Position Calibration of Microphones and Loudspeakers in Distributed Computing Platforms

Vikas C. Raykar*, Igor Kozintsev and Rainer Lienhart

***Abstract*— In this paper we present a novel algorithm to automatically determine the relative 3D positions of audio sensors and actuators in an ad-hoc distributed network of heterogeneous general purpose computing platforms such as laptops, PDAs and tablets. A closed form approximate solution is derived, which is further refined by minimizing a non-linear error function. Our formulation and solution accounts for the lack of temporal synchronization among different platforms. We compare two different estimators, one based on the Time Of Flight (TOF) and the other based on Time Difference Of Flight (TDOF). We also derive an approximate expression for the mean and covariance of the implicitly defined estimator 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. We report extensive simulation results and discuss the practical details of implementing our algorithms in a real-life system.**

***Index Terms*— Multi-channel signal processing for audio and acoustics applications, Microphone array calibration, Self-localizing sensor networks, Self-position calibration, Multidimensional scaling, Cramér-Rao bound.**

I. Introduction and Motivation

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 cost of dedicated hardware and complexity of processing algorithms. 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). An ad-hoc network of GPCs can be used to capture/render different audio-visual scenes in a distributed fashion leading to novel emerging applications. A few examples of such applications include multi-stream audio/video rendering, smart audio/video conference rooms, meeting recordings, automatic lecture summarization, hands-free voice communication, object localization, and speech enhancement. The advantage of such an approach is that 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. A prerequisite for using distributed audio-visual I/O capabilities is to put sensors and actuators into a common time and space (coordinate system). In [1] we proposed a way to provide a common time reference for multiple distributed GPCs with the precision of ten's of microseconds. In this paper we focus on 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 [2].

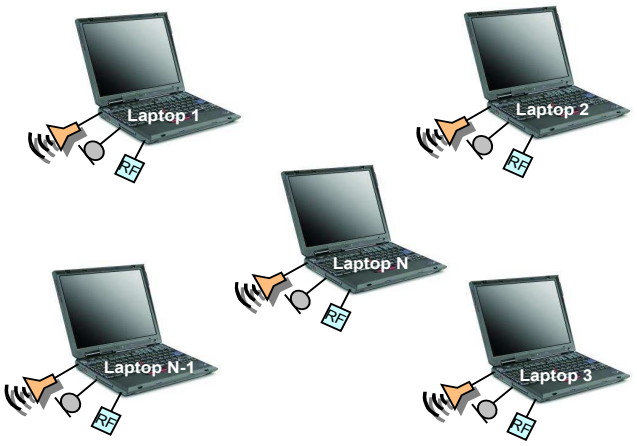


Fig. 1. Distributed computing platform consisting of N general-purpose computers along with their onboard audio sensors, actuators and wireless communication capabilities.

Figure 1 shows a schematic representation of our *distributed computing platform* consisting of N GPCs. In our setting, one of them is configured to be the master. The master controls the distributed computing platform and performs the location estimation. Each GPC is assumed to be equipped

Vikas C. Raykar is with the Perceptual Interfaces and Realities Laboratory, Institute of Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA E-mail:vikas@umiacs.umd.edu. The work was performed while he was an intern at Intel Labs, Intel Corporation, Santa Clara, USA.

Igor Kozintsev and Rainer Lienhart are with Intel Labs, Intel Corporation, Santa Clara, USA E-mail: {igor.kozintsev, rainer.lienhart}@intel.com

with audio sensors (microphones), actuators (speakers) for performing audio I/O, and wireless communication capabilities for exchanging data between each other.

A. 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. [3] 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 [2], [4], [5]. 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 [6] 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 [6]–[8]. The problem is essentially the same as in our case but the ranging method differ depending on the sensors and actuators.

B. Contributions

The following are the novel contributions of this paper.

- 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 [3], [4], [6]. 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) [9] 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 [8] 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 [3], [6], [8]. 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 [10]. We also derive the Cramèr-Rao bound and analyze the localization accuracy with respect to the number of sensors and sensor geometry.

C. Organization

The rest of the paper is organized as follows. In Section II, 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 Section III we derive an approximate closed form solution, which can be used as an initial guess for the non-linear minimization routine. In Section IV we derive the theoretical mean and covariance of the estimated parameters. The Cramér-Rao bound is derived and analyzed for its sensitivity with respect to the number of sensors and actuators as well as their geometry. In Section V, extensive simulation results are reported. Section VI gives a discussion of the issues involved in designing a practical system. Section VII, concludes with a summary of the present work, and with a discussion on possible extensions.

II. Problem Formulation

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 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.

