Self Localization of acoustic sensors and actuators on Distributed platforms

Vikas C. Raykar* Igor Kozintsev Rainer Lienhart

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

Abstract

In this paper we present a novel algorithm to automatically determine the relative 3D positions of 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 using the technique of metric multidimensional scaling, which is further refined by minimizing a non-linear error function. Our formulation and solution accounts for the errors in localization, due to lack of temporal synchronization among different platforms. The theoretical performance limit for the sensor positions is derived via the Cramér-Rao bound and analyzed with respect to the number of sensors and actuators as well as their geometry. Extensive simulation results are reported together with a discussion of the practical issues in a real-time system.

1. Introduction and Motivation

Arrays of audio/video sensors and actuators (such as microphones, cameras, loudspeakers and displays) along with array processing algorithms offer a rich set of new features for emerging E-Learning and collaboration applications. Until now, array processing was mostly out of reach for consumer applications perhaps due to significant cost of dedicated hardware and complexity of processing algorithms. At the same time, recent advances in mobile computing and communication technologies suggest a very attractive platform for implementing these algorithms. Students in classrooms, co-workers at meetings are nowadays accompanied by one or several mobile computing and communication devices like laptops, PDAs, tablets, which have multiple audio and video I/O devices onboard. Such an ad-hoc sensor/actuator network can be used to capture/render different audio-visual scenes in a distributed fashion leading to novel emerging applications. A few such applications include multi-stream audio/video rendering, image based 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 network of sensors in an ad-hoc fashion by just adding appropriate software layers. No dedicated infrastructure in terms of the sensors, actuators, multi-channel interface cards and computing power is required. However, there are several important technical and theoretical problems to be addressed before the idea of using those devices for array DSP algorithms can materialize in real-life applications.

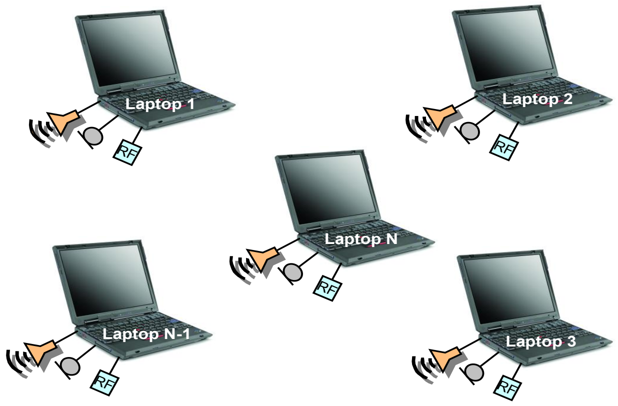


Figure 1: Distributed computing platform consisting of N general-purpose computers along with their onboard audio sensors, actuators and wireless communication capabilities.

A prerequisite for using distributed audio-visual I/O capabilities is to put sensors and actuators into a common time and space (coordinate system). In [1] we proposed a way to provide a common time reference for multiple distributed GPCs. In this paper we focus on providing a common space (coordinate system) by means of actively estimating the three dimensional positions of the sensors and actuators. Many multi-microphone array processing algorithms (like sound source localization or conventional beamforming) need to know the positions of the microphones very precisely. Current systems either place the microphones in known locations or manually calibrate them. There are

[*] 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.

some approaches which do calibration using speakers in known locations [2]. This paper offers a more general approach where no assumptions about the positions of the speakers are made. Our solution explicitly accounts for the errors in localization due to lack of temporal synchronization among different platforms. Figure 1 shows a schematic representation of our *distributed computing platform* consisting of N GPCs. One of them is configured to be the master. The master controls the distributed computing platform and performs the location estimation. Each GPC is equipped with audio sensors (microphones), actuators (loudspeakers), and wireless communication capabilities.

The problem of self-localization for a network of nodes generally involves two steps: ranging and multilateration. The ranging technology can be either based on the Time Of Flight (TOF) or the Received Signal Strength (RSS) of acoustic, ultrasound or radio frequency (RF) signals. 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 [3] and the RF TOF is quite small to be used indoors. [3] discusses systems based on ultrasound TOF using specialized hardware (like motes) as the nodes. However, our goal is to use the already available sensors and actuators on GPCs to estimate their positions. Our ranging technology is based on acoustic TOF as in [2, 4, 5]. Once we have the range estimates the Maximum Likelihood (ML) estimate can be used to get the positions. To find the solution one can assume that the locations of a few sources are known as in [2, 3] or make no such assumptions as in [4, 6]. The following are the novel contributions of this paper.

