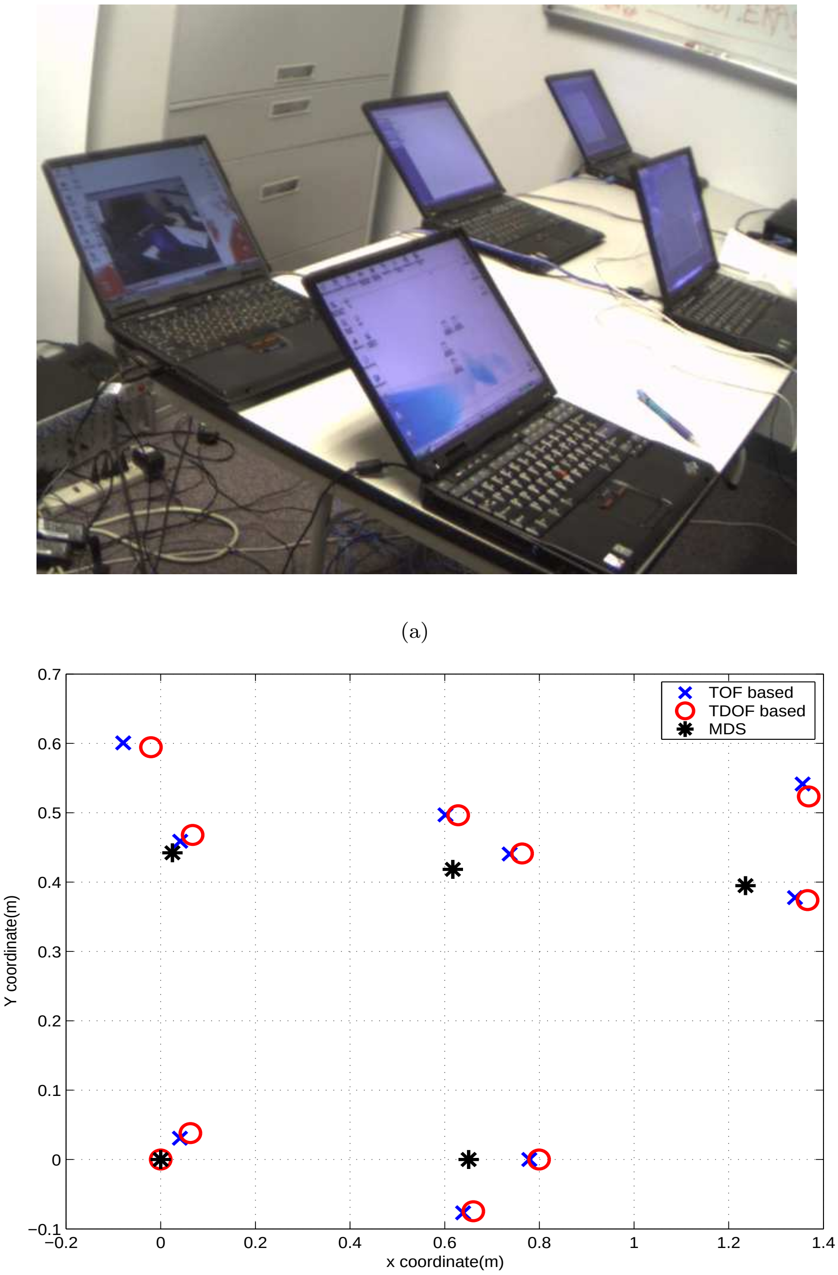
In RANdom SAmpling Consensus (RANSCAC) method we use the minimum data required to find a solution (See Table 2.2 for the minimum number required for each estimation procedure). This process is repeated on different subsets of the data to ensure that there is a high chance of one of the subsets containing only good data points. The best solution is that which maximizes the number of points whose residual is below a certain threshold. Once all the outliers are removed the estimation procedure can be done with only the good data.



(a)

(b)

Figure 5.3: (a) Synchronized setup consisting of four microphones and four loudspeakers. (b) A sample screen shot of the OpenGL display showing the positions of the microphones and loudspeakers for the synchronized setup.



(a)

(b)

Figure 5.4: (a) 2D Distributed setup consisting of 5 laptops placed on a table. (b) Results for the setup consisting of 5 laptops each having one internal microphone and speakers.