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

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),

$$TOF_{ij}^{actual} = \frac{\parallel \mathbf{m_i} - \mathbf{s_j} \parallel}{c} \tag{1}$$

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 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, 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. As a result, the TOF which we measure includes both the speaker emission start time and the microphone capture start time (See Figure 2 where $\hat{TOF}_{ij}$ 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 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). Incorporating these two the actual TOF now becomes,

$$\begin{aligned} \hat{TOF}_{ij}^{actual} &= TOF_{ij}^{actual} + ts_j - tm_i \\ &= \frac{\parallel \mathbf{m_i} - \mathbf{s_j} \parallel}{c} + ts_j - tm_i \end{aligned} \tag{2}$$

The origin can be arbitrary since $\hat{TOF}_{ij}^{actual}$ 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.

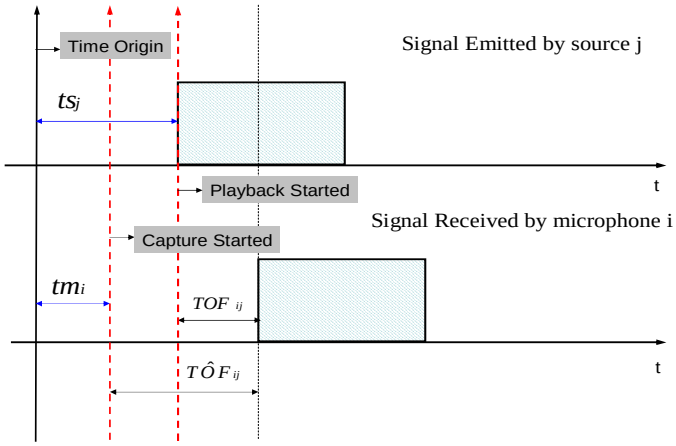


Fig. 2. Schematic indicating the unknown emission and capture start time.

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 [1]. 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.

A. Time Difference Of Flight

In this paper we propose to use the Time Difference Of Flight instead of the TOF. 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

$$TDOF_{ikj}^{actual} = \frac{\parallel \mathbf{m_i} - \mathbf{s_j} \parallel - \parallel \mathbf{m_k} - \mathbf{s_j} \parallel}{c} \tag{3}$$

Including the source emission and capture start times, it becomes

$$\hat{TDOF}_{ikj}^{actual} = \frac{\parallel \mathbf{m_i} - \mathbf{s_j} \parallel - \parallel \mathbf{m_k} - \mathbf{s_j} \parallel}{c} + tm_k - tm_i \tag{4}$$

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]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.

[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.

[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.

B. 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 function of Γ in vector form can be written as:

$$p(\Gamma/\Theta) = (2\pi)^{-\frac{N}{2}} \mid \Sigma \mid^{-\frac{1}{2}} \exp -\frac{1}{2}(\Gamma - T)^T \Sigma^{-1} (\Gamma - T) \quad (5)$$

The log-likelihood function is given by

$$\ln p(\Gamma/\Theta) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2} ln \mid \Sigma \mid -\frac{1}{2}(\Gamma - T)^T \Sigma^{-1} (\Gamma - T) \quad (6)$$

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

$$\hat{\Theta}_{ML} = \arg_{\Theta} \max F(\Theta, \Gamma)$$
$$F(\Theta, \Gamma) = -\frac{1}{2}[\Gamma - T(\Theta)]^T \Sigma^{-1} [\Gamma - T(\Theta)] \quad (7)$$

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

$$\hat{\Theta}_{ML} = \arg_{\Theta} \min[\widetilde{F}_{TDOF}(\Theta, \Gamma)]$$
$$\widetilde{F}_{TDOF}(\Theta, \Gamma) =$$
$$\sum_{j=1}^{S} \sum_{i=1}^{M} \sum_{k=i+1}^{M} \frac{(TDOF_{ikj}^{estimated} - \hat{TDOF}_{ikj}^{actual})^2}{\sigma_{ikj}^2} \quad (8)$$

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

$$\hat{\Theta}_{ML} = \arg_{\Theta} \min[\widetilde{F}_{TOF}(\Theta, \Gamma)]$$
$$\widetilde{F}_{TOF}(\Theta, \Gamma) =$$
$$\sum_{j=1}^{S} \sum_{i=1}^{M} \frac{(TOF_{ij}^{estimated} - TOF_{ij}^{actual})^2}{\sigma_{ij}^2} \quad (9)$$

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

C. 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.

D. 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 8. 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 [12] is a popular method for solving non-linear least squares problems. For more details on nonlinear minimization refer to [13]. Appendix II 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. In Section III we derive an approximate closed form solution which can be used to initialize the minimization routine.

E. 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 I 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 $M-1$ out of $M(M-1)/2$ pair of TDOF measurements for each speaker

[5]We estimate the TDOF or TOF using Generalized Cross Correlation (GCC) [11]. 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. [11].

[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).

[7]Many commercial software solutions are available for the Levenberg-Marquardt method such as *lsqnonlin* in MATLAB, *mrqmin* provided by Numerical Recipes in C [14] , and the MINPACK-1 routines [15]

TABLE I

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

TABLE II

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

are linearly independent. Assuming M=S=K, the Table II lists the minimum K required for least squares fitting.