- We propose a novel setup for array processing algorithms using a network of multiple sensors and actuators, which can be created using ad-hoc connected general purpose devices such as laptops, PDAs, and tablets.
- The position estimation problem has been derived as a maximum likelihood in several papers [4, 6, 2]. The solution turns out to be the minimum of a nonlinear cost function. Iterative nonlinear least square optimization procedures require a very close initial guess to converge to a global maximum. We propose the technique of metric Multidimensional Scaling [7] in order to get an initial guess for the nonlinear minimization problem. Using this technique, we get the approximate positions of GPCs.
- Most of the previous work on position calibration (except [5] which describes a setup based on Compaq iPAQs and motes) are formulated assuming time synchronized platforms. However in an ad-hoc distributed computing platform consisting of heterogeneous GPCs we need to explicitly account for errors due to lack of temporal synchronization. We perform an analysis of the localization errors due to lack of synchronization among multiple platforms and propose ways to account for the unknown emission start times and capture start times.
- We derive the Cramèr-Rao bound and analyze the localization accuracy with respect to the number of sensors and sensor geometry.

2. Problem Formulation

Given a set of M acoustic sensors (microphones) and S acoustic actuators (speakers) in unknown locations, our goal is to estimate their three dimensional coordinates. Each of the acoustic actuators is excited using a known calibration signal such as maximum length sequences or chirp signals, and the Time of Flight (TOF) is estimated for each of the acoustic sensors. The TOF for a given pair of microphone and speaker is defined as the time taken by the acoustic signal to travel form the speaker to the microphone.

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

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

where c the speed of sound in the acoustical medium [1] 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 multichannel sound cards to interface multiple microphones and speakers [2].

However in a typical distributed setup as shown in Figure 1, the master starts the audio capture and playback on each of the GPCs one by one. As a result the capture starts at different instants on each GPC and also the time at which the calibration signal was emitted from each loud speaker is not known. So the TOF which we measure from the signal captured includes both the speaker emission start time

[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 Celsius degrees.

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

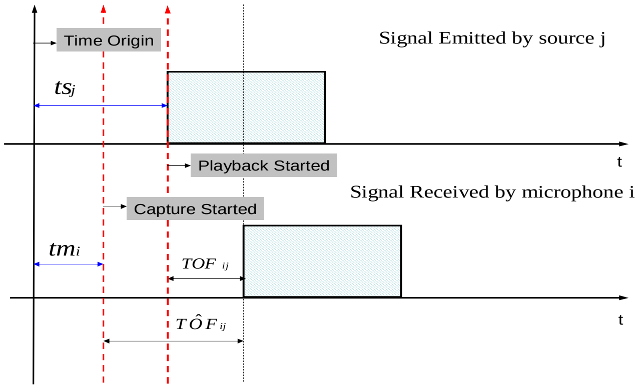


Figure 2: Schematic indicating the errors due to unknown speaker emission and microphone capture start time.

and the microphone capture start time (See Figure 2 where $T\hat{O}F_{ij}$ is what we measure and TOF_{ij} is what we require). The speaker emission start time is defined as the time at which the sound is actually emitted from the speaker. This includes the time when the play back command was issued (with reference to some time origin), the network delay involved in starting the playback on a different machine (if the speaker is on a different GPC), the delay in setting up the audio buffers and also the time required for the speaker diaphragm to start vibrating [3]. The microphone capture start time is defined as the time instant at which capture is started. This includes the time when the capture command was issued, the network delay involved in starting the capture on a different machine and the delay in transferring the captured sample from the sound card to the buffers.

Let ts_j be the emission start time for the j^{th} source and tm_i be the capture start time for the i^{th} microphone (see Figure 2). Incorporating these two the actual TOF now becomes,

$$T\hat{O}F_{ij}^{actual} = \frac{\parallel \mathbf{m_i} - \mathbf{s_j} \parallel}{c} + ts_j - tm_i \tag{2}$$

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

In this paper we propose to use the Time Difference Of Arrival (TDOA) instead of the TOF. The TDOA for a given pair of microphones and a speaker is defined as the time difference between the signal received by the two microphones [4]. Let $TDOA_{ikj}^{estimated}$ be the estimated TDOA between the i^{th} and the k^{th} microphone when the j^{th} source is excited. Let $TDOA_{ikj}^{actual}$ be the actual TDOA. It is given by

