Position Calibration of Audio Sensors and Actuators in a Distributed Computing Platform

Vikas C. Raykar*
vikas@umiacs.umd.edu

Igor Kozintsev
igor.kozintsev@intel.com

Rainer Lienhart
rainer.lienhart@intel.com

Intel Labs, Intel Corporation, Santa Clara, CA, USA

ABSTRACT

In this paper, we present a novel approach to automatically determine the positions of sensors and actuators in an ad-hoc distributed network of general purpose computing platforms. The formulation and solution accounts for the limited precision in temporal synchronization among multiple platforms. The theoretical performance limit for the sensor positions is derived via the Cramér-Rao bound. We analyze the sensitivity of localization accuracy with respect to the number of sensors and actuators as well as their geometry. Extensive Monte Carlo simulation results are reported together with a discussion of the real-time system. In a test platform consisting of 4 speakers and 4 microphones, the sensors' and actuators' three dimensional locations could be estimated with an average bias of 0.08 cm and average standard deviation of 3.8 cm.

Categories and Subject Descriptors

C.3 [**Special purpose and application based systems**]: Signal processing systems; C.2.4 [**Distributed Systems**]: Distributed applications; G. 3 [**Probability and statistics**]: Probabilistic algorithms

General Terms

Theory, Algorithms, Design, Experimentation, Performance

Keywords

Self-localization, Sensor networks, Position calibration, Microphone array calibration, Multidimensional Scaling, Cramér-Rao bound

*The author is with the Perceptual Interfaces and Reality Laboratory, University of Maryland, College Park, MD, USA. The paper was written while the author was an Intern at Intel Labs, Intel Corporation, Santa Clara, CA, USA.

MM'03, November 2–8, 2003, Berkeley, California, USA.

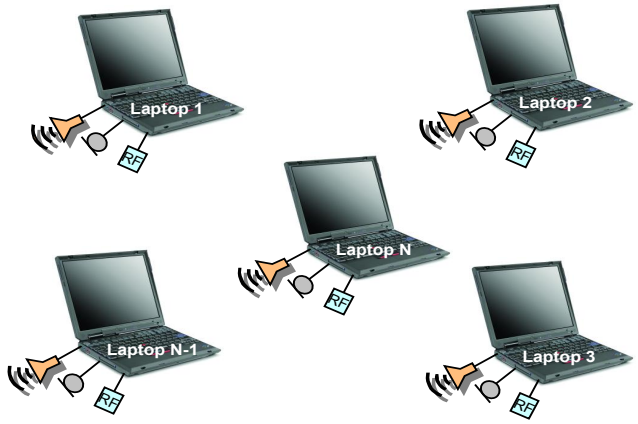


Figure 1: Distributed computing platform consisting of N general-purpose computers along with their onboard audio sensors, actuators and wireless communication capabilities.

1. INTRODUCTION

Many novel emerging multimedia applications use multiple sensors and actuators. 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. However, much of the current work has focused on setting up all the sensors and actuators on a single dedicated computing platform. Such a setup would require a lot of dedicated infrastructure in terms of the sensors, multi-channel interface cards and computing power. For example, to setup a microphone array on a single general purpose computer we need expensive multichannel sound cards and a CPU with huge computation power to process all the multiple streams.

Computing devices such as laptops, PDAs, tablets, cellular phones, and camcorders have become pervasive. We collectively refer to such devices as General Purpose Computers(GPCs). These devices are equipped with audio-visual sensors (such as microphones and cameras) and actuators (such as loudspeakers and displays). In [10], we proposed a setup to use these audio/video sensors on different devices to form a distributed network of sensors. Such an ad-hoc sensor network can be used to capture different audio-visual scenes in a distributed fashion and then use all the multiple

audio-visual streams for novel emerging applications. The advantage of such a system is that given a set of GPCs along with their sensors and actuators, it can be converted to a distributed network of sensors in an ad-hoc fashion by just adding a software wrapper on each of the GPCs.

A prerequisite for using distributed audio-visual I/O capabilities is to put the sensors and actuators into a common time and space (coordinate system). In [10] we consider the problem of providing a common time reference among multiple platforms. In this paper we focus on providing a common space by means of actively estimating the 3D positions of the sensors and actuators.

Most of the multi-microphone array processing algorithms require that the position of the microphones are known. For example in order to localize a moving speaker using a microphone array the formulation assumes that the positions of the microphones are known. If we want to beamform (spatial filtering) to a particular location then we need to know the actual microphone locations. Current systems either place the microphones in known locations or manually calibrate them. We develop a system in which the positions of the sensors on different devices are automatically calibrated using the actuators present. The solution explicitly accounts for the errors due to lack of temporal synchronization among the various sensors and actuators on different platforms. Our goal is to get the positions of the microphones and speakers on different laptops. However if the microphones and speakers are on the GPC itself we can also get the GPC location, which can be useful for location-aware computing applications.

Figure 1 shows a schematic representation of our distributed computing platform consisting of N GPCs. One of them is configured to be the master and controls and performs the location estimation. Each GPC is equipped with audio sensors (microphones), actuators (loudspeakers), and wireless communication capabilities.

1.1 Related work

The problem of self-localization of a network of nodes involves two steps: ranging and multilateration. Ranging involves the estimation of the distance between two nodes in the network. Multilateration refers to using the estimated ranges to find the position of different nodes. The ranging technology can be either based on the Time-Of-Arrival (TOA) or the Received Signal Strength (RSS) of acoustic, ultrasound or radio frequency (RF) signals. The choice of a particular technology depends on the environment and the range for which the sensor network is designed. The GPS system and long range wireless sensor networks use RF technology for range estimation. Localization using Global Positioning System (GPS) is not suitable for our applications since GPS systems do not work indoors and are very expensive. Also RSS based on RF is very unpredictable [15] and the RF TOA is very small to be used indoors. [15] discuss systems based on ultrasound TOA using specialized hardware (like motes) as the nodes. However, our goal is to use the already available sensors and actuators on the GPCs to estimate their positions. So our ranging technology is based on acoustic TOA as in [14, 11, 7].