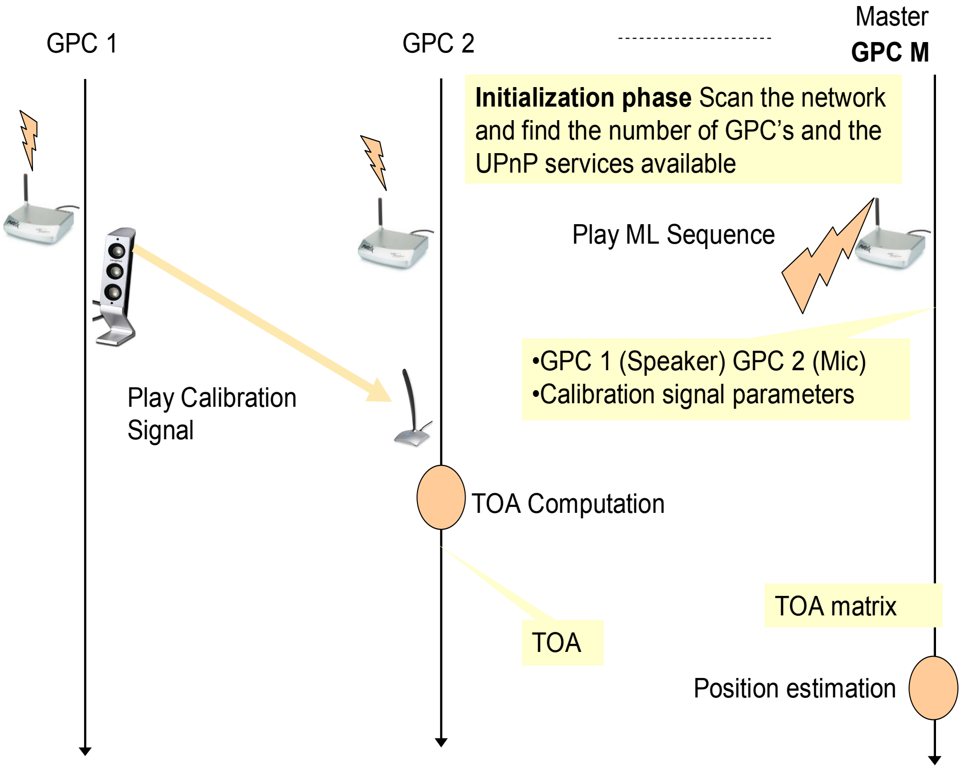

GPC 1
GPC 2
Master
GPC M
Initialization phase Scan the network and find the number of GPC's and the UPnP services available
Play ML Sequence
•GPC 1 (Speaker) GPC 2 (Mic)
•Calibration signal parameters
Play Calibration Signal
TOA Computation
TOA matrix
TOA
Position estimation

Figure 5.5: Schematic showing the distributed control scheme.

Chapter 6

Conclusions

In this thesis we described the problem of position calibration of acoustic sensors and actuators in a network of distributed general-purpose computing platforms. Our approach allows putting laptops, PDAs and tablets into a common 3D coordinate system. Together with time synchronization this creates arrays of audio sensors and actuators enabling a rich set of new multistream A/V applications on platforms that are available virtually anywhere. We also derived important bounds on performance of spatial localization algorithms, proposed optimization techniques to implement them and extensively validated the algorithms on simulated and real data.

Appendix A

Derivatives

Following are the derivatives which are needed for the minimization routine. These derivatives form the non-zero elements of the Jacobian matrix.

$$\begin{aligned}
\frac{\partial T\hat{O}F_{ij}^{actual}}{\partial mx_i} &= -\frac{\partial T\hat{O}F_{ij}^{actual}}{\partial sx_j} = \frac{mx_i - sx_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{O}F_{ij}^{actual}}{\partial my_i} &= -\frac{\partial T\hat{O}F_{ij}^{actual}}{\partial sy_j} = \frac{my_i - sy_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{O}F_{ij}^{actual}}{\partial mz_i} &= -\frac{\partial T\hat{O}F_{ij}^{actual}}{\partial sz_j} = \frac{mz_i - sz_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{O}F_{ij}^{actual}}{\partial ts_j} &= -\frac{\partial T\hat{O}F_{ij}^{actual}}{\partial tm_i} = 1
\end{aligned} \tag{A.1}$$