III. CLOSED FORM APPROXIMATE SOLUTION

In this section we make some approximations to get closed form solutions to the microphone and speaker positions and the capture start times.

A. 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 $\hat{TOF}_{ii}$, $\hat{TOF}_{jj}$, $\hat{TOF}_{ij}$ and $\hat{TOF}_{ji}$. Assuming no noise these are related to the actual TOF as follows:

$$\begin{aligned} \hat{TOF}_{ii} &= TOF_{ii} + ts_i - tm_i \\ \hat{TOF}_{jj} &= TOF_{jj} + ts_j - tm_j \\ \hat{TOF}_{ij} &= TOF_{ij} + ts_j - tm_i \\ \hat{TOF}_{ji} &= TOF_{ji} + ts_i - tm_j \end{aligned} \quad (10)$$

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.

$$\begin{aligned} TOF_{ii} \approx TOF_{jj} \approx 0 \\ TOF_{ij} \approx TOF_{ji} \end{aligned} \quad (11)$$

Substituting we have the following equations:

$$\begin{aligned} \hat{TOF}_{ii} &\approx ts_i - tm_i \\ \hat{TOF}_{jj} &\approx ts_j - tm_j \\ \hat{TOF}_{ij} &\approx TOF_{ij} + ts_j - tm_i \\ \hat{TOF}_{ji} &\approx TOF_{ij} + ts_i - tm_j \end{aligned} \quad (12)$$

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

$$TOF_{ij} \approx \frac{(\hat{TOF}_{ij} + \hat{TOF}_{ji}) - (\hat{TOF}_{ii} + \hat{TOF}_{jj})}{2} \quad (13)$$

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

$$\begin{aligned} ts_i &\approx \hat{TOF}_{ii} + tm_i \\ tm_j &\approx \frac{(\hat{TOF}_{ij} - \hat{TOF}_{ji}) + (\hat{TOF}_{ii} - \hat{TOF}_{jj})}{2} + tm_i \end{aligned} \quad (14)$$

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.

B. 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) [9]. 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 [16]. 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.

If proximity data are based on the Euclidean distances, then classical metric MDS [9] 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) [14], 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 $\| B - XX^T \|$.

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 [9] 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 I for a detailed derivation of how to convert the distance matrix to the scalar product matrix.

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

$$d_{ij} \approx \frac{c\,(T\hat{O}F_{ij} + T\hat{O}F_{ji} - T\hat{O}F_{ii} - T\hat{O}F_{jj})}{2} \qquad (15)$$

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 II-C. The approximate locations of the GPCs are slightly perturbed to get the initial guess for the microphone and speaker locations. The following table summarizes the complete algorithm:

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**:
  - *Form the approximate distance matrix D. (Equation 15)*
  - *Assume $tm_1 = 0$ (microphone on which capture was started first) and get the approximate microphone capture and speaker emission start times. (Equation 14)*
  - *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 (SVD of B).*
  - *Translate, rotate and mirror the coordinates to the coordinate system specified in STEP 3.*
  - *Slightly perturb the coordinates to get approximate initial guess for the microphone and speaker coordinates.*
- **STEP 5**: *Minimize both 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.*

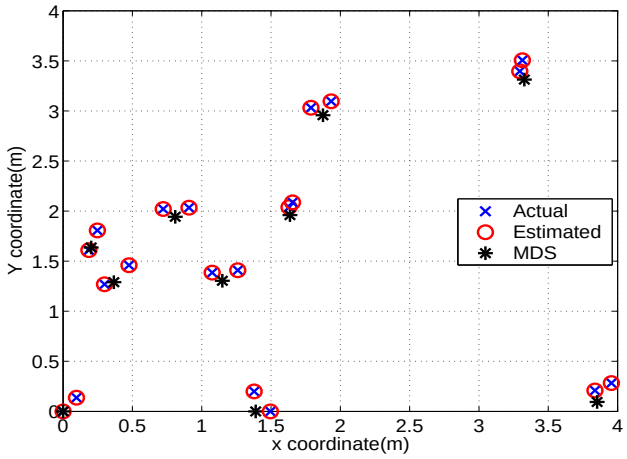


Fig. 3. Results of Multidimensional Scaling for a network consisting of 10 GPCs each having one microphone and one speaker.

Figure 3 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.