Once we have the range estimates the Maximum Likelihood (ML) estimate can be used to get the positions. To formulate the solution we can either assume that we know the locations of a few sources (beacons) [14, 15] or design an completely ad-hoc system, where even the source locations are unknown [13, 18]. [9] discusses a system for laptop localization based on wireless ethernet. However our aim is to localize the microphones and speakers and not the GPCs. Our algorithm works assuming that all the microphones and speakers are in the same room. Partitions and walls obstruct the path of the sound. In such cases RF technology might be useful.

1.2 Contributions

- We propose a novel setup for multi-microphone array processing algorithms, using a network of multiple sensors and actuators which can be created using ad-hoc connected general purpose devices without expensive hardware or computing power.
- We automatically calibrate the positions of the sensors using actuators in unknown source locations. To the best of our knowledge, most of the previous work on position calibration (except [7] 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 account for errors due to lack of temporal synchronization. The main contribution of this paper is to formulate and solve the problem of self-localization for a distributed computing platform. We do an extensive analysis, on the errors due to lack of synchronization and propose novel formulations to account for them.
- We also derive the Cramèr-Rao bound and analyze the localization accuracy with respect to the number of sensors and sensor geometry.

1.3 Paper Organization

The rest of the paper is organized as follows. In Section 2, we formulate the problem for a conventional synchronized platform. Section 3 discusses the problems arising on a distributed computing platform and explicitly accounts for the limited precision in temporal synchronization. Section 4 discusses the different issues involved in the non-linear minimization. In Section 5, 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 6, extensive simulation results are reported. Section 7 gives a thorough discussion of the real-time system. Section 8, concludes with a summary of the present work, and with a discussion on possible extensions.

2. PROBLEM FORMULATION

Given a set of M acoustic sensors and S acoustic actuators in unknown locations, our goal is to estimate their three dimensional coordinates. Each of the acoustic actuators is excited using a known calibration signal such as Maximum Length (ML) sequences or chirp signals, and the Time of Arrival (TOA) is estimated for each of the acoustic sensors. The TOA 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. Measuring the TOA and knowing the speed of sound in the acoustical medium we can calculate the distances between each source and all microphones. Using all these pairwise distances and assuming that the TOAs are corrupted by additive white

Gaussian noise of known variance we can derive a Maximum Likelihood (ML) estimate for the unknown microphone and speaker locations.

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

- At any given instant we know the number of sensors and actuators in the network.
- The signals emitted from each of the speakers do not interfere with each other. 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 *emission start time* and the *capture start time* are zero or they are both equal. The emission start time is defined as the time after which the sound is actually emitted from the speaker once the play command has been issued in software. The capture start time is defined as the time at which the actual capture starts once the capture command is issued. This is an unrealistic assumption. In all practical cases (except on a perfectly synchronized platform) the emission and capture start times are never equal and worse it can vary with time depending on the sound card, the interrupts and the background tasks on the processor. In the next section we relax this assumption and show how this uncertainty can be incorporated in our formulation.

2.1 Maximum Likelihood Estimate

Assume we have M microphones and S sources. 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} source, respectively. Let ξ be the $(M+S)\times 3$ matrix where each row is formed by the spatial coordinates of each sensor, i.e. $\xi = [\mathbf{m_1}, .., \mathbf{m_M}, \mathbf{s_1}, .., \mathbf{s_S}]^T$

We excite one of the S sources at a time and measure the TOA at each of the M microphones. Let τ_{ij} be the estimated TOA and t_{ij} the actual TOA for the i^{th} microphone due to the j^{th} source. The actual TOA for the i^{th} microphone due to the j^{th} source is given by

$$t_{ij} = \frac{\parallel \mathbf{m_i} - \mathbf{s_j} \parallel}{c} \tag{1}$$

where c the speed of sound in the acoustical medium [1]. Let the measured TOA, τ_{ij} be corrupted by zero-mean additive white Gaussian noise [2] n_{ij} with known variance σ_{ij}^2, i.e

$$\tau_{ij} = t_{ij} + n_{ij} \tag{2}$$

Assuming that each of the TOAs are independently corrupted by zero-mean additive white Gaussian noise the likelihood function of τ_{ij} given ξ can be written as:

$$p\,[\,\tau_{ij}, i \in [1,M], j \in [1,S]\,;\,\xi\,] = \prod_{j=1}^{S}\prod_{i=1}^{M} \frac{1}{\sqrt{2\pi\sigma_{ij}^2}}\, exp\,[\frac{-(\tau_{ij}-t_{ij})^2}{2\sigma_{ij}^2}] \tag{3}$$

The log-likelihood function is:

$$ln(\,p\,[\,\tau_{ij}, i \in [1,M], j \in [1,S]\,;\,\xi\,]\,) = -\sum_{j=1}^{S}\sum_{i=1}^{M}[ln(\sqrt{2\pi\sigma_{ij}^2}) + \frac{(\tau_{ij}-t_{ij})^2}{2\sigma_{ij}^2}] \tag{4}$$

The Maximum Likelihood (ML) estimate $\hat{\xi_{ML}}$ is the one which maximizes the log likelihood function, or equivalently one which minimizes:

$$F_{ML}(\xi) = \sum_{j=1}^{S}\sum_{i=1}^{M} \frac{(\tau_{ij}-t_{ij})^2}{\sigma_{ij}^2} \tag{5}$$

$$\hat{\xi_{ML}} = \arg_{\xi} \min[F_{ML}(\xi)] \tag{6}$$

Since t_{ij} depends only on pairwise distance, any translation and rotation of the global minimum found, 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 in 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 (3D) or Y-axis (2D) we specify one more node to lie in the positive Z-axis (in 3D) 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 $y_2 > 0$.

