Position calibration of microphones and loudspeakers in distributed computing platforms

V.C. Raykar, I.V. Kozintsev, Rainer Lienhart

Angaben zur Veröffentlichung / Publication details:

Raykar, V.C., I.V. Kozintsev, and Rainer Lienhart. 2005. "Position calibration of microphones and loudspeakers in distributed computing platforms." *IEEE Transactions on Speech and Audio Processing* 13 (1): 70–83. https://doi.org/10.1109/tsa.2004.838540.

Position Calibration of Microphones and Loudspeakers in Distributed Computing Platforms

Vikas C. Raykar, *Student Member, IEEE*, Igor V. Kozintsev, *Member, IEEE*, and Rainer Lienhart, *Associate Member, IEEE*

***Abstract*—In this paper, we present a novel algorithm to automatically determine the relative three-dimensional (3-D) 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 nonlinear 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 and the other based on time difference of flight. 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 estimating the sensor 3-D positions are derived via the Cramér–Rao bound (CRB) 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*—Cramér–Rao bound (CRB), distributed sensor networks, microphone array calibration, multidimensional scaling, self-localizing sensor networks.**

I. Introduction

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 the 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, coworkers at meetings, family members at home are nowadays, accompanied by one or several mobile computing and communication devices like laptops, personal digital assistants (PDAs), tablets, with multiple audio and video sensors/actuators 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 multistream 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, multichannel 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 input/output (I/O) capabilities is to put sensors and actuators into a common time and space (coordinate system). In [1], [2] 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 (3-D) positions of the sensors and actuators. Many multimicrophone audio 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 [5]. In ad-hoc deployed arrays it is rather tedious and very often not accurate to get the microphone positions manually using a tape or laser devices. Also the geometry of the array may change over time frequently, either accidently, or due to redeployment. So automatic position calibration of multiple sensors and actuators is very essential. In this paper, we propose a method to automatically determine the 3-D positions of multiple microphones and speakers.

Fig. 1 shows a schematic representation of our proposed *distributed computing platform* consisting of N GPCs. Each GPC is assumed to be equipped with audio sensors (microphones),

Manuscript received August 27, 2003; revised January 19, 2004. Portions of this paper have appeared as conference papers [3] and [4]. This work was performed while V. Raykar was an Intern at Intel Laboratories, Santa Clara, CA. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Diemer de Vries.

V. C. Raykar is with the Perceptual Interfaces and Reality Laboratory, Institute of Advanced Computer Studies, University of Maryland, College Park, MD 20742 USA (e-mail: vikas@umiacs.umd.edu).

Igor V. Kozintsev is with Intel Laboratories, Intel Corporation, Santa Clara, CA 95052-8119 USA (e-mail: igor.v.kozintsev@intel.com).

R. Lienhart was with Architecture Research and Machine Learning, Intel Laboratories, Santa Clara, CA 9505–8119 USA. He is now with the Computer Science Department, University of Augsburg, Augsburg, Germany (e-mail: Rainer.Lienhart@informatik.uni-augsburg.de).

Digital Object Identifier 10.1109/TSA.2004.838540

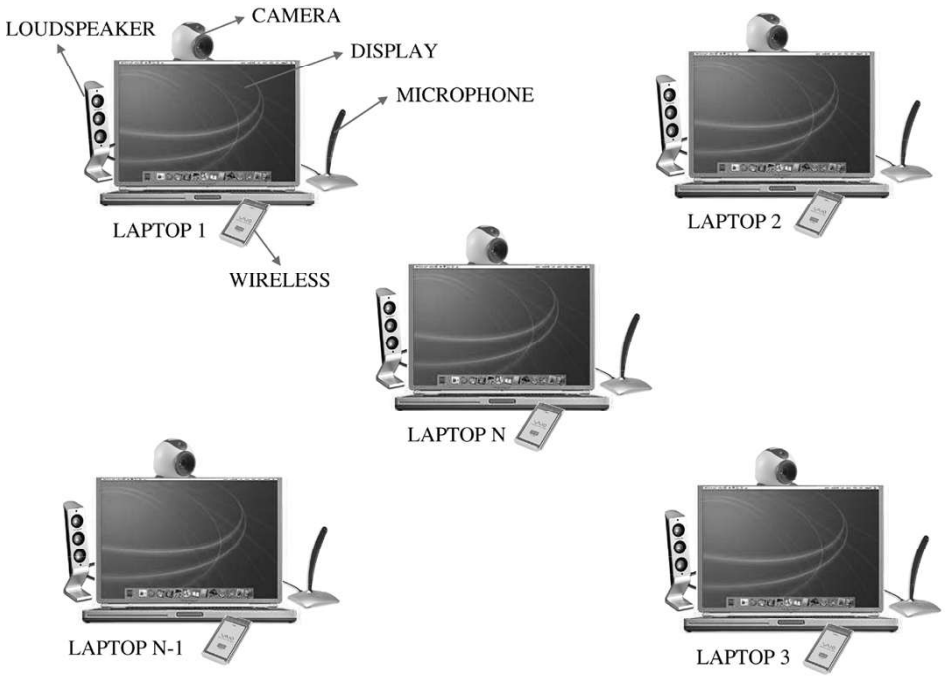

Fig. 1. Distributed computing platform consisting of N GPCs along with their onboard audio/video sensors, actuators, and wireless communication capabilities.

actuators (speakers)[1] for performing audio I/O, and wireless communication capabilities for exchanging data between each other. In practice, one GPC controls the distributed computing platform and performs the location estimation.

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. An experimental setup for automatic calibration of a large-aperture microphone array using acoustic signals from transducers whose locations are known is described in [6]. We follow a more general approach where we assume that the speaker locations are also unknown.

A lot of related theoretical work can be found in [5], and [7]–[9]. 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, the playback and the capture start time are generally unknown. A recent paper [10] accounts only for the unknown source emission time. Our solution explicitly accounts for the errors in localization due to lack of temporal synchronization among different platforms. The solution turns out to be a nonlinear 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 [10]–[15]. The problem is essentially the same as in our case, but the ranging method differ depending on the sensors and actuators.

[1]In the rest of the paper, when we refer to speaker we mean a loudspeaker and not a person speaking. We use the term speaker and loudspeaker interchangeably.

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 [6], [8], [10]. 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 optimum. We propose to use metric multidimensional scaling (MDS) [16] in order to get an approximate initial guess for the microphone and speaker locations.
- Most of the previous work on position calibration (except [12], 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 among the different platforms. We perform an analysis of the localization errors due to imprecise synchronization and propose ways to account for the unknown speaker emission start times and microphone capture start times.
- Most of the existing localization methods use the time of flight (TOF) approach for position calibration [6], [10], [12]. We show that for distributed computing platforms, the method based on time difference of flight (TDOF) can outperform the TOF method.
- We derive the approximate mean and covariance of the implicitly defined estimator using the implicit function theorem and Taylors' series expansion as in [17], [18]. We also derive the Cramér–Rao bound (CRB) and analyze the localization accuracy with respect to the number of sensors and sensor geometry.