IV. Estimator bias and variance

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 Taylors' series it is possible to derive approximate expressions for the mean and variance of implicitly defined estimators [10]. In this section we derive the approximate expressions for both the mean and variance of the estimators. We follow the same approach as in [10]. The ML estimate of Θ is the one which maximizes the log likelihood ratio and is given by Equation 7. In further derivation we need the first and second derivatives of Equation 7 with respect to

Θ and Γ. Using the generalized chain rule it can be shown that for Equation 7 the vector derivatives are as follows

$$\begin{aligned}\nabla_\Theta F(\Theta,\Gamma) &= J^T\Sigma^{-1}(\Gamma - T)\\ \nabla_\Gamma F(\Theta,\Gamma) &= \Sigma^{-1}(\Gamma - T)\\ \nabla_\Theta\nabla_\Theta F(\Theta,\Gamma) &= -J^T\Sigma^{-1}J\\ \nabla_\Gamma\nabla_\Gamma F(\Theta,\Gamma) &= \Sigma^{-1}\\ \nabla_\Gamma\nabla_\Theta F(\Theta,\Gamma) &= \Sigma^{-1}J\\ \nabla_\Theta\nabla_\Gamma F(\Theta,\Gamma) &= -J^T\Sigma^{-1}\end{aligned} \tag{16}$$

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

$$[J]_{ij} = \frac{\partial t_i(\Theta)}{\partial \theta_j} \tag{17}$$

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

A. Estimator Covariance

The ML estimate of Θ is the one which maximizes the log likelihood ratio defined in Equation 7. The maximum can be found by setting the first derivative to zero i.e.

$$\nabla_\Theta F(\Theta,\Gamma)\mid_{\Theta=\hat{\Theta}} = \mathbf{0} \tag{18}$$

where $\mathbf{0}$ is a zero column vector of length P. The implicit function theorem guarantees that Equation 18 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 18 can be written as

$$\nabla_\Theta F(h(\Gamma),\Gamma) = \mathbf{0} \tag{19}$$

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

$$h(\Gamma) \approx h(\Gamma_m) + [\nabla_\Gamma h(\Gamma)^T\mid_{\Gamma=\Gamma_m}]^T(\Gamma - \Gamma_m) \tag{20}$$

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

$$Cov(h(\Gamma)) \approx [\nabla_\Gamma h(\Gamma)^T\mid_{\Gamma=\Gamma_m}]^T Cov(\Gamma)[\nabla_\Gamma h(\Gamma)^T\mid_{\Gamma=\Gamma_m}] \tag{21}$$

Note we do not know $h(\Gamma)$. Differentiating Equation 19 with respect to Γ and evaluating at Γ_m yields

$$\nabla_\Theta\nabla_\Theta F(h(\Gamma_m),\Gamma_m)[\nabla_\Gamma h(\Gamma_m)^T]^T + \nabla_\Theta\nabla_\Gamma F(h(\Gamma_m),\Gamma_m) = \mathbf{0} \tag{22}$$

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

$$[\nabla_\Gamma h(\Gamma_m)^T]^T = -[\nabla_\Theta\nabla_\Theta F(h(\Gamma_m),\Gamma_m)]^{-1}\nabla_\Theta\nabla_\Gamma F(h(\Gamma_m),\Gamma_m) \tag{23}$$

Substituting from Equation 16 we get

$$[\nabla_\Gamma h(\Gamma_m)^T]^T = -[J^T\Sigma^{-1}J]^{-1}J^T\Sigma^{-1} \tag{24}$$

Using this in the covariance expression, we final arrive at

$$Cov\hat{\Theta} = [J^T\Sigma^{-1}J]^{-1} \tag{25}$$

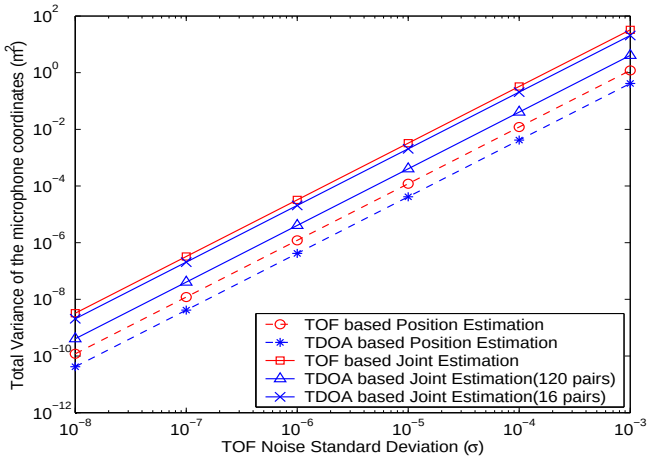


Fig. 4. Cramér-Rao bound on the total variance of the unknown microphone coordinates as a function of TOF noise standard deviation σ for different estimation procedures. For the TDOF- based method the noise variance was taken as twice that of the TOF variance. The network had a total of 16 microphones and 16 speakers.

B. Estimator Mean

Taking the expectation of the first order Taylor series expansion in Equation 20

$$E(h(\Gamma)) \approx h(\Gamma_m) = h(T) \tag{26}$$

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 25.