3. SYSTEMATIC ERRORS

In the previous section we developed the ML estimate in a perfectly synchronized distributed sensor network and assumed that the measured TOA is corrupted by zero mean additive white Gaussian noise due to two reasons: (1) ambient noise and (2) room reverberation. These kind of errors

[1]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 degree Celsius. For improved position calibration it is beneficial to integrate a temperature sensor into the system. It is also possible to include the speed of sound as a parameter to be estimated, as in [14].

[2]We estimate the TOA using Generalized Crosscorrelation(GCC) [8]. The estimated TOA 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 and known variance [8]. In general the variance depends on the signal spectra.

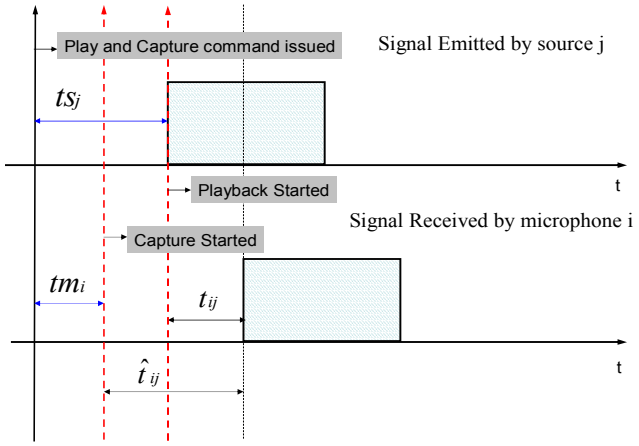


Figure 2: Schematic indicating the errors due to unknown emission start time (ts_j) and capture start time (tm_i).

in measurement are called *statistical* errors. However there can be certain errors, which are not statistical in nature. They are called *systematic* errors. One example is the error caused by the minimization routine. Other causes of errors, particularly in distributed platforms, are due the lack of synchronization among different microphones and speakers on different platforms.

As discussed in the previous section, t_{ij} the actual TOA for the i^{th} microphone due to the j^{th} source is given by Equation 1. In the previous formulation we assumed that our estimated TOA, τ_{ij} was corrupted by additive white Gaussian noise and so our model was

$$\tau_{ij} = t_{ij} + n_{ij} \tag{7}$$

where n_{ij} is normally distributed with mean zero and variance σ_{ij}^2.

Let us define the estimated TOA that is corrupted by both the statistical and systematic errors as $\hat{\tau_{ij}}$ where

$$\hat{\tau_{ij}} = \hat{t_{ij}} + n_{ij} \tag{8}$$

$\hat{t_{ij}}$ is the version of t_{ij} corrupted by systematic errors.

Let ts_j be the emission start time for the j^{th} source. (See Figure 2). The *emission start time* is defined as the time after which the sound is actually emitted from the speaker once the command has been issued. This includes the network delay (if the loudspeaker is on a different GPC), the delay in setting up the audio buffers and also the time required for the loudspeaker diaphragm to start vibrating. The emission start time is generally unknown and depends on the particular sound card and the system state such as the processor workload, interrupts, and the processes scheduled at the given instant. Let tm_i be the capture start time for the i^{th} microphone, i.e. the time instant at which sampling is started once the command is issued. It can be greater than or less than ts_j [3]. Let $\Delta t_{ij} = ts_j - tm_i$. $\hat{t_{ij}}$ is related to t_{ij} as

$$\hat{t_{ij}} = t_{ij} + \Delta t_{ij} = t_{ij} + ts_j - tm_i \tag{9}$$

[3]In a typical setup we first start the audio capture on all the devices and playback the calibration signal on each of them one by one. Hence for most cases the capture start time is less than the emission start time.

Thus a systematic error in the order of $ts_j - tm_i$ is introduced to the TOA estimate $\hat{\tau_{ij}}$. We assumed that the play and capture command were issued together. However if they were issued at different instants than the issue time can also be included in the emission and capture start times. For this we need a time reference. Assuming capture is started before playback we can assume that $tm_1 = 0$ i.e the time at which the first microphone started capturing is our origin.

We propose two methods to tackle the problem of emission and capture start times:

- 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 [10]. This reference signal can be captured in one of the input channels ,demodulated and used to estimate Δt_{ij}, 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.
- The other solution is to jointly estimate the unknown source emission and capture start time together with the microphone and source coordinates. We can incorporate both ts_j and tm_i as additional parameters to be estimated. Let us redefine ξ to include all unknown parameters for notational convenience. We can arrange all the parameters to be estimated as a vector ξ:

$$\xi = \{\mathbf{m_1}, .., \mathbf{m_M}; \mathbf{s_1}, .., \mathbf{s_S}; \mathbf{tm_1}, ..., \mathbf{tm_M}; \mathbf{ts_1}, ..., \mathbf{ts_S}\} \tag{10}$$

The ML estimate is same as in the previous case

$$\hat{\xi_{ML}} = \arg_\xi \min[\sum_{j=1}^{S}\sum_{i=1}^{M} \frac{(\hat{\tau_{ij}} - \hat{t_{ij}})^2}{\sigma_{ij}^2}] \tag{11}$$

Similar to fixing a reference coordinate system in space we introduce a reference time line by setting $tm_1 = 0$.

4. NON-LINEAR LEAST SQUARES

The ML estimate for the node coordinates of the microphones and loudspeakers is implicitly defined as the minimum of the non-linear function given in Equation 11. 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 more efficient. The Levenberg-Marquardt method [5] is a popular method for non-linear least squares problems. It is a compromise between steepest descent and Newton's methods. For more details on nonlinear minimization refer to, for example [5].