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

Including the source emission and capture start times, it becomes

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

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

2.1 Maximum Likelihood (ML) Estimate

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

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

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

$$\begin{aligned} \hat{\Theta}_{ML} &= \arg_{\Theta} \max F(\Theta, \Gamma) \\ F(\Theta, \Gamma) &= -\frac{1}{2}[\Gamma - T(\Theta)]^T \Sigma^{-1} [\Gamma - T(\Theta)] \end{aligned} \tag{6}$$

Assuming that each of the TDOAs are independently corrupted by zero-mean additive white Gaussian noise [5] of

[3] 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.

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

[5] We estimate the TDOA or TOF using Generalized Cross Correlation (GCC)[9]. The estimated TDOA 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 [9].

variance σ_{ikj}^2 the ML estimate turns out to be a nonlinear least squares problem (in this case Σ is a diagonal matrix), i.e.

$$\hat{\Theta}_{ML} = \arg_{\Theta} \min[\widetilde{F}_{ML}(\Theta, \Gamma)]$$

$$\widetilde{F}_{ML}(\Theta, \Gamma) =$$

$$\sum_{j=1}^{S} \sum_{i=1}^{M} \sum_{k=i+1}^{M} \frac{(TDOA_{ikj}^{estimated} - T\hat{D}OA_{ikj}^{actual})^2}{\sigma_{ikj}^2} \qquad (7)$$

Since the solution depends only on pairwise distances, any translation, rotation and reflection of the global minimum found will also be a global minimum. In order to make the solution invariant to rotation and translation 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 $(x_2, y_2, 0)$. 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 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 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 $y_2 > 0$. Similar to fixing a reference coordinate system in space we introduce a reference time line by setting $tm_1 = 0$.

3. Problem Solution

The ML estimate for the node coordinates of the microphones and loudspeakers is implicitly defined as the minimum of a non-linear function. The solution is same as a nonlinear weighted least squares problem. The Levenberg-Marquardt method is a popular method for solving nonlinear least squares problems. For more details on nonlinear minimization refer to [10]. Least squares optimization requires that the total number of observations is greater than or equal to the total number of parameters to be estimated. This imposes a minimum number of microphones and speakers required for the position estimation method to work. Assuming M=S=K, Table 1 lists the minimum K required for the algorithm.

Table 1: Minimum value of Microphone Speaker Pairs (K) required for different estimation procedures (D-Dimension)

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

One problem with minimization is that it can often get stuck in a local minima. In order to avoid this we need a good starting guess. We use the technique of metric multidimensional scaling (MDS) [7] to get a closed form approximation for the microphone and speaker positions, which is used as a starting point for the minimization routine. MDS is a popular method 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.

Given a set of N GPCs, let X be a $N \times 3$ matrix where each row represents the 3D coordinates of each GPC. Then the $N \times N$ matrix $B = XX^T$ is called the dot product matrix. By definition, B is a symmetric positive definite matrix, so the rank of B (i.e the number of positive eigen values) is equal to the dimension of the datapoints i.e. 3 in this case. Also based on the rank of B we can find whether the GPCs are on a plane (2D) or distributed in 3D. Starting with a matrix B (possibly corrupted by noise), it is possible to factor it to get the matrix of coordinates X. One method to factor B is to use singular value decomposition (SVD) [11], i.e., $B = U\Sigma U^T$ where Σ is a $N \times N$ diagonal matrix of singular values. The diagonal elements are arranged as $s_1 \geq s_2 \geq s_r > s_{r+1} = ... = s_N = 0$, where r is the rank of the matrix B. The columns of U are the corresponding singular vectors. We can write $X' = U\Sigma^{1/2}$. From X' we can take the first three columns to get X. If the elements of B are exact (i.e., they are not corrupted by noise), then all the other columns are zero. It can be shown that SVD factorization minimizes the matrix norm $\parallel B - XX^T \parallel$.

In practice we can estimate the distance matrix D where the ij^{th} element is the Euclidean distance between the i^{th} and the j^{th} GPC. We have to convert this distance matrix D into a dot product matrix B. In order to form the dot product matrix we need to choose some point as the origin of our coordinate system. Any point can be selected as the origin, but Togerson [7] 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 obtain the dot product matrix B using the cosine law which relates the distance between two vectors to their lengths and the cosine of the angle between them. Refer to Appendix I for a detailed derivation of how to convert the distance matrix to the scalar product matrix.