$$
\begin{aligned}
\frac{\partial T\hat{DOF}_{ikj}^{actual}}{\partial mx_i} &= \frac{mx_i - sx_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{DOF}_{ikj}^{actual}}{\partial mx_k} &= -\frac{mx_k - sx_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}_{ikj}^{actual}}{\partial my_i} &= \frac{my_i - sy_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{DOF}_{ikj}^{actual}}{\partial my_k} &= -\frac{my_k - sy_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}_{ikj}^{actual}}{\partial mz_i} &= \frac{mz_i - sz_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{DOF}_{ikj}^{actual}}{\partial mz_k} &= -\frac{mz_k - sz_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}_{ikj}^{actual}}{\partial sx_j} &= -\frac{mx_i - sx_j}{c\|m_i - s_j\|} + \frac{mx_k - sx_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}_{ikj}^{actual}}{\partial sy_j} &= -\frac{my_i - sy_j}{c\|m_i - s_j\|} + \frac{my_k - sy_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}_{ikj}^{actual}}{\partial sz_j} &= -\frac{mz_i - sz_j}{c\|m_i - s_j\|} + \frac{mz_k - sz_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}_{ikj}^{actual}}{\partial tm_k} &= -\frac{\partial T\hat{DOF}_{ikj}^{actual}}{\partial tm_i} = 1
\end{aligned}
\tag{A.2}
$$

Appendix B

Distance matrix to a dot product matrix

Let us say we choose the k^{th} GPC as the origin of our coordinate system. Let d_{ij} and b_{ij} be the distance and dotproduct respectively, between the i^{th} and the j^{th} GPC. Referring to Figure B.1, using the cosine law,

$$d_{ij}^2 = d_{ki}^2 + d_{kj}^2 - 2d_{ki}d_{kj}cos(\alpha) \tag{B.1}$$

The dot product b_{ij} is defined as

$$b_{ij} = d_{ki}d_{kj}cos(\alpha) \tag{B.2}$$

Combining the above two equations,

$$b_{ij} = \frac{1}{2}(d_{ki}^2 + d_{kj}^2 - d_{ij}^2) \tag{B.3}$$

However this is with respect to the k^{th} GPC as the origin of the coordinate system. We need to get the dot product matrix with the centroid as the origin. Let B be the dot product matrix with respect to the k^{th} GPC as the origin and let B^* be the dot product matrix with the centroid of the data points as the origin. Let X^* be to matrix of coordinates with the origin shifted to the

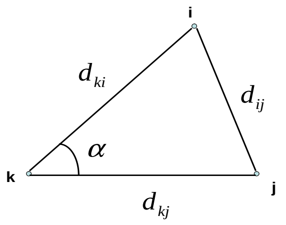


Figure B.1: Law of cosines

centroid.

$$X^* = X - \frac{1}{N}\mathbf{1}_{N\times N}X \tag{B.4}$$

where $\mathbf{1}_{N\times N}$ is an $N \times N$ matrix who's all elements are 1. So now B^* can be written in terms of B as follows:

$$B^* = X^*X^{*T} = (X - \frac{1}{N}\mathbf{1}_{N\times N}X)(X - \frac{1}{N}\mathbf{1}_{N\times N}X)^T \tag{B.5}$$

$$= XX^T - \frac{1}{N}XX^T\mathbf{1}_{N\times N} - \frac{1}{N}\mathbf{1}_{N\times N}XX^T + \frac{1}{N^2}\mathbf{1}_{N\times N}XX^T\mathbf{1}_{N\times N} \tag{B.6}$$

$$= B - \frac{1}{N}B\mathbf{1}_{N\times N} - \frac{1}{N}\mathbf{1}_{N\times N}B + \frac{1}{N^2}\mathbf{1}_{N\times N}B\mathbf{1}_{N\times N} \tag{B.7}$$

Hence the ij^{th} element in B^* is given by

$$b^*_{ij} = b_{ij} - \frac{1}{N}\sum_{l=1}^{N} b_{il} - \frac{1}{N}\sum_{m=1}^{N} b_{mj} + \frac{1}{N^2}\sum_{o=1}^{N}\sum_{p=1}^{N} b_{op} \tag{B.8}$$

Substituting Equation B.3 we get

$$b^*_{ij} = -\frac{1}{2}\left[d^2_{ij} - \frac{1}{N}\sum_{l=1}^{N} d^2_{il} - \frac{1}{N}\sum_{m=1}^{N} d^2_{mj} + \frac{1}{N^2}\sum_{o=1}^{N}\sum_{p=1}^{N} d^2_{op}\right] \tag{B.9}$$

This operation is also known as double centering i.e. subtract the row and the column means from its elements and add the grand mean and then multiply by $-\frac{1}{2}$.

BIBLIOGRAPHY

[1] http://intel.com/technology/upnp/.

[2] http://www.intel.com/software/products/perflib/.