C. Cramér-Rao Bound

The Cramér-Rao bound gives a lower bound on the variance of *any* unbiased estimate [17]. 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 [17]

$$E\left[(\hat{\Theta} - \Theta)(\hat{\Theta} - \Theta)^T\right] \geq F^{-1}(\Theta) \tag{27}$$

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

$$F(\Theta) = E\left\{[\nabla_\Theta \ln p(\Gamma/\Theta)]\,[\nabla_\Theta \ln p(\Gamma/\Theta)]^T\right\} \tag{28}$$

Any estimate which satisfies the bound with an equality is called an efficient estimate. The ML estimate is consistent and asymptotically efficient [17].

The derivative of the log-likelihood function (see Equation 6) can be found using the generalized chain rule and is given by (refer Equation 16)

$$\nabla_\Theta \ln p(\Gamma/\Theta) = J^T\Sigma^{-1}(\Gamma - T) \tag{29}$$

where J is the *Jacobian*. Substituting this in Equation 28 and taking the expectation the Fishers Information matrix is,

$$F = J^T\Sigma^{-1}J \tag{30}$$

$$Cov\hat{\Theta} \geq [J^T \Sigma^{-1} J]^{-1} \quad (31)$$

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

If we assume that all the microphone and source locations are unknown, the Fisher Information matrix $J^T\Sigma^{-1}J$ 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.

The diagonal terms of $[J^T\Sigma^{-1}J]^{-1}$ represent the error variance for estimating each of the parameters in Θ. In the next few sections we explore the dependency of the error variance on different parameters. Figure 4 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 [8].

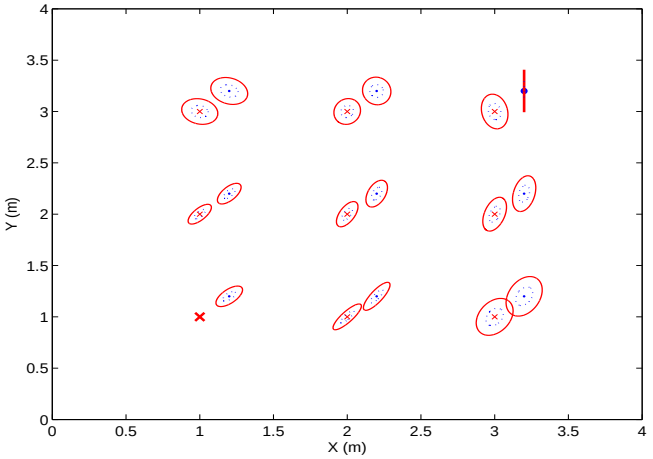
D. Effect of nuisance parameters

The speaker emission start time and the microphone capture start time can be considered as the nuisance parameters since we are interested only in the microphone and speaker coordinates. The effect of the nuisance parameters on the Cramér Rao bound can be seen from Figure 4, 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.

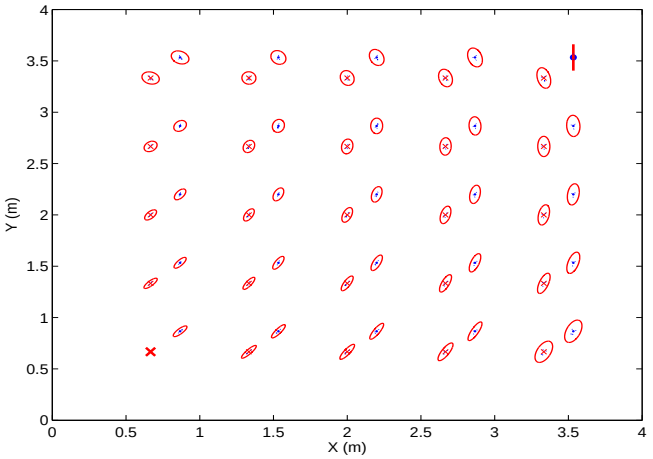
E. Increasing the number of sensors and actuators

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 5(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 uncertainity ellipses. In order to give a fair comparison,

[8]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.



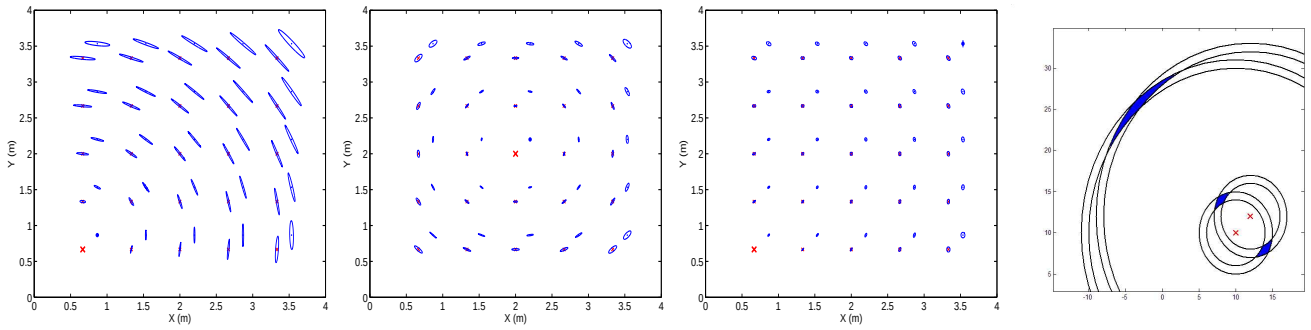
(a)



(b)

Fig. 5. 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 noise variance of 2×10^{-9} was assumed for the TDOF-based method. Figure 5(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 uncertainity per unknown variable will decrease.