In the case of M microphones and S speakers we cannot use MDS directly because we cannot measure all the pairwise distances. We can measure the distance between each speaker and all the microphones. However, we cannot measure the distance between two microphones or two speakers. In order to apply MDS, we cluster microphones and speakers, which are close together. In practice, it is justified by the fact that the microphones and the speakers on the same GPC are close together. Assuming that all GPCs have at least one microphone and one speaker, we can measure the

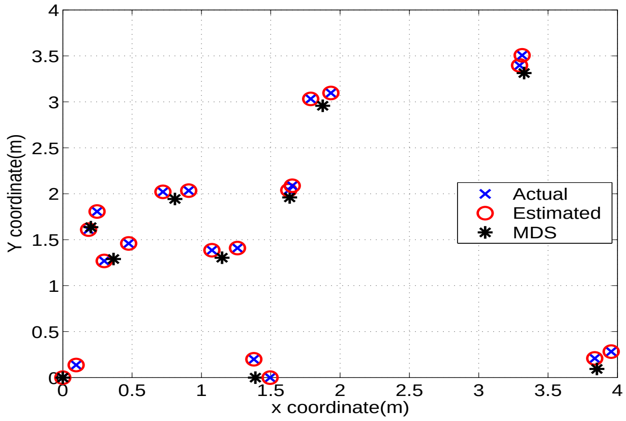


Figure 3: Results of Multidimensional Scaling for a network consisting of 10 GPCs each having one microphone and one speaker.

distance between the speakers on one GPC and the microphones on the other and vice versa. Taking the average we get an approximate distance between the two GPCs. The position estimate obtained using MDS has the centroid as the origin and an arbitrary orientation. Therefore, the solution obtained using MDS is translated, rotated and reflected to the reference coordinate system discussed earlier. Figure 3 shows an example with 10 laptops each having one microphone and one speaker. The actual locations of the sensors and actuators are shown as ’x’. The ’*’s are the approximate GPC locations resulting from MDS. As can be seen the MDS result is very close to the true microphone and speaker locations. Each GPC location got using MDS is randomly perturbed to be used as a initial guess for the microphones and speakers on that GPC. The ’o’ are the results from the ML estimation procedure using the perturbed MDS locations as the initial guess. The algorithm can be summarized as follows:

ALGORITHM

Say we have M microphones and S speakers

- **STEP 0**: *Form a Coordinate system by selecting 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.*
- **STEP 1**: *Compute the $M \times S$ Time Of Flight (TOF) matrix.*
- **STEP 2**:
  - *Convert the TOF matrix into an approximate distance matrix by appropriately clustering the closest microphones and speakers.*
  - *Get the approximate positions of the clustered entities using metric Multidimensional Scaling.*
  - *Translate, rotate and mirror the coordinates to the coordinate system specified in STEP 0.*
- **STEP 3**:
  - *Slightly perturb the coordinates from STEP 2 to get approximate initial guess for the microphone and speaker coordinates.*
  - *Set an approximate initial guess for the microphone capture start time*
  - *Minimize the TDOA based error function using the Levenberg-Marquardat method to get the final positions of the microphones and speakers.*

---

4. Cramér-Rao bound

The Cramér-Rao bound gives a lower bound on the variance of *any* unbiased estimate [12]. It does not depend on the particular estimation method used. In this section, we derive the Cramér-Rao bound (CRB) assuming our estimator is unbiased. The variance of any unbiased estimator $\hat{\Theta}$ of Θ is bounded as [12]

$$E\left[(\hat{\Theta}-\Theta)(\hat{\Theta}-\Theta)^T\right] \geq F^{-1}(\Theta) \tag{8}$$

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

$$F(\Theta) = E\left\{[\nabla_\Theta \ln p(\Gamma/\Theta)]\,[\nabla_\Theta \ln p(\Gamma/\Theta)]^T\right\} \tag{9}$$