[3] http://www.netlib.org/minpack/.

[4] http://www.portaudio.com/.

[5] http://www.xmission.com/nate/glut.html.

[6] S. Bédard, B. Champagne, and A. Stéphanne. Effects of room reverberation on time-delay estimation performance. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages II–261 – II–264, 1994.

[7] J. Benesty. Adaptive eigen value decomposition algorithm for passive acoustic source localization. *J. Acoust. Soc. Am.*, 107(1):384–391, 2000.

[8] D. P. Betrsekas. *Nonlinear Programming.* Athena Scientific, 1995.

[9] M. Brandstein, J. Adcock, and H. Silverman. A practical time-delay estimator for localizing speech sources with a microphone array. *Comput. Speech Lang.*, 9:153–169, September 1995.

[10] A. R. Chowdhury and R. Chellappa. Statistical bias and the accuracy of 3d reconstruction from video. *Submitted to International Journal of Computer Vision.*

[11] J. A. Fessler. Mean and variance of implicitly defined biased estimators (such as penalized maximum likelihood): Applications to tomography. *IEEE Trans. on Image Processing*, 5(3):493–506, March 1996.

[12] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization.* 1981.

[13] L. Girod, V. Bychkovskiy, J. Elson, and D. Estrin. Locating tiny sensors in time and space: A case study. In *Proc. International Conference on Computer Design*, September 2002.

[14] J. E. Greenberg and P. M. Zurek. *Microphone-Array Hearing Aids*, chapter 11, pages 229–253. Microphone arrays-Signal Processing Techniques and Applications. Springer-Verlag, 2001.

[15] Régine Le Bouquin Jeannès, Pascal Scalart, Gérard Faucon, and Chirstophe Beaugeant. Combined noise and echo reduction in hands-free systems: A survey. *IEEE Trans. Speech Audio Processing*, 9:808–820, Nov. 2001.

[16] Nabuo Kawaguchi, Shigeki Matsubara, Hiroyuki Iwa, Shoji Kajita, Kazuya Takeda, Fumitada Itakura, and Yasuyoshi Inagaki. Construction of speech corpus in moving car environment. In *Proc. Int. Conf. Spoken Language Processing*, volume III, pages 362–365, 2000.

[17] C. H. Knapp and G. C. Carter. The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-24(4):320–327, August 1976.

[18] C. H. Knapp and G. C. Carter. The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-24:320–327, Aug. 1976.

[19] R. Lienhart, I. Kozintsev, S. Wehr, and Minerva Yeung. On the importance of exact synchronization for distributed audio processing. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, April 2003.

[20] R. Moses, D. Krishnamurthy, and R. Patterson. A self-localization method for wireless sensor networks. *Eurasip Journal on Applied Signal Processing Special Issue on Sensor Networks*, 2003(4):348–358, March 2003.

[21] B. C. Ng and C. M. S See. Sensor-array calibration using a maxilmum-likelihood approach. *IEEE Trans. Acoust., Speech, Signal Processing*, 44(6):827–835, June 1996.

[22] S. Nordholm, I. Claesson, and N Grbić. *Optimal and Adaptive Microphone Arrays for Speech Input in Automobiles*, chapter 14, pages 307–329. Microphone arrays-Signal Processing Techniques and Applications. Springer-Verlag, 2001.

[23] S. Oh and V. Viswanathan. Hands-free voice communication in an automobile with a microphone array. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 281–284, 1992.

[24] M. Omologo, M. Matassoni, and P Svaizer. *Speech Recognition with Microphone Arrays*, chapter 15, pages 331–353. Microphone arrays-Signal Processing Techniques and Applications. Springer-Verlag, 2001.

[25] M. Omologo, P. Svaizer, and Matassoni. Environmental conditions and aocustic transduction in hands-free speech recognition. *Speech Communication*, 25:75–95, 1998.

[26] H. P. Press, S. A. Teukolsky, W. T. Vettring, and B. P. Flannery. *Numerical Recipes in C The Art of Scientific Computing.* Cambridge University Press, 2 edition, 1995.

[27] Y. Rockah and P. M. Schultheiss. Array shape calibration using sources in unknown locations Part II: Near-field sources and estimator implementation. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-35(6):724–735, June 1987.

[28] J. M. Sachar, H. F. Silverman, and W. R. Patterson III. Position calibration of large-aperture microphone arrays. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages II–1797 – II–1800, 2002.

[29] A. Savvides, C. C. Han, and M. B. Srivastava. Dynamic fine-grained localization in ad-hoc wireless sensor networks. In *Proc. International Conference on Mobile Computing and Networking*, July 2001.