(a) (b) (c) (d)

Fig. 6. 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 ($\times$) 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.

F. Sensor Geometry - How to select a coordinate system?

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 6 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 6(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 6(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 6(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 6(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.

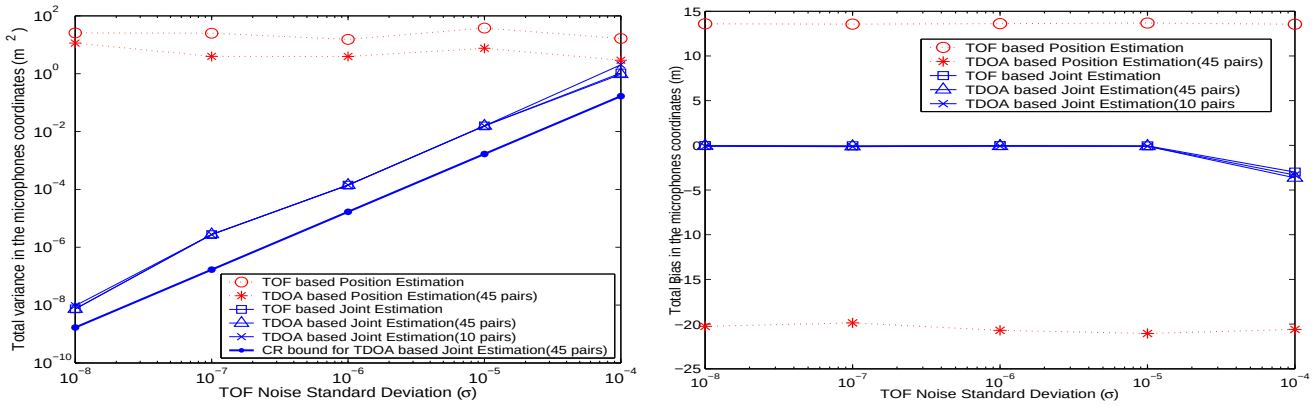
V. Monte Carlo Simulation Results

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 7(a) and Figure 7(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.

VI. Implementation Details

A. 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 $s(t) = Acos(2\pi(f_0 + (\frac{f_1 - f_0}{T})t)) \quad 0 \leq t \leq T$ In our system, we used the chirp signal of 512 samples at 44.1kHz (11.61 ms) as our calibration signal. The instantaneous frequency



(a) (b)

Fig. 7. (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.

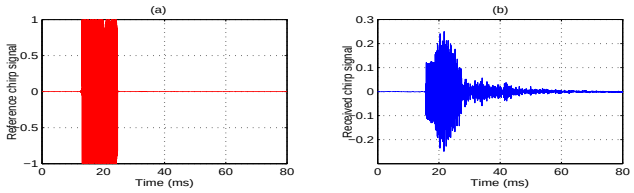


Fig. 8. (a) The loopback reference chirp signal (b) the chirp signal received by one of the microphones

varied linearly from 5 kHz to 10 kHz. The initial and the final frequency was chosen to lie in the common passband of the microphone and the speaker frequency response. The chirp signal send by the speaker is convolved with the room impulse response resulting in the spreading of the chirp signal. Figure 8(a) shows the chirp signal as sent out by the soundcard to the speaker. This signal is recorded by looping the output channel directly back to an input channel, on a multichannel sound card. The initial delay is due to the emission start time and the capture start time. Figure 8(b) shows the corresponding chirp signal received by the 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.