C. Organization

The rest of this paper is organized as follows. In Section II, we formulate the problem. In Section III we derive the maximum likelihood (ML) estimator, which turns out to be a solution of a nonlinear optimization problem. In Section IV we derive an approximate closed form solution, which can be used as an initial guess for the nonlinear minimization routine. In Section V we derive the theoretical mean and covariance of the estimated parameters. The CRB is derived and analyzed as a function of the number of sensors and actuators as well as their geometry. In Section VI, extensive simulation results are reported. Section VII gives a discussion of the issues involved in designing a practical system. Section VIII, concludes with a summary of the present work.

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 relative 3-D 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 calibrated again.

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 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.[2] 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 $M \times S$ TOF measurements constitute our observations, based on which we have to estimate the microphone and speaker positions.

Let $\mathbf{m_i}$ for $i \in [1, M]$ and $\mathbf{s_j}$ for $j \in [1, S]$ be the 3-D column vectors representing the spatial coordinates of the ith microphone and jth speaker, respectively. We excite one of the S speakers at a time and measure the TOF at each of the M microphones. Let $\mathrm{TOF}_{ij}^{\mathrm{actual}}$ be the actual TOF for the ith microphone due to the jth source. Assuming a direct path the actual TOF is

$$\mathrm{TOF}_{ij}^{\mathrm{actual}} = \frac{\|\mathbf{m_i} - \mathbf{s_j}\|}{c} \tag{1}$$

where c is the speed of sound in the acoustical medium[3] and $\|\,\|$ 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 dedicated hardware or multichannel sound cards to interface multiple microphones and speakers.[4]

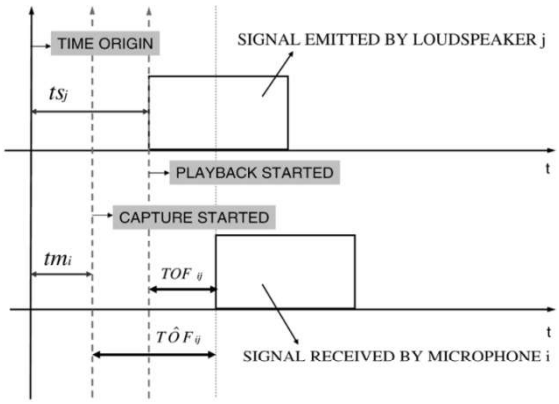

Fig. 2. Schematic indicating the unknown speaker emission start time ts_j and microphone capture start time tm_i for the ith microphone and the jth speaker.

However in a typical distributed setup of GPCs as shown in Fig. 1, 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 Fig. 2 where $\widehat{\mathrm{TOF}}_{ij}$ is what we measure and TOF_{ij} is the actual time of flight.).

According to Fig. 2 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 jth source and tm_i be the capture start time for the ith microphone with respect to some origin (Fig. 2). Incorporating these two the actual TOF now becomes

$$\begin{aligned} \widehat{\mathrm{TOF}}_{ij}^{\mathrm{actual}} &= \mathrm{TOF}_{ij}^{\mathrm{actual}} + ts_j - tm_i \\ &= \frac{\|\mathbf{m_i} - \mathbf{s_j}\|}{c} + ts_j - tm_i. \end{aligned} \tag{2}$$

[2]In some papers, TOF is referred to as time of arrival (TOA).

[3]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 degrees Celsius.

[4]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 origin can be arbitrary since $\widehat{\mathrm{TOF}}_{ij}^{\mathrm{actual}}$ 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 as our origin. We define all other times with respect to this origin.

A. TDOF

In addition to using TOF for localization we propose to use the TDOF. 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. [5] Let $\mathrm{TDOF}_{ikj}^{\mathrm{actual}}$ be the actual TDOF between the ith and the kth microphone when the jth source is excited. It is given by

$$\mathrm{TDOF}_{ikj}^{\mathrm{actual}} = \frac{||\mathbf{m_i} - \mathbf{s_j}|| - ||\mathbf{m_k} - \mathbf{s_j}||}{c}. \quad (3)$$

Including the source emission and capture start times, it becomes

$$\widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}} = \frac{||\mathbf{m_i} - \mathbf{s_j}|| - ||\mathbf{m_k} - \mathbf{s_j}||}{c} + tm_k - tm_i. \quad (4)$$

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 have reduced the number of parameters to be estimated.

III. Problem Solution

In this section, we derive the maximum likelihood estimator for the microphone and speaker locations based on the TDOF/TOF observations.

A. ML Estimate

Assuming an additive Gaussian[6] noise model for the TDOF observations we can derive the ML estimate as follows. Let Θ be a $P \times 1$ column vector, representing all the unknown non-random parameters to be estimated (microphone and speaker coordinates and microphone capture start times). Let Γ be a $N \times 1$ column vector, representing noisy TDOF measurements. Let $T(\Theta)$, be a $N \times 1$ column vector, representing the actual value of the observations. Note that $T(\Theta)$ is a function of Θ, the parameters to be estimated. Then our model for the observations is

$$\Gamma = T(\Theta) + \eta \quad (5)$$

where η is the zero-mean additive white Gaussian noise vector of length N 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

$$p(\Gamma/\Theta) = (2\pi)^{-\frac{N}{2}} |\Sigma|^{-\frac{1}{2}} \exp -\frac{1}{2}[\Gamma - T(\Theta)]^T \Sigma^{-1} [\Gamma - T(\Theta)]. \quad (6)$$

The log-likelihood function is given by

$$\ln p(\Gamma/\Theta) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| -\frac{1}{2}[\Gamma - T(\Theta)]^T \Sigma^{-1} [\Gamma - T(\Theta)]. \quad (7)$$

The ML estimate of Θ is the one which maximizes the likelihood ratio (or equivalently, the log likelihood ratio) and is given by

$$\hat{\Theta}_{\mathrm{ML}} = \arg_{\Theta}\max \mathrm{F}(\Theta, \Gamma)$$
$$F(\Theta, \Gamma) = -\frac{1}{2}[\Gamma - T(\Theta)]^T \Sigma^{-1} [\Gamma - T(\Theta)]. \quad (8)$$

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

$$\hat{\Theta}_{\mathrm{ML}} = \arg_{\Theta} \min[\hat{F}_{\mathrm{TDOF}}(\Theta, \Gamma)]$$
$$\hat{F}_{\mathrm{TDOF}}(\Theta, \Gamma) = \sum_{j=1}^{S}\sum_{i=1}^{M-1}\sum_{k=i+1}^{M} \frac{\left(\widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{estimated}} - \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}\right)^2}{\sigma_{ikj}^2}. \quad (9)$$

Θ is a vector of the parameters to be estimated, i.e., the microphone and the source coordinates and the microphone capture start times. Note that when use TDOF the speaker emission start time gets canceled.

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

$$\hat{\Theta}_{\mathrm{ML}} = \arg_{\Theta} \min[\hat{F}_{\mathrm{TOF}}(\Theta, \Gamma)]$$
$$\hat{F}_{\mathrm{TOF}}(\Theta, \Gamma) = \sum_{j=1}^{S}\sum_{i=1}^{M} \frac{\left(\widehat{\mathrm{TOF}}_{ij}^{\mathrm{estimated}} - \widehat{\mathrm{TOF}}_{ij}^{\mathrm{actual}}\right)^2}{\sigma_{ij}^2}. \quad (10)$$

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

B. Reference Coordinate System

Cost function, defined in (9), can have multiple global minima since the TOF and TDOF depends on pairwise distances. Any translation and rotation of the coordinate system does not change the value of the function to be minimized. In order to eliminate multiple global minima, we need to setup a reference coordinate system. 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 $(x_2, y_2, 0)$. We are fixing a plane so that the sensor configuration cannot be translated or rotated. Similarly, 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

[5] 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.