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

$$\nabla_\Theta \ln p(\Gamma/\Theta) = J^T\Sigma^{-1}(\Gamma - T) \tag{10}$$

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

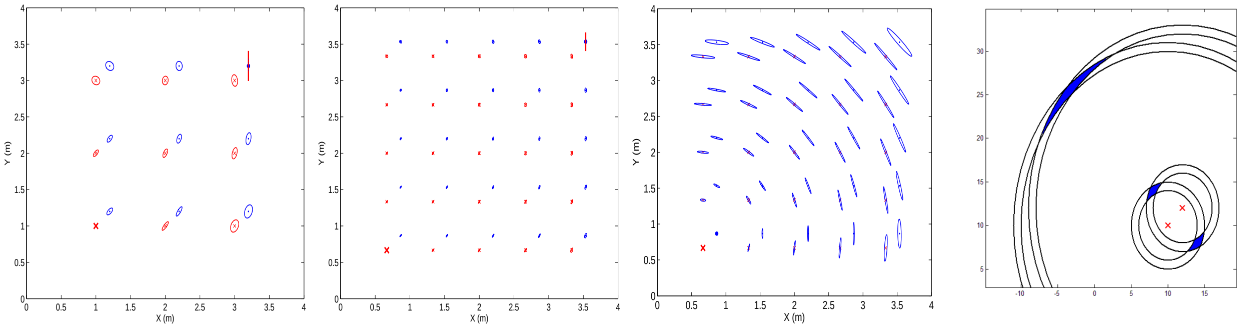
$$[J]_{ij} = \frac{\partial t_i(\Theta)}{\partial \theta_j} \tag{11}$$

Substituting this in Equation 9 and taking the expectation the Fishers Information matrix is,

$$F = J^T\Sigma^{-1}J \tag{12}$$

$$Cov\hat{\Theta} \geq [J^T\Sigma^{-1}J]^{-1} \tag{13}$$

If we assume that all the microphone and source locations are unknown, the Fisher Information matrix $J^T\Sigma^{-1}J$ is rank deficient and hence not invertible. This is because the solution to the ML estimation problem as formulated is not invariant to rotation, translation and reflection. In order to make the Fisher Information matrix invertible we remove

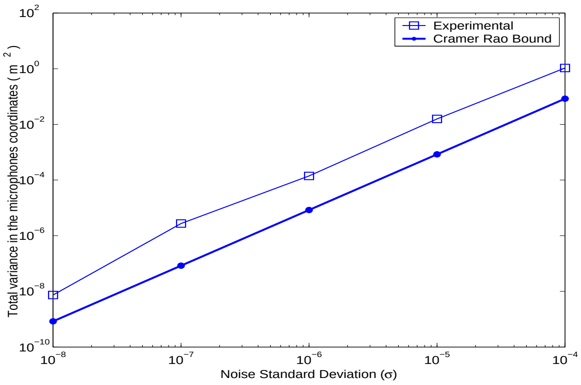


(a) (b) (c) (d)

Figure 4: 95% uncertainty ellipses for a regular 2 dimensional array of (a) 9 speakers and 9 microphones, (b)and (c) 25 speakers and 25 microphones. Noise variance for all cases is $\sigma^2 = 10^{-9}$. The microphones are represented as crosses (×) 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). In (c) the known nodes are close to each other and in (a) and (b) they are spread out one at each corner of the grid. (d) schematic to explain the shape of the uncertainty ellipses.

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.

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 as can be seen from Figure 4(a) and 4(b) which shows the 95% uncertainty ellipses for different number of sensors and actuators. 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 $O(n^2)$. So the linear increase in the number of unknown parameters, is compensated by the quadratic increase in the available measurements.



(a)

Figure 5: The total variance in the microphone coordinates with increasing noise standard deviation σ. The sensor network consisted of 16 microphones and 16 speakers. The Cramér Rao bound is also plotted.

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. In Figure 4(c) 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 reason for this can be explained for a simple case where we know the locations of two speakers (see Figure 4(d). A circular band centered at each speaker represents the uncertainty in the distance estimation. The intersection of the two bands corresponding to the two speakers gives the uncertainty region for the position of the sensor. 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 4(b) the known sensor nodes are on the edges of the network. As can be seen there is a substantial reduction in the dimensions of the uncertainty ellipses. In order to minimize the error due to Gaussian noise we should choose the three reference nodes (in 3D) as far as possible.