[30] A. Stephene and B. Champagne. Cepstral prefiltering for time delay estimation in reverberant environments. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, volume 5, pages 3055–3058, May 1995.

[31] M. Steyvers. Multideimnsional scaling. *Encyclopedia of Cognitive Science*, 2002.

[32] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17:401–419, 1952.

[33] H. L. Van Trees. *Detection, Estimation, and Modulation Theory*, volume Part 1. Wiley-Interscience, 2001.

[34] C. Wang, S. Griebel, P Hsu, and M. Brandstein. Real-time automated video and audio capture with multiple camera and microphones. *Journal of VLSI Signal Processing Systems*, 29(1/2):81–100, Aug/Sep 2001.

[35] H. Wang and P. Chu. Voice source localization for automatic camera pointing system in videoconferencing. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 187–190, Apr. 1997.

[36] A. J. Weiss and B. Friedlander. Array shape calibration using sources in unknown locations-a maxilmum-likelihood approach. *IEEE Trans. Acoust., Speech, Signal Processing*, 37(12):1958–1966, December 1989.

[37] D. Zotkin, R. Duraiswami, V. Philomin, and L Davis. Smart videoconferencing. In *ICME*, pages 1597–1600, Aug. 2000.

Vikas Chandrakant Raykar

3409 Tulane drive, Apt. no. 13, Hyattsville MD 20783

(h) 301-422-6069 (w) 301-405-8753 (cell) 408-242-1863

vikas@umiacs.umd.edu

http://cvl.umiacs.umd.edu/users/vikas/

EDUCATION

Master of Science, December 2003

Advisors: Dr. Rama Chellappa and Dr. Ramani Duraiswami

Thesis: Position Calibration of Acoustic Sensors and actuators on General Distributed Computing Platforms

Department of Electrical Engineering

University of Maryland, College Park, MD

Major: Signal Processing Minor: Computer Engineering

Current GPA: 3.785/4.0

Bachelor of Engineering, May 2001

Electronics and Communication Engineering

Regional Engineering College, Trichy, India

Aggregate: 87.97% Equivalent GPA:4.0/4.0

Department Rank:1/51

PUBLICATIONS

1. *Position Calibration of Audio sensors and actuators in a distributed computing platform*

   Vikas C. Raykar, Igor Kozintsev and Rainer Lienhart (ACM Multimedia 2003, Berkeley, CA, USA, November 2003)

2. *Self Localization of acoustic sensors and actuators on distributed platforms* Vikas C. Raykar, Igor Kozintsev and Rainer Lienhart (ICCV 2003 International Workshop on Multimedia Technologies in E-Learning and Collaboration, Nice, France, October 2003)

3. *Tracking a moving speaker using excitation source information* Vikas C. Raykar, B. Yegnanarayana, S. R. Mahadeva Prasanna and Ramani Duraiswami (Eurospeech 2003, Geneva, September 2003)

4. *Extracting significant features from the HRTF* Vikas C. Raykar, B.Yegnanarayana Ramani Duraiswami and Larry Davis (International Conference on Auditory Displays 2003, Boston, July 2003)

5. *Virtual audio system customization using visual matching of ear parameters* D.Zotkin, R.Duraiswami, L.Davis, V.C.Raykar and A.Mohan (ICPR 2002, Quebec City, Canada. August 2002)

6. *Head Related Impulse Response Interpolation for Dynamic Spatialization* V.C.Raykar, T. V. Shreenivas and R.Raman (Texas Instruments DSPS Fest-2k, Bangalore, India, November 2000.)

SUBMITTED

1. *Speaker Localization using Excitation source information in speech* Vikas C. Raykar, B.Yegnanarayana, S. R. Mahadeva Prasanna and Ramani Duraiswami (submitted IEEE Trans. Speech and Audio Processing)

2. *Position Calibration of Microphones and Loudspeakers in Distributed Computing Platforms* Vikas C. Raykar, Igor Kozintsev and Rainer Lienhart (submitted IEEE Trans. Speech and Audio Processing Special Issue on Multi-channel Signal Processing for Audio and Acoustics Applications)

3. *Position Calibration of Multiple Microphones* Vikas C. Raykar and Ramani Duraiswami

(submitted to ICASSP 2003)

PATENTS FILED

1. Three-Dimensional Position Calibration of Audio Sensors and Actuators on a Distributed Computing Platform. (filed on 05/09/2003 along with Igor Kozintsev and Rainer Lienhart)
2. Method for 3-Dimensional position calibration of audio sensors and actuators on a distributed computing platform. (filed on 09/18/2003 along with Igor Kozintsev and Rainer Lienhart)

RESEARCH / WORK EXPERIENCE

Intern 02/2003 to 08/2003

Future Platforms Lab, Intel Research Labs, Intel Corporation, Santa Clara CA

Mentors:Dr.Igor Kozintsev and Dr. Rainer Lienhart

- Designed and implemented novel algorithms for 3D position calibration of a network of microphones/cameras and speakers/displays on distributed computing platforms.
- The work resulted in 3 publications and 2 patents filed.