[6] We estimate the TDOF or TOF using generalized cross correlation (GCC) [19]. 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 [19].

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	$(M-1)S$	$(D+1)M + DS - \frac{D(D+1)}{2} - 1$

2-D) we specify one more node to lie in the positive Z-axis (or positive Y-axis in 2-D). Also the reflections along the X-axis and Y-axis (for 3-D) are eliminated by assuming that the nodes that form the coordinate system 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 also multiple global minima of the cost function due to shifts in the time axis. Similar to fixing a reference coordinate system in space we introduce a reference time line by setting $tm_1 = 0$.

C. Nonlinear Least Squares

The ML estimate for the node coordinates of the microphones and speakers is implicitly defined as the minimum of the nonlinear function (9). This function can be minimized using generic numerical optimization methods. However, there exist specialized methods like the Gauss–Newton and the Levenberg–Marquardt methods that are often more efficient in practice. The Levenberg–Marquardt method [20] is a popular method for solving nonlinear least squares problems. For more details on nonlinear minimization refer to [21]. Appendix I gives the non zero partial derivatives needed for the minimization routines.[7]

D. Minimum Number of Microphones and Speakers Required

The total number of observations should be greater than or equal to the total number of parameters to be estimated. This defines 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 $(M-1)S$ out of $M(M-1)S/2$ pair of TDOF measurements are linearly independent. In Table I, TOF/TDOF Position refers to the case where we are estimating only the positions of the microphones and speakers, i.e., the TOF/TDOF is not corrupted by the capture and the emission start times. TOF/TDOF Joint refers to the case where we are jointly estimating the emission and capture start times along with the microphone and speaker coordinates.

[7]Many software solutions are available for the Levenberg–Marquardt method such as *lsqnonlin* in MATLAB, *mrqmin* provided by Numerical Recipes in C [22], and the MINPACK-1 routines [23]

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

Assuming $M = S = K$, the Table II lists the minimum K required for the estimation procedure. Assuming each GPC has one microphone and one speaker this gives the minimum number of GPCs required.

IV. CLOSED FORM APPROXIMATE SOLUTION

The common problem with minimization methods is that they often get stuck in a local minima. In this section, we derive an approximate closed form solution, which can be used to initialize the minimization routine.

A. Initial Guess for Capture and Emission Start Times

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

$$\begin{aligned}
\widehat{\mathrm{TOF}}_{ii} &= \mathrm{TOF}_{ii} + ts_i - tm_i \\
\widehat{\mathrm{TOF}}_{jj} &= \mathrm{TOF}_{jj} + ts_j - tm_j \\
\widehat{\mathrm{TOF}}_{ij} &= \mathrm{TOF}_{ij} + ts_j - tm_i \\
\widehat{\mathrm{TOF}}_{ji} &= \mathrm{TOF}_{ji} + ts_i - tm_j.
\end{aligned} \tag{11}$$

Assuming sufficient closeness between the microphone and speaker on the same GPC compared to the distance between two GPCs, the following approximations can be made:

$$\begin{aligned}
\mathrm{TOF}_{ii} &\approx \mathrm{TOF}_{jj} \approx 0 \\
\mathrm{TOF}_{ij} &\approx \mathrm{TOF}_{ji}.
\end{aligned} \tag{12}$$

We are making the assumption that the microphone and the speaker on the same GPC can be considered as one node. Substituting, we have the following equations:

$$\begin{aligned}
\widehat{\mathrm{TOF}}_{ii} &\approx ts_i - tm_i \\
\widehat{\mathrm{TOF}}_{jj} &\approx ts_j - tm_j \\
\widehat{\mathrm{TOF}}_{ij} &\approx \mathrm{TOF}_{ij} + ts_j - tm_i \\
\widehat{\mathrm{TOF}}_{ji} &\approx \mathrm{TOF}_{ij} + ts_i - tm_j.
\end{aligned} \tag{13}$$

TABLE III
ALGORITHM [M MICROPHONES AND S SPEAKERS]

- **STEP 1**:
  - *Measure the $M \times S$ Time Of Flight ($\hat{TOF}$) matrix.*
- **STEP 2**:
  - *Form the approximate distance matrix D. (Equation 16)*
  - *Assume $tm_1 = 0$ (microphone on which capture was started first) and get the approximate microphone capture and speaker emission start times. (Equation 15)*
  - *Convert the distance matrix D to the dot product matrix B (Appendix II). 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 the TDOF based error function (Equation 9) 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.*

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

$$\mathrm{TOF}_{ij} \approx \frac{(\widehat{\mathrm{TOF}}_{ij} + \widehat{\mathrm{TOF}}_{ji}) - (\widehat{\mathrm{TOF}}_{ii} + \widehat{\mathrm{TOF}}_{jj})}{2}. \quad (14)$$

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

$$ts_i \approx \widehat{\mathrm{TOF}}_{ii} + tm_i$$
$$tm_j \approx \frac{(\widehat{\mathrm{TOF}}_{ij} - \widehat{\mathrm{TOF}}_{ji}) + (\widehat{\mathrm{TOF}}_{ii} - \widehat{\mathrm{TOF}}_{jj})}{2} + tm_i. \quad (15)$$

Setting 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

Multidimensional Scaling (MDS): Given the pairwise Euclidean distances between N nodes their relative positions can be determined by means of metric MDS [16]. 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 [24]. 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 [16] can exactly recreate the configuration. Given a set of N GPCs, let X be a $N \times 3$ matrix where each row represents the 3-D 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., at most three. Based on the rank of B we can find whether all GPCs are on a plane (or even line) or distributed in 3-D. Starting with a matrix B (in practice 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) [22], 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 $s_1 \geq s_2 \geq s_r > s_{r+1} = \cdots = s_N = 0$, 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 ijth element is the Euclidean distance between the ith and the jth 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 [16] 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

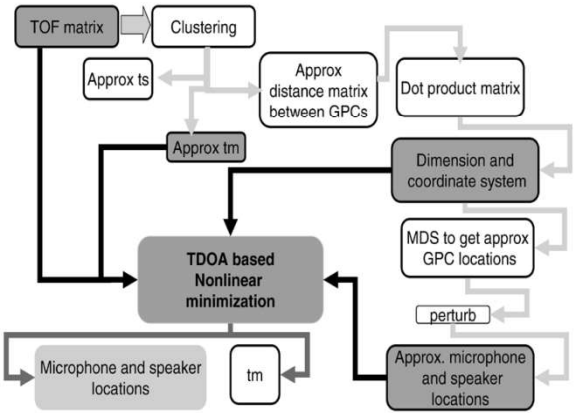

Fig. 3. Flowchart of the complete algorithm.