The following are the non zero partial derivatives [4] needed for the minimization routines: [5]

[4]These derivatives form the non-zero elements of the Jacobian matrix. For least squares problems the gradient and the Hessian can be got from the Jacobian

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

$$\frac{\partial \hat{t}_{ij}}{\partial mx_i} = -\frac{\partial \hat{t}_{ij}}{\partial sx_j} = \frac{\partial t_{ij}}{\partial mx_i} = -\frac{\partial t_{ij}}{\partial sx_j} = \frac{mx_i - sx_j}{c\|m_i - s_j\|}$$
$$\frac{\partial \hat{t}_{ij}}{\partial my_i} = -\frac{\partial \hat{t}_{ij}}{\partial sy_j} = \frac{\partial t_{ij}}{\partial my_i} = -\frac{\partial t_{ij}}{\partial sy_j} = \frac{my_i - sy_j}{c\|m_i - s_j\|}$$
$$\frac{\partial \hat{t}_{ij}}{\partial mz_i} = -\frac{\partial \hat{t}_{ij}}{\partial sz_j} = \frac{\partial t_{ij}}{\partial mz_i} = -\frac{\partial t_{ij}}{\partial sz_j} = \frac{mz_i - sz_j}{c\|m_i - s_j\|}$$
$$\frac{\partial \hat{t}_{ij}}{\partial ts_j} = -\frac{\partial \hat{t}_{ij}}{\partial tm_i} = \frac{\partial t_{ij}}{\partial ts_j} = -\frac{\partial t_{ij}}{\partial tm_i} = 1 \quad (12)$$

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. The following are a few points, which can be exploited for a better initial guess:

- If we have an approximate idea of the microphone and speaker positions, then we can initialize manually.
- Use the previous geometry as the initial guess, if the sensor geometry changes and recalibration is needed. This procedure implicitly assumes that geometry does not change drastically.
- Minimize the function from different initial guesses and choose the one with the minimum value.
- Assuming that the microphones and speakers on a given computing platform are approximately at the same position and given all the pairwise distances between the computing platforms, we can use classical Multidimensional Scaling approach [16] to determine the coordinates from the Euclidean distance matrix. This involves converting the symmetric distance matrix to a matrix of scalar products with respect to some origin and then perform a singular value decomposition to obtain the matrix of coordinates[6]. This matrix of coordinates can be used as our initial guess.
- Use results from video if available. It may be difficult to find the location of the actual microphone on the laptop, but an approximate location of the laptop can be found easily using multiple cameras. This rough estimate can be used as an initial guess.

5. CRAMÉR-RAO BOUND

The Cramér-Rao bound gives a lower bound on the variance of *any* unbiased estimate [17]. In this section, we first derive the Cramér-Rao bound (CRB) for the estimate of the node coordinates, i.e., the matrix ξ, and then discuss the influence of the number of sensors and actuators and the sensor geometry on the lower bound. We have not included the unknown emission and capture start times in our derivation, however, it can be easily extended by adding them as extra nuisance parameters.

Let Φ, be a vector of length $3(M+S)\times 1$, representing all the unknown non-random parameters to be estimated.

$$\Phi = [\Phi_m \Phi_s]^T$$
$$\Phi_m = [mx_1, my_1, mz_1, ..., mx_M, my_M, mz_M]$$
$$\Phi_s = [sx_1, sy_1, sz_1, ..., sx_S, sy_S, sz_S] \quad (13)$$

[6]Let B_k be the scalar product matrix with respect to the k^{th} laptop as the origin and X be the matrix of the cartesian coordinates of the laptops. Then $B_k = XX^T$. So B_k is positive semidefinite and hence we can calculate $X = U\Sigma^{1/2}$ where the singular decomposition of B_k is given by $B_k = U\Sigma U^T$

where mx_i, my_i, and mz_i are the x, y and z coordinates of the i^{th} microphone and sx_i, sy_i, and sz_i are the x, y and z coordinates of the i^{th} speaker. Let Γ, be a vector of length $MS\times 1$, representing our noisy measurements of the TOAs:

$$\Gamma = [\tau_{11}, \tau_{12}, \ldots, \tau_{1S}, \ldots\ldots, \tau_{M1}, \tau_{M2}, \ldots, \tau_{MS}]^T \quad (14)$$

Let $T(\Phi)$, be a vector of length $MS\times 1$, representing the actual TOAs.

$$T(\Phi) = [t_{11}, t_{12}, \ldots, t_{1S}, \ldots\ldots, t_{M1}, t_{M2}, \ldots, t_{MS}]^T \quad (15)$$

Then according to our Gaussian noise model,

$$\Gamma = T(\Phi) + N \quad (16)$$

where N is the zero-mean additive white Gaussian noise vector of length $MS\times 1$ where each element has the variance σ_{ij}^2. Also let us define Σ to be the $MS \times MS$ diagonal covariance matrix.

$$\Sigma = diag[\sigma_{11}^2 \ldots \sigma_{1S}^2 \ldots \sigma_{M1}^2 \ldots \sigma_{MS}^2] \quad (17)$$

The variance of any unbiased estimator $\hat{\Phi}$ of Φ is bounded as [17]

$$E\left[(\hat{\Phi} - \Phi)(\hat{\Phi} - \Phi)^T\right] \geq J^{-1}(\Phi) \quad (18)$$

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

$$J \triangleq E\left\{\left[\frac{\partial}{\partial \Phi}\ln p(\Gamma/\Phi)\right]\left[\frac{\partial}{\partial \Phi}\ln p(\Gamma/\Phi)\right]^T\right\} \quad (19)$$

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

In order to derive the Cramér-Rao bound, we write the likelihood function in vector form as:

$$p(\Gamma/\Phi) = (2\pi)^{-\frac{MS}{2}} \mid \Sigma \mid^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\Gamma - T)^T \Sigma^{-1}(\Gamma - T)\right] \quad (20)$$