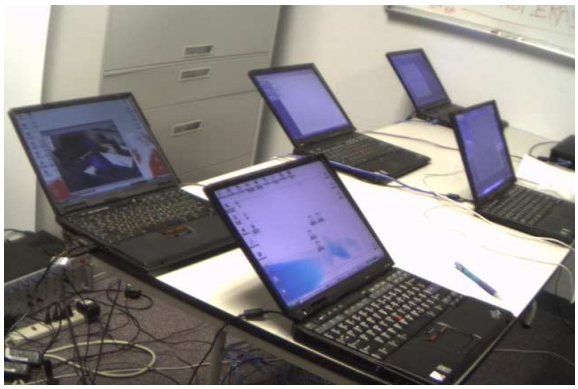
B. Time Delay Estimation

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 accurately in noisy and reverberant environments. The time-delay may be found by locating the peak in the cross-correlation of the signals received over the two microphones. However this method is not robust to noise and reverberations. Knapp and Carter [11] developed a ML estimator for determining the time delay between signals received at two spatially separated sensors in the presence of uncorrelated noise. In this method, the delay estimate is the time lag which maximizes the cross-correlation between filtered versions of the received signals [11]. The cross-correlation of the filtered versions of the signals is called as the Generalized Cross Correlation (GCC) function. The GCC function $R_{x_1x_2}(\tau)$ is computed as [11] $R_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} W(\omega)X_1(\omega)X_2^*(\omega)e^{j\omega\tau}d\omega$, where $X_1(\omega)$, $X_2(\omega)$ are the Fourier transforms of the microphone signals $x_1(t)$, $x_2(t)$, respectively and $W(\omega)$ is the weighting function. The two most commonly using weighting functions are the ML and the Phase Transform (PHAT) weighting. The ML weighting function, accentuates the signal passed to the correlator at frequencies for which the signal-to-noise ratio is the highest and, simultaneously suppresses the noise power [11]. This ML weighting function performs well for low room reverberation. As the room reverberation increases this method shows severe performance degradations. Since the spectral characteristics of the received signal are modified by the multipath propagation in a room, the GCC function is made more robust by deemphasizing the frequency dependent weightings. The Phase Transform is one extreme where the magnitude spectrum is flattened. The PHAT weighting is given by $W_{PHAT}(\omega) = \frac{1}{|X_1(\omega)X_2^*(\omega)|}$. By flattening out 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 emphasizes on both the low and high SNR regions, and hence it works well only when the noise level is low. For low noise rooms the PHAT method performs moderately well.

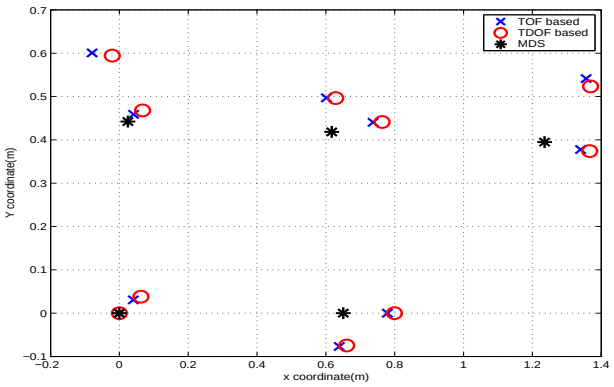
C. Testbed Setup and Results

The algorithm has been tested in a real time distributed setup using 5 laptops (IBM T-series Thinkpads with Intel Pentium series processors). Figure 9(a) shows our experimental setup. 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 [9]. The ground truth was measured manually to validate the results from the position calibration methods. For our experiments we used the internal microphones and speakers in the laptop.

[9]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.



(a)



(b)

Fig. 9. (a) Our experimental setup (b) Results for a setup consisting of 5 laptops each having one internal microphone and speaker.

Capture and play back was done using the free, cross platform, open-source, audio I/O library Portaudio [18]. Most of the signal processing tasks were implemented using the Intel Integrated Performance Primitives (IPP) For the non-linear minimization we used the *mrqmin* routine from Numerical Recipes in C [14]. For the distributed platform we used the Universal Plug and Play (UPnP) [19] 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 [19]. 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. For the setup consisting of 5 microphones and 5 speakers, Figure 9 shows the estimated positions of the microphones and speakers using both the methods. 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. We also implemented the same system on a synchronized platform for which the error was 3.8 cm. 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 [1]. This scheme essentially gives the exact sampling rate on each of the GPCs.

VII. Conclusions

In this paper 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 I
Converting the 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 10, using the cosine law,

$$d_{ij}^2 = d_{ki}^2 + d_{kj}^2 - 2d_{ki}d_{kj}cos(\alpha) \tag{32}$$

The dot product b_{ij} is defined as

$$b_{ij} = d_{ki}d_{kj}cos(\alpha) \tag{33}$$

Combining the above two equations,

$$b_{ij} = \frac{1}{2}(d_{ki}^2 + d_{kj}^2 - d_{ij}^2) \tag{34}$$

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 centroid.

$$X^* = X - \frac{1}{N}\mathbf{1}_{N\times N}X \tag{35}$$

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:

$$\begin{aligned} B^* &= X^*X^{*T} \\ &= B - \frac{1}{N}B\mathbf{1}_{N\times N} - \frac{1}{N}\mathbf{1}_{N\times N}B + \frac{1}{N^2}\mathbf{1}_{N\times N}B\mathbf{1}_{N\times N} \end{aligned}$$

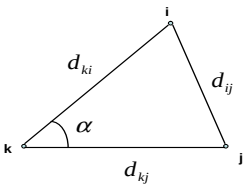


Fig. 10. Law of cosines

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

$$b^*_{ij} = b_{ij} - \frac{1}{N}\sum_{l=1}^{N} b_{il} - \frac{1}{N}\sum_{m=1}^{N} b_{mj} + \frac{1}{N^2}\sum_{o=1}^{N}\sum_{p=1}^{N} b_{op} \quad (36)$$

Substituting Equation 34 we get

$$b^*_{ij} = -\frac{1}{2}\left[d^2_{ij} - \frac{1}{N}\sum_{l=1}^{N} d^2_{il} - \frac{1}{N}\sum_{m=1}^{N} d^2_{mj} + \frac{1}{N^2}\sum_{o=1}^{N}\sum_{p=1}^{N} d^2_{op}\right] \quad (37)$$

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 $-\frac{1}{2}$.