Graduate Research Assistant 08/2001 to 02/2003

Perceptual Interfaces and Realities Laboratory, University of Maryland, College Park MD

Research Advisor: Dr. Ramani Duraiswami

- Implemented a real time video conferencing setup using a microphone array and a pan-tilt camera which includes auditory source localization, automatic camera pointing, face detection and multi channel speech enhancement. Also worked on Temporal and Spectral Decomposition of HRTFs.

Research Fellow 11/1999 to12/1999 and 04/2000 to 05/2000

Speech and Audio laboratory, Indian Institute of Science, Bangalore, India

Advisor: Dr.T. V. Sreenivas

- Implemented a real-time 3D spatial audio system using Head Related Transfer Functions (HRTFs). Also worked on modelling and interpolation of HRTFs.

Intern 05/20/1999 to 06/25/1999

Centre for Artificial Intelligence and Robotics, Bangalore, India

Advisor: Dr. Ambalal V. Patel

- Implemented PI and PD controller using Fuzzy Logic.

SKILLS

- Expertise in C, MATLAB & Win32 Programming.
- Experience in building real-time video conferencing systems with Matrox EVI-D30 pan-tilt camera and PowerDAQ data acquisition board.
- Built a real time face detection system using Firewire camera and openCV.
- Coding experience with MIL Image processing library, Intel Integrated Performance Primitives (IPP) Library, openCV and portaudio.
- Working knowledge of C++, FORTRAN, DirectX and Verilog.
- Working knowledge of MPI and openMP.
- Digital Signal Processors including Texas Instruments TMS320C30, TMS320C54.
- Assembly languages such as 8085, 8086 & 8051/52 microcontrollers.
- Worked on a wide range of platforms including MS-DOS 6.2, Windows XP/NT/200/95, Unix(Solaris)& Linux (Suse).
- Also familiar with packages like MS-Office, LaTeX and HTML programming.

COURSE PROJECTS

- Evaluation of kernel methods like KPCA, KLDA and KBDA for face detection.

- Fast Kernel Principal Component Analysis for the Polynomial and Gaussian kernels.
- Classification and Regression using Linear Networks, Multilayer Perceptrons and RBF's
- Video Codec Implementation, DC Image Extraction and Shot Segmentation.
- Verilog Implementation of pipelined CPU with precise interrupt handling.
- Optimization methods for sound source localization.
- Implementation of the Frost beamformer.

HONORS

- Best Outgoing student in the Electronics and communication Engineering Department at Regional Engineering College, Trichy, India for the year 2000-2001.
- Recipient of the prestigious National Science Fellowship Award (Engineering Stream) for the year 1999 under the KVPY scheme funded by the Department of Science and Technology (DST),Government of India.
- Recipient of the National Talent Search Examination (NTSE) scholarship in the year 1995.
- Was 18th rank in State level Entrance Examination into engineering Colleges.
- Was 13th rank in State level public examination.
- Was 2nd rank in Karnataka State in X standard public examination.

ACTIVITIES

- Organizer of National Level Technical Symposium Probe 99 held in Trichy, India.
- Participated in the KVPY summer robotics camp for engineering students held at IIT, Bombay, India.
- Student member of IEEE Signal processing society.
- Member of SPIC MACAY.

INTERESTS

Sketching, Painting, Amateur robotics, Astronomy.

REFERENCES

- Dr.Ramani Duraiswami(Director, Perceptual Interfaces and Reality Laboratory, University of Maryland, CollegePark)
- Dr.Rama Chellappa(Professor and Director of the Center for Automation Research, at the University of Maryland in College Park)
- Dr.B. Yegnanarayana(Visiting Professor, University of Maryland, CollegePark)
- Dr. Igor Kozintsev, Intel Research Labs, Santa Clara, CA
- Dr. Rainer Lienhart, Intel Research Labs, Santa Clara, CA