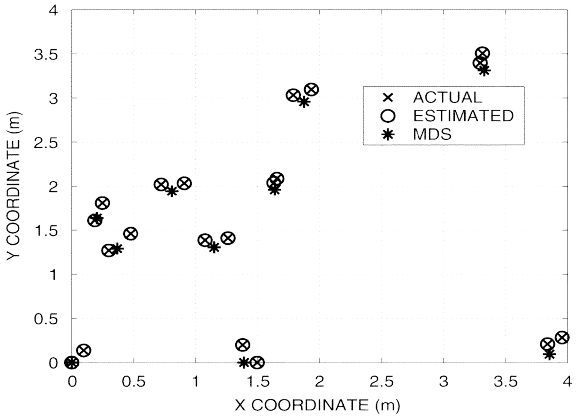

Fig. 4. Results in two dimensions for a network consisting of ten GPCs each having one microphone and one speaker. The actual locations of the sensors and actuators are shown as "x." The "*"s are the approximate GPC locations as determined by MDS. The estimated locations are further refined by ML estimation and marked as "o"s.

the cosine law, which relates the distance between two vectors to their lengths and the cosine of the angle between them. Refer to Appendix II for a detailed derivation of how to convert the distance matrix to the scalar product matrix.

Multidimensional Scaling With Clustering: In our case of M microphones and S speakers we cannot use MDS directly because 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 ith and jth GPC is given by

$$d_{ij} \approx \frac{c(\widehat{\mathrm{TOF}}_{ij} + \widehat{\mathrm{TOF}}_{ji} - \widehat{\mathrm{TOF}}_{ii} - \widehat{\mathrm{TOF}}_{jj})}{2} \tag{16}$$

where c is the speed of the sound.

The positions estimated by MDS are obtained with respect to the centroid as the origin and an arbitrary orientation. They are therefore converted into the reference coordinate system described in Section III-B. The approximate locations of the GPCs are also slightly perturbed to get the initial guess for the microphone and speaker locations, which are further refined by the nonlinear-minimization routine. Table III summarizes the complete algorithm and Fig. 3, shows the flowchart.

Fig. 4 shows an example with 10 GPCs 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 refined by ML estimation and marked as "o"s.

V. 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 [see (8)]. 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. In this section, we derive the approximate expressions for both the mean and variance of the estimators. We follow the same approach as in [17], [18].

In further derivation, we need the first and second derivatives of (8) with respect to Θ and Γ. Using the generalized chain rule it can be shown that for (8) the vector derivatives are as follows:

$$\begin{aligned}
\nabla_{\Theta} F(\Theta,\Gamma) &= J^T \Sigma^{-1}(\Gamma - T(\Theta)) \\
\nabla_{\Gamma} F(\Theta,\Gamma) &= -\Sigma^{-1}(\Gamma - T(\Theta)) \\
\nabla_{\Theta}\nabla_{\Theta} F(\Theta,\Gamma) &= -J^T \Sigma^{-1} J \\
\nabla_{\Theta}\nabla_{\Gamma} F(\Theta,\Gamma) &= J^T \Sigma^{-1}
\end{aligned} \tag{17}$$

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

$$[J]_{ij} = \frac{\partial t_i(\Theta)}{\partial \theta_j}. \tag{18}$$

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

A. Estimator Covariance

In this section we use the Taylors' 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 (8). The maximum can be found by setting the first derivative to zero, i.e.,

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

where $\mathbf{0}$ is a zero column vector of length P. The implicit function theorem guarantees that (19) implicitly defines a vector valued function $\hat{\Theta} = h(\Gamma) = [h_1(\Gamma), h_1(\Gamma), \ldots, h_P(\Gamma)]^T$ that

maps the observation vector Γ to the parameter vector $\hat{\Theta}$. Equation (19) can be written as

$$\nabla_\Theta F(h(\Gamma), \Gamma) = \mathbf{0}. \tag{20}$$

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
Γ. Then expanding $h(\Gamma)$ around Γ_m we get

$$h(\Gamma) \approx h(\Gamma_m) + [\nabla_\Gamma h(\Gamma)^T|_{\Gamma=\Gamma_m}]^T(\Gamma - \Gamma_m) \tag{21}$$

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

$$\mathrm{Cov}(h(\Gamma)) \approx [\nabla_\Gamma h(\Gamma)^T|_{\Gamma=\Gamma_m}]^T \mathrm{Cov}(\Gamma)[\nabla_\Gamma h(\Gamma)^T|_{\Gamma=\Gamma_m}]. \tag{22}$$

Note, we do not know $h(\Gamma)$. Differentiating (20), with respect to Γ and evaluating at Γ_m

$$\nabla_\Theta\nabla_\Theta F(h(\Gamma_m), \Gamma_m)[\nabla_\Gamma h(\Gamma_m)^T]^T + \nabla_\Theta\nabla_\Gamma F(h(\Gamma_m).\Gamma_m) = \mathbf{0}. \tag{23}$$

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

$$[\nabla_\Gamma h(\Gamma_m)^T]^T = -[\nabla_\Theta\nabla_\Theta F(h(\Gamma_m), \Gamma_m)]^{-1} \times \nabla_\Theta\nabla_\Gamma F(h(\Gamma_m), \Gamma_m). \tag{24}$$

Substituting from (17) we get

$$[\nabla_\Gamma h(\Gamma_m)^T]^T = -[-J^T\Sigma^{-1}J]^{-1}J^T\Sigma^{-1}. \tag{25}$$

Using this in the covariance expression, we finally arrive at

$$\mathrm{Cov}\hat{\Theta} \approx [J^T\Sigma^{-1}J]^{-1}. \tag{26}$$

B. Estimator Mean

Taking the expectation of the first-order Taylors' series expansion in (21)

$$E(h(\Gamma)) \approx h(\Gamma_m) = h(T) \tag{27}$$

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 Taylors' series expansion, but it involves third-order derivatives and generally we cannot get simple form as in (26).

C. CRB

The CRB gives a lower bound on the variance of *any* unbiased estimate [25]. It does not depend on the particular estimation method used. In this section, we derive the CRB assuming our estimator is unbiased. The variance of any unbiased estimator $\hat{\Theta}$ of Θ is bounded as [25]

$$E[(\hat{\Theta} - \Theta)(\hat{\Theta} - \Theta)^T] \geq F^{-1}(\Theta) \tag{28}$$

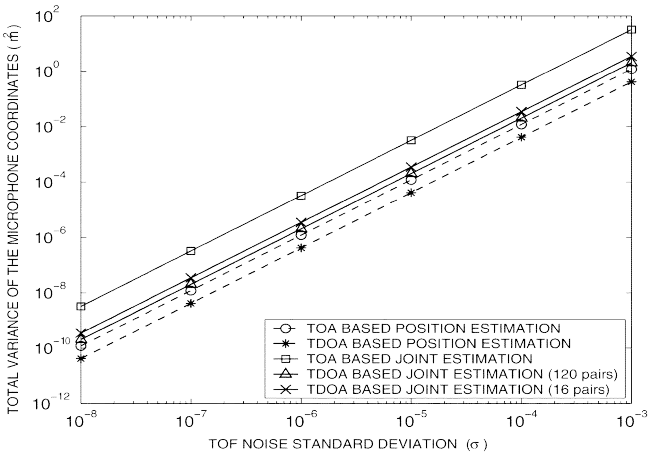

Fig. 5. CRB on the total variance of the unknown microphone coordinates as a function of TOF noise standard deviation σ for different estimation procedures. For the TDOF-based method the noise variance was taken as twice that of the TOF variance. The network had a total of 16 microphones and 16 speakers.