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

$$\frac{\partial}{\partial \Phi}\ln p(\Gamma/\Phi) = \left[\frac{\partial}{\partial \Phi}T(\Phi)\right]^T \Sigma^{-1}(\Gamma - T) \quad (21)$$

Substituting this in Equation 19 and taking the expectation the Fishers Information matrix is,

$$J = \left[\frac{\partial}{\partial \Phi}T(\Phi)\right]^T \Sigma^{-1}\left[\frac{\partial}{\partial \Phi}T(\Phi)\right] \quad (22)$$

Let us define,

$$\Upsilon = \left[\frac{\partial T(\Phi)}{\partial \Phi}\right] \quad (23)$$

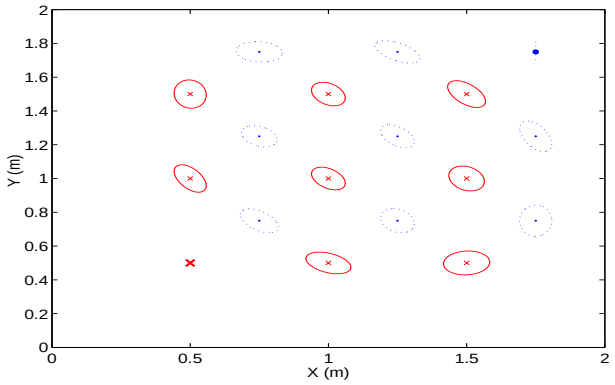
where Υ is an $MS \times 3(M+S)$ matrix of partial derivatives (see Equation 12). Then

$$J = \Upsilon^T \Sigma^{-1} \Upsilon \quad (24)$$

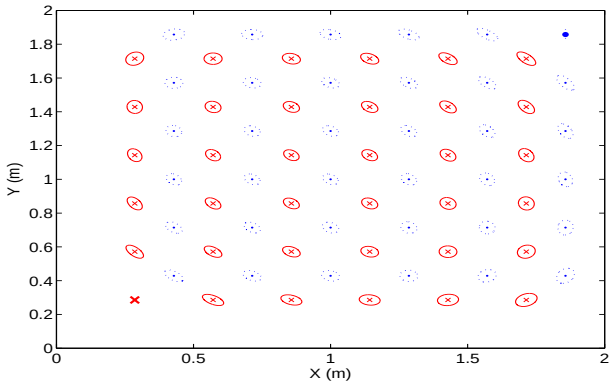
If we assume $\Sigma = \sigma^2 I$ i.e. all the noise components are independent and have the same variance σ^2 then,

$$J = \frac{1}{\sigma^2}\Upsilon^T \Upsilon \quad (25)$$

If we assume that all the microphone and source locations are unknown, the matrix J is rank deficient and hence



(a)



(b)

Figure 3: 95% uncertainty ellipses for a regular 2 dimensional array of (a) 9 speakers and 9 microphones, (b) 36 speakers and 36 microphones. Noise variance in both cases is $\sigma^2 = 10^{-8}$. 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).

not invertible. This is because the solution to the ML estimation problem as formulated is 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^{-1} represent the lower bound for the error variance for estimating each of the parameters in Φ.

5.1 Effect of the number of nodes

As the number of nodes increases in the network, the CRB decreases i.e. more the number of microphones and speakers in the network, the lesser the error in estimating their positions. Figure 3(a) shows the 95% uncertainty ellipses for a regular two dimensional array consisting of 9 microphones (shown as crosses (x)) and 9 speakers (shown as dots (.)). We fixed the position of one microphone and the x coordinate of one speaker. The fixed microphone and speaker are shown in bold. For the fixed speaker only the variance in y direction is shown since the x coordinate is fixed. The noise variance was assumed to be 10^{-8}. For a given noise variance, we can say with 95% probability the estimated position will

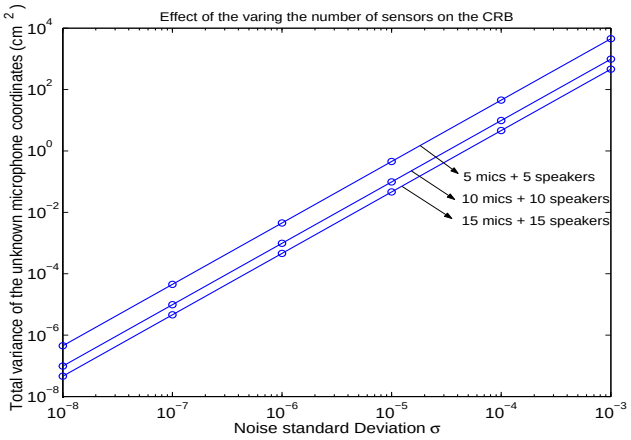


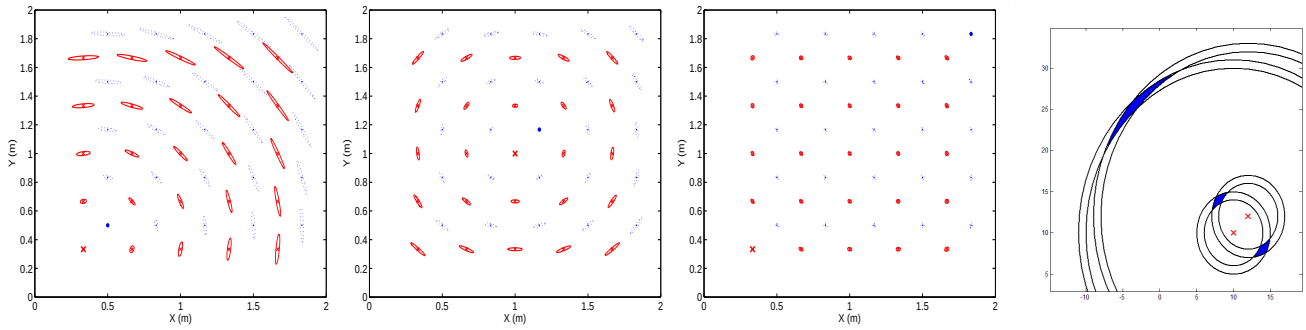
Figure 4: Cramér-Rao bound on the total variance of all the unknown microphone coordinates as a function of noise standard deviation σ for different number of microphones and speakers given the positions of 1 microphone and 2 speakers.