We also performed a series of simulations in order to compare the experimental performance with the theoretical bound. 16 microphones and 16 speakers were randomly se-

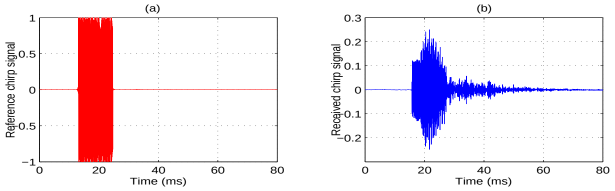


Figure 6: (a) The loopback reference chirp signal (b) the chirp signal received by one of the microphones

lected to lie in a room of dimensions $4.0m \times 4.0m \times 4.0m$. Based on the geometry of the setup and a known microphone capture start time, the actual TDOA between each speaker and a pair 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 Levenberg-Marqurdat method was used as the minimization routine. For each noise variance σ^2, the results were averaged over 2000 trials. Figure 5(a) shows the total variance of all the unknown microphone coordinates plotted against the noise standard deviation σ. The Cramér Rao bound for TDOA based Joint Estimation procedure is also shown. The estimator was unbiased for low noise variances.

5. Experimental Details and Results

We implemented a prototype system consisting of 6 microphones and 6 speakers. The real-time setup has been tested in a synchronized as well as a distributed setup using laptops. The ground truth was measured manually to validate the results from the position calibration methods. In order to measure the TOF accurately the calibration signal has to be appropriately selected and the parameters properly tuned. Chirp signals and 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. 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 8 kHz. The initial and the final frequency was chosen to lie in the common pass band 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 6(a) shows the chirp signal as sent out by the soundcard to the speaker. This signal is recorded by looping the output channels directly back to an input channel. Figure 6(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. One of the problems in accurately estimating the TOF is due to the multipath propagation caused by room reflections. This can be seen in the received chirp signal where the initial part corresponds to the direct signal and the rest are the room reflections. 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 [9] developed the Generalized Cross Correlation (GCC) method. In this method, the delay estimate is the time lag which maximizes the cross-correlation between filtered versions of the received signals [9]. 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 [9] $R_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} W(\omega)X_1(\omega)X_2^*(\omega)e^{j\omega\tau}d\omega$ 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 PHAT weighting. The 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 $W_{PHAT}(\omega) = 1/|X_1(\omega)X_2^*(\omega)|$. 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. In practice, the sensors' and actuators' three dimensional locations could be estimated with an average bias of 0.08 cm and average standard deviation of 3 cm (results averaged over 100 trials). Our algorithm assumed that the sampling rate is known for each laptop and the clock does not drift. However in practice the sampling rate is not as specified and the clock can also drift. Hence our real time setup integrates the distributed synchronization scheme using ML sequence as proposed in [1] to resample and align the different audio streams. 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.

6. Summary and Conclusions

In this paper we described the problem of localization of sound 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 multi stream A/V applications on platforms that are available virtually anywhere. We also derived important bounds on performance of spatial localization algorithms, proposed optimization techniques to implement them and extensively validated the algorithms on simulated and real data.

Appendix I

Converting the Distance matrix to a dot product matrix
Let us say we choose the k^{th} GPC as the origin of our coordinate system. Let d_{ij} and b_{ij} be the distance and dotproduct respectively, between the i^{th} and the j^{th} GPC. Referring to Figure 7, using the cosine law,

$$d_{ij}^2 = d_{ki}^2 + d_{kj}^2 - 2d_{ki}d_{kj}cos(\alpha) \tag{14}$$

The dot product b_{ij} is defined as

$$b_{ij} = d_{ki}d_{kj}cos(\alpha) \tag{15}$$

Combining the above two equations,

$$b_{ij} = \frac{1}{2}(d_{ki}^2 + d_{kj}^2 - d_{ij}^2) \tag{16}$$

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

$$X^* = X - \frac{1}{N}\mathbf{1}_{N\times N}X \tag{17}$$

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

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

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

$$b_{ij}^* = b_{ij} - \frac{1}{N}\sum_{l=1}^{N} b_{il} - \frac{1}{N}\sum_{m=1}^{N} b_{mj} + \frac{1}{N^2}\sum_{o=1}^{N}\sum_{p=1}^{N} b_{op} \tag{18}$$

Substituting Equation 16 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 $-\frac{1}{2}$.

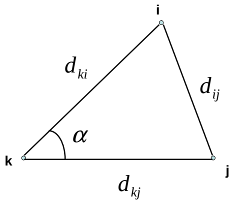


Figure 7: Law of cosines

References

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

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

[3] 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.

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

[5] 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.

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

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

[8] K. M. MacMillan, M. Droettboom, and I. I Fujinaga, "Audio latency measuremnts of desktop operating systems," in *International Computer Music Conference.*, 2001.

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

[10] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. 1981.

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

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