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

$$F(\Theta) = E\{[\nabla_\Theta \ln p(\Gamma/\Theta)][\nabla_\Theta \ln p(\Gamma/\Theta)]^T\}. \tag{29}$$

The derivative of the log-likelihood function [see (7)] can be found using the generalized chain rule and is given by [refer (17)]

$$\nabla_\Theta \mathrm{lnp}(\Gamma/\Theta) = \mathrm{J}^\mathrm{T}\Sigma^{-1}(\Gamma - \mathrm{T}) \tag{30}$$

where J is the *Jacobian*. Substituting this in (29) and taking the expectation the Fishers information matrix is

$$F = J^T\Sigma^{-1}J \tag{31}$$

$$\mathrm{Cov}\hat{\Theta} \geq [J^T\Sigma^{-1}J]^{-1}. \tag{32}$$

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

D. Rank of the Fisher Information Matrix

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

$$\mathrm{Cov}[\hat{\Theta}] \geq \frac{1}{\sigma^2}[J^TJ]^{-1} = F^{-1} \tag{33}$$

where $F = \sigma^2 J^T J$. 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.

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. Fig. 5, shows CRB on the total variance (sum of the individual variances) 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]

E. 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 CRB can be seen from Fig. 5 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. TOF has more nuisance parameters and hence it has a higher bound on 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. Fig. 5 demonstrates that the bound on the estimation variance decreases as the number of used TDOF measurements grows.

F. 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. Fig. 6(a) shows the 95% uncertainty ellipses for a regular 2-D array consisting of nine microphones and nine 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. Fig. 6(b) shows the corresponding 95% uncertainty ellipses for a 2-D 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.

G. Sensor Geometry—How to Select a Good 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. Fig. 7 shows the 95% uncertainty ellipses for a regular 2-D array containing 25 microphones and 25 speakers for different positions of the known nodes. In Fig. 7(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 perpendicular 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 Fig. 7(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 Fig. 7(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

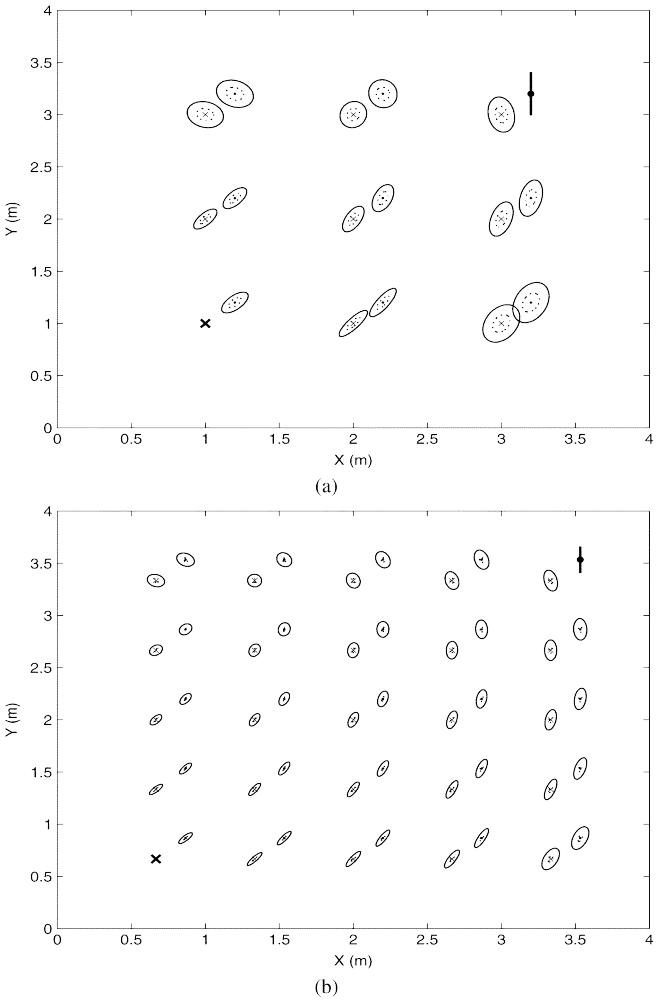

Fig. 6. 95% uncertainty ellipses for a regular 2-D array of (a) nine speakers and nine microphones. (b) 25 speakers and 25 microphones. Noise variance in both cases is $\sigma^2 = 10^{-9}$ for the TOF-based method and $\sigma^2 = 2 \times 10^{-9}$ for the TDOF-based method. The microphones are represented as crosses ($\times$) and the speakers as dots ($\cdot$). 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.

[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.

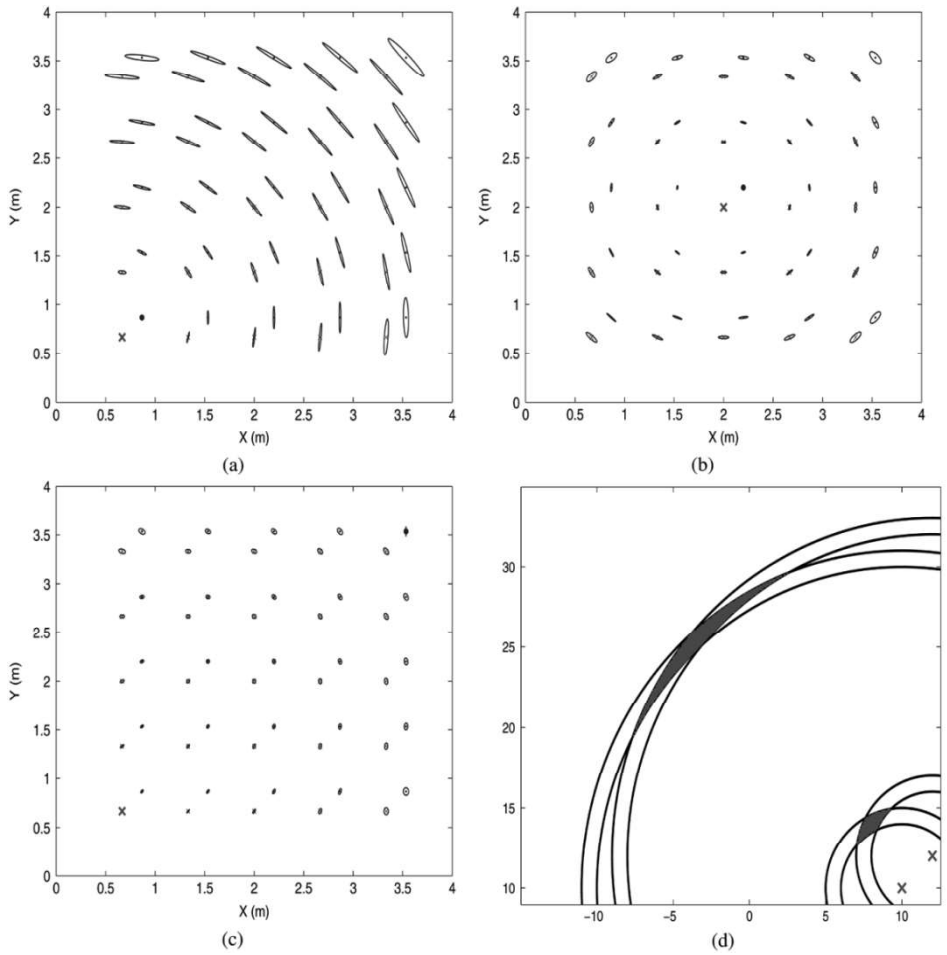

Fig. 7. 95% uncertainty ellipses for a regular 2-D array of 25 microphones and 25 speakers for different positions of the known microphone and for different x coordinates of the known speaker. In (a) and (b) the known nodes are close to each other and in (c) they are spread out one at each corner of the grid. The microphones are represented as crosses ($\times$) and the speakers as dots ($\cdot$). 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.