lie in the ellipse. Figure 3(b) shows the corresponding 95% uncertainty ellipses for a two dimensional array consisting of 36 microphones and 36 speakers. It can be seen that as the number of sensors in the network increases the dimensions of the uncertainty ellipses decreases. This can also be seen from Figure 4, where the lower bound on the total variance of the unknown microphone coordinates is plotted for different number of nodes in a three dimensional network.

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 TOA 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 $O(n^2)$. So the linear increase in the number of unknown parameters, is compensated by the quadratic increase in the available measurements.

5.2 Effect of sensor geometry

The geometry of the network plays an important role in CRB. It is possible to analyze how to place the sensors in order to achieve a lower CRB. In an ad-hoc network, however, such analysis is of little benefit. In our formulation we assumed that we know the positions of a certain number of nodes, i.e we fix three of the nodes to lie in the x-y plane. The CRB depends on which of the sensor nodes are assumed to have known positions. Figure 5 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 5(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 5(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 speak-



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

Figure 5: 95% uncertainty ellipses for a regular 2 dimensional array of 25 microphones shown in solid lines and 25 speakers shown in dotted lines 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

ers as shown in Figure 5(d). Each annulus 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 faraway from each other as possible. In Figure 5(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.

6. MONTE CARLO SIMULATIONS

We performed a series of Monte Carlo simulations to compare the experimental performance with the Cramèr Rao bound (CRB). Ten microphones and ten speakers were randomly selected in a room of dimensions $2.0m \times 2.0m \times 2.0m$. Based on the geometry of the setup the actual TOA 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 TOA matrix was given as an input to the Levenberg-Marquadrat minimization routine. The positions of two microphones and two speakers were assumed to be known. The starting point for the minimization procedure was chosen to lie within a sphere of 50cm for each of the nodes in the network. For each noise variance σ^2, the results were averaged over 2000 trials corresponding to different initial guesses.

Figure 6(a), shows the total variance of all the unknown microphone coordinates plotted against the noise standard deviation σ. The corresponding CRB is also shown. It can be seen that the experimental results match closely with the theoretical bound. Figure 6(d) shows the average bias. The estimator shows a slight bias. The bias could be due to the particular optimization method used or due to the finite number of trials.

6.1 Random vs. intelligent initial guess

The common problem with minimization methods is that they may get stuck in a local minimum. To avoid this we need a very good initial guess of the locations. We regard starting points within 50cm of the node's true location as intelligent initial guesses. Figure 6(b) shows the total variance of the unknown microphone coordinates plotted against the noise standard deviation σ for intelligent and random initial guess. Figure 6(e) shows the corresponding average bias. It can be seen that even though the bias is not that high, the variance is substantially higher for the random case. Hence the choice of initial configuration is very crucial.

6.2 Estimation of source emission time

We also performed a series of simulations where the TOA was assumed to be corrupted by the unknown source emission times. Figure 6(c) and 6(f) show the total variance and average bias of the unknown microphone coordinates plotted against the source emission time with and without accounting the source emission time in the ML estimation procedure. It can be seen that with the increase of unaccounted source emission times, the bias and variance increase. However, if the source emission times are estimated, too, then the bias becomes nearly zero and the variance is also much lower.

7. SYSTEM DESIGN ISSUES

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

7.1 Calibration signals

In order to measure the TOA accurately the calibration signal has to be appropriately selected and the parameters properly tuned. Chirp signals and Maximum Likelihood (ML) sequences are the two most popular sequences used. 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

$$s(t) = Acos(2\pi(f_0 + (\frac{f_1 - f_0}{T})t)) \qquad 0 \leq t \leq T \qquad (26)$$