APPENDIX II
DERIVATIVES

Following are the derivable which are needed for the minimization routine and the Cramer-Rao bound. These derivatives form the non-zero elements of the Jacobian matrix.

$$\begin{aligned}
\frac{\partial T\hat{DOF}^{actual}_{ikj}}{\partial mx_i} &= \frac{mx_i - sx_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{DOF}^{actual}_{ikj}}{\partial mx_k} &= -\frac{mx_k - sx_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}^{actual}_{ikj}}{\partial my_i} &= \frac{my_i - sy_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{DOF}^{actual}_{ikj}}{\partial my_k} &= -\frac{my_k - sy_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}^{actual}_{ikj}}{\partial mz_i} &= \frac{mz_i - sz_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{DOF}^{actual}_{ikj}}{\partial mz_k} &= -\frac{mz_k - sz_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}^{actual}_{ikj}}{\partial sx_j} &= -\frac{mx_i - sx_j}{c\|m_i - s_j\|} + \frac{mx_k - sx_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}^{actual}_{ikj}}{\partial sy_j} &= -\frac{my_i - sy_j}{c\|m_i - s_j\|} + \frac{my_k - sy_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}^{actual}_{ikj}}{\partial sz_j} &= -\frac{mz_i - sz_j}{c\|m_i - s_j\|} + \frac{mz_k - sz_j}{c\|m_k - s_j\|} \\
\frac{\partial T\hat{DOF}^{actual}_{ikj}}{\partial tm_k} &= -\frac{\partial T\hat{DOF}^{actual}_{ikj}}{\partial tm_i} = 1
\end{aligned} \quad (38)$$

$$\begin{aligned}
\frac{\partial T\hat{OF}^{actual}_{ij}}{\partial mx_i} &= -\frac{\partial T\hat{OF}^{actual}_{ij}}{\partial sx_j} = \frac{mx_i - sx_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{OF}^{actual}_{ij}}{\partial my_i} &= -\frac{\partial T\hat{OF}^{actual}_{ij}}{\partial sy_j} = \frac{my_i - sy_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{OF}^{actual}_{ij}}{\partial mz_i} &= -\frac{\partial T\hat{OF}^{actual}_{ij}}{\partial sz_j} = \frac{mz_i - sz_j}{c\|m_i - s_j\|} \\
\frac{\partial T\hat{OF}^{actual}_{ij}}{\partial ts_j} &= -\frac{\partial T\hat{OF}^{actual}_{ij}}{\partial tm_i} = 1
\end{aligned} \quad (39)$$

ACKNOWLEDGMENT

The authors would like to acknowledge the help of Dr. Bob Liang, Dr. Amit Roy Chowdhury and Dr. Ramani Duraisami who contributed valuable comments and suggestions for this work.

REFERENCES

[1] R. Lienhart, I. Kozintsev, S. Wehr, and M. Yeung, "On the importance of exact synchronization for distributed audio processing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, April 2003.
[2] 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*, vol. ASSP-35, pp. 724–735, June 1987.
[3] 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*, pp. II–1797 – II–1800, 2002.
[4] A. J. Weiss and B. Friedlander, "Array shape calibration using sources in unknown locations-a maxilmum-likelihood approach," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1958–1966, December 1989.
[5] B. C. Ng and C. M. S. See, "Sensor-array calibration using a maxilmum-likelihood approach," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 44, pp. 827–835, June 1996.
[6] 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*, vol. 2003, pp. 348–358, March 2003.
[7] 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.
[8] 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.
[9] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, pp. 401–419, 1952.
[10] J. A. Fessler, "Mean and variance of implicitly defined biased estimators (such as penalized maximum likelihood): Applications to tomography," *IEEE Trans. on Image Processing*, vol. 5, pp. 493–506, March 1996.
[11] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 320–327, August 1976.
[12] D. P. Betrsekas, *Nonlinear Programming*. Athena Scientific, 1995.
[13] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. 1981.
[14] 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 ed., 1995.
[15] "http://www.netlib.org/minpack/."
[16] M. Steyvers, "Multideimnsional scaling," *Encyclopedia of Cognitive Science*, 2002.
[17] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, vol. Part 1. Wiley-Interscience, 2001.
[18] "http://www.portaudio.com/."
[19] "http://intel.com/technology/upnp/."