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 Fig. 7(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 3-D) 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.

VI. 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 4.0 m $\times$ 4.0 m $\times$ 4.0 m. 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. Fig. 8(a) and (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 CRB 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 lower bound on estimation variance, experimentally all the joint estimation procedures showed almost the same variance. The estimator is unbiased for low noise variances.

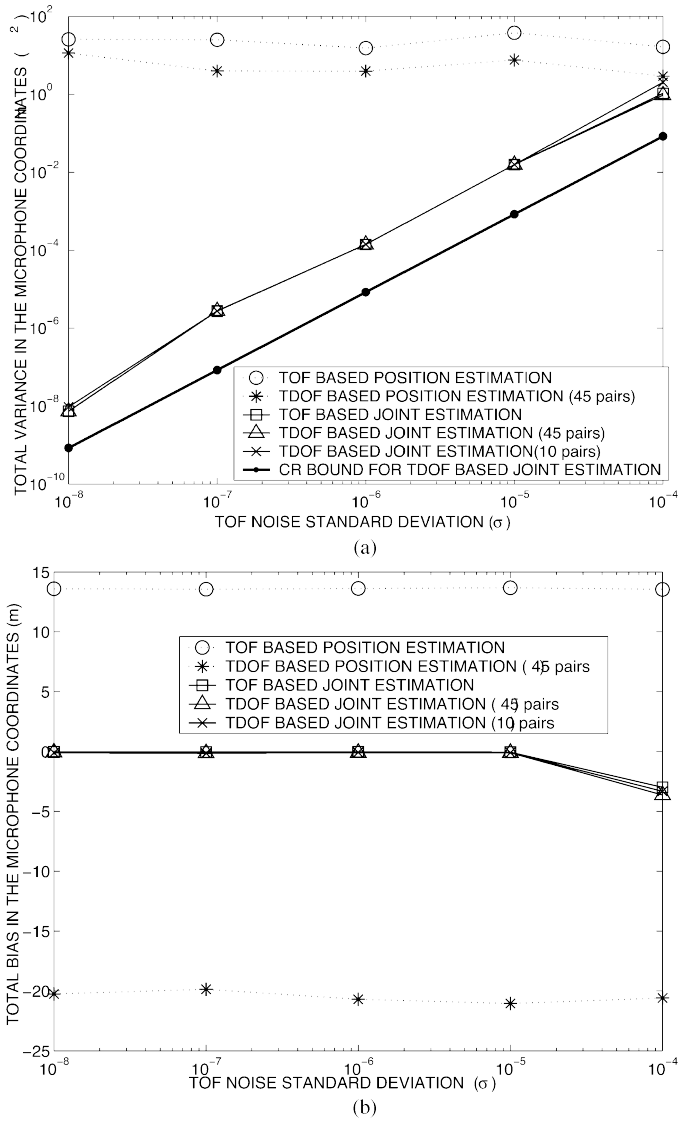

Fig. 8. (a) Total variance. (b) Total bias of all the microphone coordinates for increasing TOF noise standard deviation σ. The sensor network consisted of 16 microphones and 16 speakers. The results are shown for both the TOF and TDOF-based position and joint estimation. The CRB for the TDOF-based joint estimation is also plotted. For the TDOF-based method the noise variance was taken as twice that of the TOF variance.

VII. 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.

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

$$s(t) = A\cos\left(2\pi\left(f_0 + \left(\frac{f_1 - f_0}{T}\right)t\right)\right) \quad 0 \le t \le T. \tag{34}$$

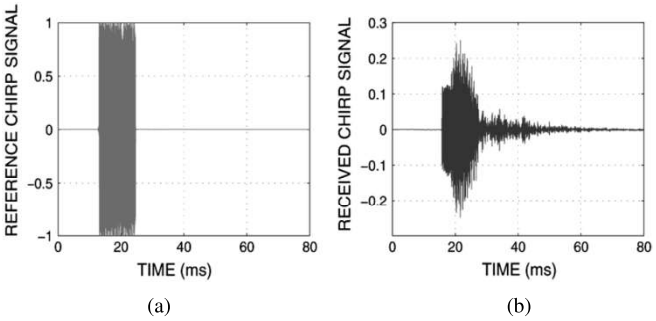

Fig. 9. (a) Loopback reference chirp signal. (b) Chirp signal received by one of the microphones.

In our system, we used the chirp signal of 512 samples at 44.1 kHz (11.61 ms) as our calibration signal. The instantaneous frequency 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. Fig. 9(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. Fig. 9(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 spread 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, a 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 [19] developed an 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 [19]. 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 [19]

$$R_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} W(\omega)X_1(\omega)X_2^*(\omega)e^{j\omega\tau}d\omega \tag{35}$$

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 [19]. 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

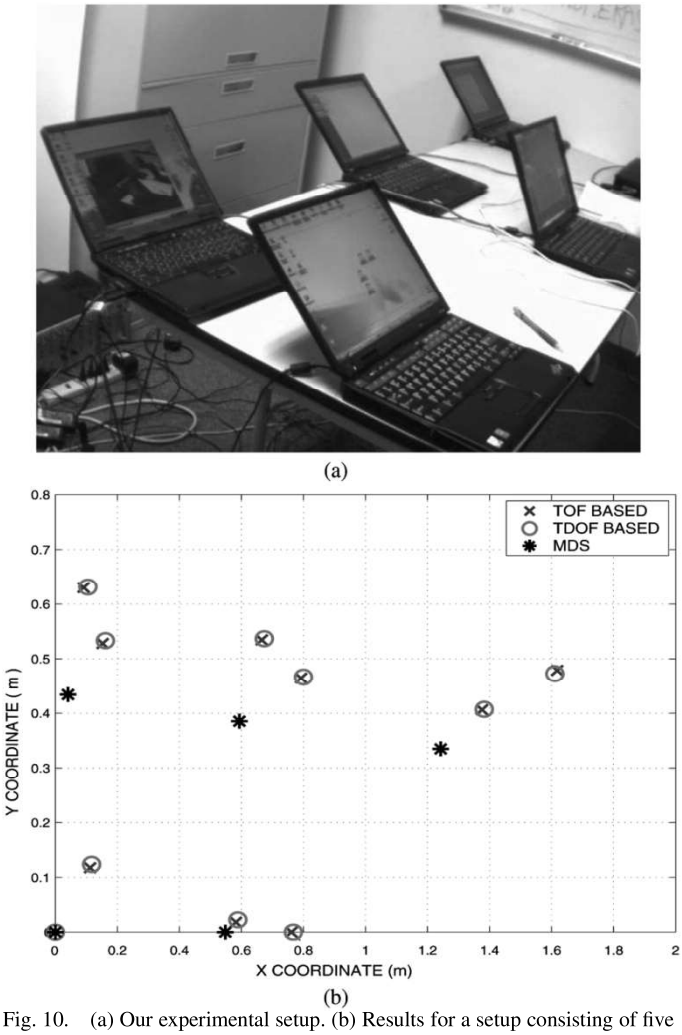

Fig. 10. (a) Our experimental setup. (b) Results for a setup consisting of five laptops each having one internal microphone and speakers.