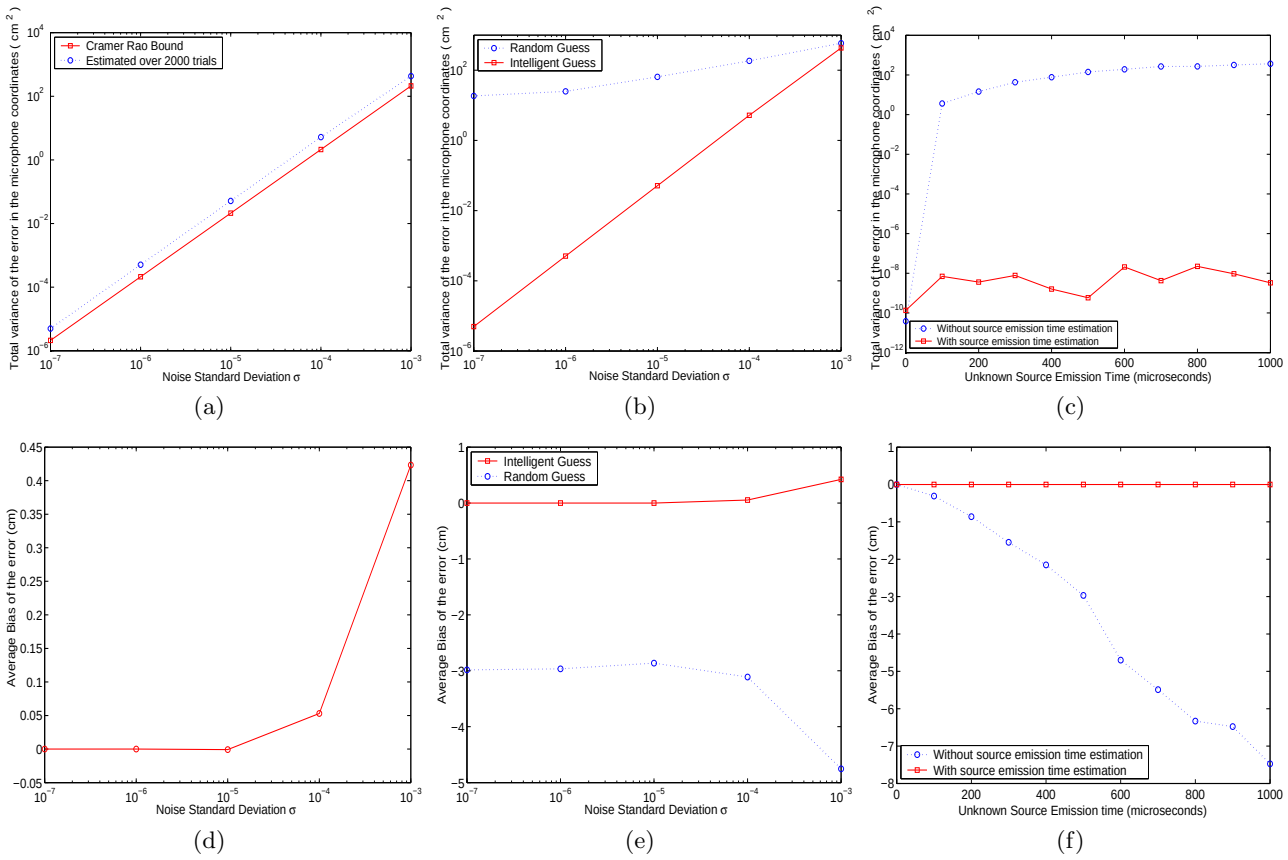


Figure 6: Monte Carlo Simulation results: (a) (b) and (c) Total variance of the error in all the unknown microphone coordinates and (d)(e) and (f) average bias of the error for a network consisting of 10 microphones and 10 speakers. The positions of 2 microphones and 2 speakers are assumed to be known.

In our system, we used the chirp signal of 512 samples at 44.1kHz (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. Figure 7(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 7(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. Figure 7(c) and Figure 7(d) show the magnitude of the frequency response of the transmitted chirp signal and the received chirp signal, respectively.

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

7.2 TOA 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 TOA 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 [8] developed a Maximum Likelihood (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 [8]. 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 [8]

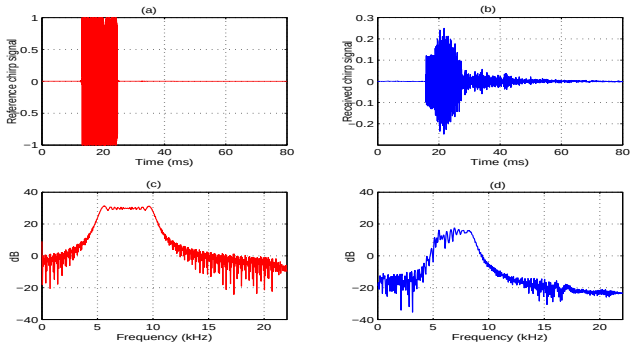


Figure 7: (a) The loopback reference chirp signal (b) the chirp signal received by one of the microphones (c) the magnitude of the frequency response of the reference signal and (d) the received chirp signal

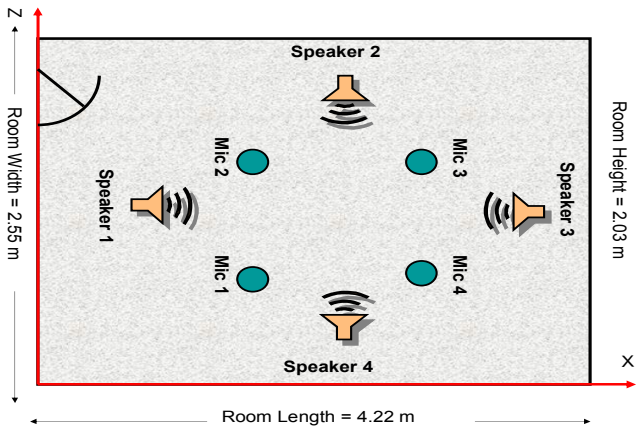


Figure 8: Top view of the whisper room containing 4 microphones and 4 speakers

$R_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} W(\omega)X_1(\omega)X_2^*(\omega)e^{j\omega\tau}d\omega$, 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 [8]. 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. For a practical room we can estimate the room noise, and use the combined ML and PHAT weighting by appropriately emphasizing each weighting function based on the noise levels [6]. A more accurate estimate of the peak can be found by upsampling the GCC function.

7.3 Testbed Setup

The real-time setup has been tested in a synchronized as well as a distributed setup using laptops. Figure 8 shows the top view of our experimental setup. Four omnidirectional microphones (RadioShack) and four loudspeakers (Mackie HR624) were setup in a room with low reverberation and low ambient noise. The ground truth was measured manually to validate the results from the position calibration methods. In a synchronized setup, the microphones and loudspeakers were interfaced using an RME DIGI9652 card. For a distributed implementation the loudspeakers and the microphones were connected to four laptops (3 IBM T-series Thinkpads and one Dell laptop). All the laptops had Intel Pentium series processors.

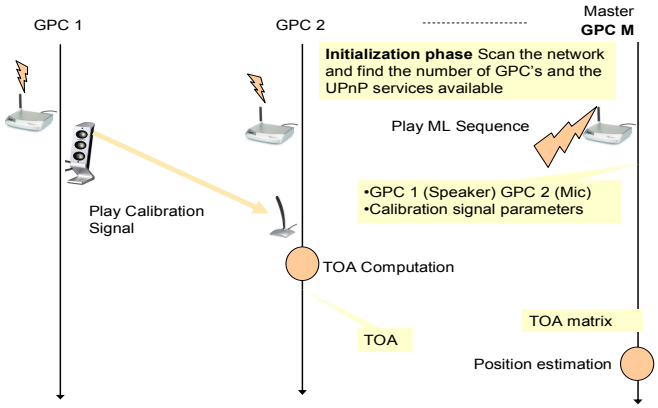


Figure 9: Schematic showing the distributed control scheme.

7.4 Software details

Capture and play back was done using the free, cross platform, open-source, audio I/O library Portaudio [3]. Most of the signal processing tasks were implemented using the Signal Processing Library in Intel® Integrated Performance Primitives (IPP). IPP is a cross-platform low-level software layer that abstracts multimedia functionality from the processor underneath providing highly optimized code [2]. For the non-linear minimization we used the *mrqmin* routine from Numerical Recipes in C [12]. For displaying the calibrated microphones and speakers we used the OpenGL Utility Toolkit (GLUT) ported to Win32 [4].

For the distributed platform we used the UPnP [1] technology to form an adhoc network and control the audio devices on different platforms. UPnP technology is a distributed, open networking architecture that employs TCP/IP and other Internet technologies to enable seamless proximity networking [1]. The real time setup integrates the distributed synchronization scheme using ML sequence as proposed in [10]. Figure 9 shows a schematic of the TOA computation protocol. Each of the laptops has an UPnP service running for playing the chirp signal and capturing the audio stream. One of the GPC's is configured to be the master which plays the ML sequence as described in [10]. A

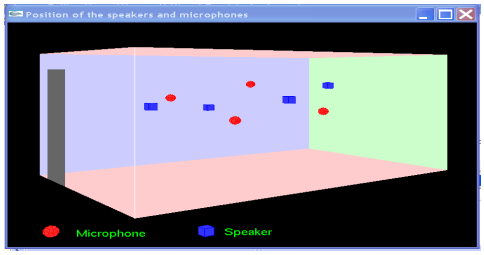


Figure 10: A sample screen shot of the OpenGL display.

program on the master scans the network for all the available UPnP players. Then the chirp signal is played on each of the devices one after the other and the signal is captured. The TOA computation is distributed among all the laptops, in that each laptop computes its own TOA and reports it back to the master. The master performs the minimization routine once it has the TOA matrix.

As regards to CPU utilization the TOA estimation consumes negligible resources. If we use a good initial guess via the Multidimensional Scaling technique then the minimization routine converges within 8 to 10 iterations.

7.5 Results

For the setup consisting of 4 speakers and 4 microphones, the sensors' and actuators' three dimensional locations could be estimated with an average bias of 0.08 cm and average standard deviation of 3.8 cm (results averaged over 100 trials). In order to display the microphones and speakers in context of the room, the positions of two speakers and one microphone was assumed to be known. Figure 10 shows a snapshot of the OpenGL display, showing the estimated locations of the speakers and microphones.

8. SUMMARY AND FURTHER STUDIES

In this paper we described the problem of localization 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 and enables a rich set of new multistream A/V applications on platforms that 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. There are a number of ways to improve localization in the future. The one we are currently pursuing is targeted at using Time Difference Of Arrival instead of Time Of Arrival and getting closed form approximations to be used as initial guess for the minimization routine.

9. ACKNOWLEDGMENTS

The authors would like to acknowledge the help of Dr. Bob Liang, Dr.Amit Roy Chowdhury and Dr.Ramani Duraiswami who contributed valuable comments and suggestions for this work. We would also like to thank the three anonymous reviewers and our shepherd Dr.Dongyan Xu for the reviews and comments which helped to improve the overall quality of the paper.

10. REFERENCES

[1] http://intel.com/technology/upnp/.
[2] http://www.intel.com/software/products/perflib/.
[3] http://www.portaudio.com/.
[4] http://www.xmission.com/nate/glut.html.
[5] D. P. Betrsekas. *Nonlinear Programming*. Athena Scientific, 1995.
[6] M. Brandstein, J. Adcock, and H. Silverman. A practical time-delay estimator for localizing speech sources with a microphone array. *Comput. Speech Lang.*, 9:153–169, September 1995.
[7] 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.
[8] C. H. Knapp and G. C. Carter. The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-24(4):320–327, August 1976.
[9] 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 *Proceedings of The Eighth ACM International Conference on Mobile Computing and Networking (MOBICOM)*, Atlanta, GA, USA, Sept. 2002.
[10] 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.
[11] R. Moses, D. Krishnamurthy, and R. Patterson. A self-localization method for wireless sensor networks. *Eurasip Journal on Applied Signal Processing Special Issue on Sensor Networks*, 2003(4):348–358, March 2003.
[12] H. P. Press, S. A. Teukolsky, W. T. Vettring, and B. P. Flannery. *Numerical Recipes in C The Art of Scientific Computing*. Cambridge University Press, 2 edition, 1995.
[13] Y. Rockah and P. M. Schultheiss. Array shape calibration using sources in unknown locations Part II: Near-field sources and estimator implementation. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-35(6):724–735, June 1987.
[14] J. M. Sachar, H. F. Silverman, and W. R. Patterson III. Position calibration of large-aperture microphone arrays. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages II–1797 – II–1800, 2002.
[15] 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.
[16] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17:401–419, 1952.
[17] H. L. Van Trees. *Detection, Estimation, and Modulation Theory*, volume Part 1. Wiley-Interscience, 2001.
[18] A. J. Weiss and B. Friedlander. Array shape calibration using sources in unknown locations-a maxilmum-likelihood approach. *IEEE Trans. Acoust., Speech, Signal Processing*, 37(12):1958–1966, December 1989.