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

$$W_{\mathrm{PHAT}}(\omega) = \frac{1}{|X_1(\omega)X_2^*(\omega)|}. \tag{36}$$

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 five laptops (IBM T-series Thinkpads with Intel Pentium series processors). Fig. 10(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 2-D 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. Capture and play back was done using the free, cross platform, open-source, audio I/O library Portaudio [26]. Most of the signal processing tasks were implemented using the Intel Integrated Performance Primitives (IPPs). For the nonlinear minimization we used the *mrqmin* routine from Numerical Recipes in C [22]. For the distributed platform we used the Universal Plug and Play (UPnP) [27] technology to form an ad hoc 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 [27]. For the setup consisting of five microphones and five speakers, Fig. 10(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 6.2 cm. We also implemented the same system on a synchronized platform for which the error was 3.8 cm.

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

VIII. Conclusion

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 3-D 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
Derivatives

Following are the derivatives which are needed for the minimization routine and calculation of the covariance matrix. These derivatives form the nonzero elements of the Jacobian matrix

$$\begin{aligned}
\frac{\partial \widehat{\mathrm{TOF}}_{ij}^{\mathrm{actual}}}{\partial mx_i} &= -\frac{\partial \widehat{\mathrm{TOF}}_{ij}^{\mathrm{actual}}}{\partial sx_j} = \frac{mx_i - sx_j}{c\|m_i - s_j\|} \\
\frac{\partial \widehat{\mathrm{TOF}}_{ij}^{\mathrm{actual}}}{\partial my_i} &= -\frac{\partial \widehat{\mathrm{TOF}}_{ij}^{\mathrm{actual}}}{\partial sy_j} = \frac{my_i - sy_j}{c\|m_i - s_j\|} \\
\frac{\partial \widehat{\mathrm{TOF}}_{ij}^{\mathrm{actual}}}{\partial mz_i} &= -\frac{\partial \widehat{\mathrm{TOF}}_{ij}^{\mathrm{actual}}}{\partial sz_j} = \frac{mz_i - sz_j}{c\|m_i - s_j\|} \\
\frac{\partial \widehat{\mathrm{TOF}}_{ij}^{\mathrm{actual}}}{\partial ts_j} &= -\frac{\partial \widehat{\mathrm{TOF}}_{ij}^{\mathrm{actual}}}{\partial tm_i} = 1 \\
\frac{\partial \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}}{\partial mx_i} &= \frac{mx_i - sx_j}{c\|m_i - s_j\|} \\
\frac{\partial \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}}{\partial mx_k} &= -\frac{mx_k - sx_j}{c\|m_k - s_j\|}
\end{aligned} \tag{37}$$

$$\frac{\partial \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}}{\partial my_i} = \frac{my_i - sy_j}{c\|m_i - s_j\|}$$
$$\frac{\partial \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}}{\partial my_k} = -\frac{my_k - sy_j}{c\|m_k - s_j\|}$$
$$\frac{\partial \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}}{\partial mz_i} = \frac{mz_i - sz_j}{c\|m_i - s_j\|}$$
$$\frac{\partial \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}}{\partial mz_k} = -\frac{mz_k - sz_j}{c\|m_k - s_j\|}$$
$$\frac{\partial \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}}{\partial sx_j} = -\frac{mx_i - sx_j}{c\|m_i - s_j\|} + \frac{mx_k - sx_j}{c\|m_k - s_j\|}$$
$$\frac{\partial \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}}{\partial sy_j} = -\frac{my_i - sy_j}{c\|m_i - s_j\|} + \frac{my_k - sy_j}{c\|m_k - s_j\|}$$
$$\frac{\partial \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}}{\partial sz_j} = -\frac{mz_i - sz_j}{c\|m_i - s_j\|} + \frac{mz_k - sz_j}{c\|m_k - s_j\|}$$
$$\frac{\partial \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}}{\partial tm_k} = -\frac{\partial \widehat{\mathrm{TDOF}}_{ikj}^{\mathrm{actual}}}{\partial tm_i} = 1. \quad (38)$$

Appendix II
Converting the Distance Matrix to a Dot Product Matrix

Let us say we choose the kth GPC as the origin of our coordinate system. Let d_{ij} and b_{ij} be the distance and dotproduct respectively, between the ith and the jth GPC. Referring to Fig. 11, using the cosine law

$$d_{ij}^2 = d_{ki}^2 + d_{kj}^2 - 2d_{ki}d_{kj}\cos(\alpha). \quad (39)$$

The dot product b_{ij} is defined as

$$b_{ij} = d_{ki}d_{kj}\cos(\alpha). \quad (40)$$

Combining the above two equations

$$b_{ij} = \frac{1}{2}\left(d_{ki}^2 + d_{kj}^2 - d_{ij}^2\right). \quad (41)$$

However, this is with respect to the kth 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 kth 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

$$X^* = X - \frac{1}{N}\mathbf{1}_{N\times N}X \quad (42)$$

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:

$$B^* = X^*X^{*T}$$
$$= B - \frac{1}{N}B\mathbf{1}_{N\times N} - \frac{1}{N}\mathbf{1}_{N\times N}B + \frac{1}{N^2}\mathbf{1}_{N\times N}B\mathbf{1}_{N\times N}.$$

Hence, the ijth element in B^* is given by

$$b_{ij}^* = b_{ij} - \frac{1}{N}\sum_{l=1}^{N} b_{il} - \frac{1}{N}\sum_{m=1}^{N} b_{mj} + \frac{1}{N^2}\sum_{o=1}^{N}\sum_{p=1}^{N} b_{op}. \quad (43)$$

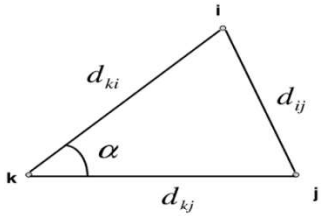

Fig. 11. Law of cosines.

Substituting (41) we get

$$b_{ij}^* = -\frac{1}{2}\left[d_{ij}^2 - \frac{1}{N}\sum_{l=1}^{N} d_{il}^2 - \frac{1}{N}\sum_{m=1}^{N} d_{mj}^2 + \frac{1}{N^2}\sum_{o=1}^{N}\sum_{p=1}^{N} d_{op}^2\right].$$

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 $-(1/2)$.

Acknowledgment

The authors wish to acknowledge the help of B. Liang, A. R. Chowdhury, R. Duraiswami, and R. Chellappa 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. Acoustics, Speech, Signal Processing*, Apr. 2003, pp. IV-840–IV-843.

[2] R. Lienhart, I. Kozintsev, and S. Wehr, "Universal synchronization scheme for distributed audio-video capture on heterogenous computing platforms," in *Proc. ACM Multimedia*, Nov. 2003, pp. 263–266.

[3] V. C. Raykar, I. Kozintsev, and R. Lienhart, "Position calibration of audio sensors and actuators in a distributed computing platform," in *Proc. ACM Multimedia*, Nov. 2003, pp. 572–581.

[4] ——, "Self localization of acoustic sensors and actuators on distributed platforms," in *Proc. Int. Workshop Multimedia Technologies E-Learning Collaboration*, Oct. 2003.

[5] Y. Rockah and P. M. Schultheiss, "Array shape calibration using sources in unknown locations part II: Near-field sources and estimator implementation," *IEEE Trans. Acous., Speech, Signal Process.*, vol. ASSP-35, no. 6, pp. 724–735, Jun. 1987.

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

[7] Y. Rockah and P. M. Schultheiss, "Array shape calibration using sources in unknown locations part I: Far-field sources," *IEEE Trans. Acous., Speech Signal Process.*, vol. ASSP-35, no. 3, pp. 286–299, Mar. 1987.

[8] A. J. Weiss and B. Friedlander, "Array shape calibration using sources in unknown locations-a maxilmum-likelihood approach," *IEEE Trans. Acous., Speech, Signal Process.*, vol. 37, pp. 1958–1966, Dec. 1989.

[9] B. C. Ng and C. M. S. See, "Sensor-array calibration using a maximum-likelihood approach," *IEEE Trans. Acous., Speech, Signal Process.*, vol. 44, no. 6, pp. 827–835, Jun. 1996.

[10] R. Moses, D. Krishnamurthy, and R. Patterson, "A self-localization method for wireless sensor networks," *Eurasip J. Appl. Signal Process. Special Issue on Sensor Networks*, vol. 2003, pp. 348–358, Mar. 2003.

[11] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc wireless sensor networks," in *Proc. 5th Int. Conf. Mobile Computing Networking*, Jul. 2001, pp. 166–179.

[12] L. Girod, V. Bychkovskiy, J. Elson, and D. Estrin, "Locating tiny sensors in time and space: A case study," in *Proc. IEEE Int. Conf. Computer Design*, Sep. 2002, pp. 214–219.

[13] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, "Scalable coordination for wireless sensor networks: Self-configuring localization systems," in *Proc. 6th Int. Symp. Communication Theory Applications*, Jul. 2001, pp. II-1797–II-1800.

[14] K. Whitehouse and D. Culler, "Calibration as parameter estimation in sensor networks," in *Proc. 1st ACM Int. Workshop Sensor Networks Applications*, Sep. 2002, pp. 59–67.
[15] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach, "Robotics-based location sensing using wireless Ethernet," in *Proc. 8th ACM Int. Conf. Mobile Computing Networking (MOBICOM)*, Atlanta, GA, Sep. 2002, pp. 227–238.
[16] W. S. Torgerson, "Multidimensional scaling: I. Theory and method," *Psychometrika*, vol. 17, pp. 401–419, 1952.
[17] J. A. Fessler, "Mean and variance of implicitly defined biased estimators (such as penalized maximum likelihood): Applications to tomography," *IEEE Trans. Image Process.*, vol. 5, no. 10, pp. 493–506, Mar. 1996.
[18] A. R. Chowdhury and R. Chellappa, "Statistical bias and the accuracy of 3d reconstruction from video," *Int. J. Comput. Vis.*, vol. 55, pp. 27–53, Oct. 2003.
[19] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acous., Speech, Signal Process.*, vol. ASSP-24, no. 4, pp. 320–327, Aug. 1976.
[20] D. P. Betrsekas, *Nonlinear Programming*. Cambridge, MA: Athena Scientific, 1995.
[21] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. San Diego, CA: Academic, 1981.
[22] H. P. Press, S. A. Teukolsky, W. T. Vettring, and B. P. Flannery, *Numerical Recipes in C the Art of Scientific Computing*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1995.
[23] Univ. Tenn., Knoxville, TN, Oak Ridge Nat. Lab., Oak Ridge, TN [Online]. Available: http://www.netlib.org/minpack/
[24] M. Steyvers, "Multideimnsional scaling," *Encyclopedia of Cognitive Science*, 2002.
[25] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. New York: Wiley, 2001, vol. Part 1.
[26] [Online]. Available: http://www.portaudio.com/
[27] Intel Corp., Santa Clara, CA [Online]. Available: http://intel.com/technology/upnp/

Vikas C. Raykar (S'00) received the B.S. degree in electronics and communication engineering from the Regional Engineering College, Trichy, India, in 2001, the M.S. degree in electrical engineering the University of Maryland, College Park, in 2003, and is currently pursuing the Ph.D. degree in computer science from the same university.

He is currently working as a Research Assistant at the Perceptual Interfaces and Reality Laboratory, Institute of Advanced Computer Studies, University of Maryland, College Park. He has published seven papers in major conferences. His research interests broadly span computer audition/vision and specifically include spatial audio, auditory source localization and microphone/camera array calibration.

Igor V. Kozintsev (M'00) received the B.S. degree (with honors) in electrical engineering from the Moscow State Technical University, Moscow, Russia, in 1994, and the M.S. and Ph.D. degrees in electrical engineering, from the University of Illinois at Urbana-Champaign (UIUC), in 1997 and 2000, respectively.

Since 1996, he has been a Research Assistant at the Image Formation and Processing Laboratory, the Beckman Institute for Advanced Science and Technology, UIUC. In May 2000, he joined Intel Laboratories, Intel Corporation, Santa Clara, CA, where he currently holds the position of a Senior Researcher. His research interests include multimedia processing, digital signal processing, wireless communications, and networking.

Rainer Lienhart (S'97–A'98) received the Ph.D. degree in computer science from the University of Mannheim (UM), Mannheim, Germany, in 1998.

Currently, he is a Full Professor at the University of Augsburg, Augsburg, Germany, heading the Multimedia Computing Laboratory. From 1998 to 2004, he was a Staff Researcher at Intel Corporation, the Microprocessor Research Laboratory, Santa Clara, CA, where he worked on transforming a network of heterogeneous, distributed computing platforms into an array of audio/video sensors and actuators capable of performing complex DSP tasks such as distributed beamforming, audio rendering, audio/visual tracking, and camera array processing. At the same time, he also continued his work on media mining, where he is well-known for his work in video content analysis with contributions in text detection/recognition, commercial detection, face detection, shot and scene detection, and automatic video abstraction. While a student at UM, he was a member of the Movie Content Analysis Project (MoCA). He has published over 50+ papers in major conferences and journals and filed 20+ patents. His research interests include image/video/audio content analysis, machine learning, scalable signal processing, scalable learning, scalable and adaptive algorithms, ubiquitous and distributed media computing in heterogeneous networks, and peer-to-peer networking and mass media sharing.