

COORDINATE-FREE CALIBRATION OF AN ACOUSTICALLY DRIVEN CAMERA POINTING SYSTEM

Evan Ettinger and Yoav Freund

{ettinger, yfreund}@cs.ucsd.edu
University of California at San Diego
Computer Science and Engineering

ABSTRACT

We present a camera pointing system controlled by real-time calculations of sound source locations from a microphone array. Traditional audio localization techniques require explicit estimates of the spatial coordinates for each microphone in the array. In addition, positional information for the camera is needed to use such techniques to drive a camera pointing system. Sometimes this positioning can be done by hand, but for large aperture microphone arrays with many elements this is impractical. We show that in this setting, where elements are placed in an ad-hoc manner, explicitly learning the microphone positions is an unnecessary step. We give a calibration method whose focus is learning the mapping from time delays between pairs of microphones to the associated pan and tilt a PTZ-camera should be given to point at. This curtails the need to explicitly learn the microphone and camera positions. We use this method to calibrate a real-time camera pointing system used by the UCSD interactive display.

Index Terms— Audio Sound Localization, A/V System Calibration.

1. INTRODUCTION

Camera steering is an important component of many surveillance, videoconferencing and interactive kiosk systems. Two camera steering systems are presented in [1, 2], with both having camera pointing logic driven by a small microphone array. In [1], the entire system is encased inside a housing making it self-contained and small enough so that precise locations of the microphones and the camera's focal point can be measured. A similar setup is presented in [2]. In this work, we instead present a camera steering system driven by a microphone array where all elements (including the camera) can be placed in an ad-hoc manner at much further distances. Moreover, we do not require or assume any knowledge of the locations of either the microphones or the camera.

Sound localization techniques via microphone arrays can be summarized into two major motivating paradigms. The first technique is based on maximizing the steered response power (SRP) of a beamformer [3], which, for example, can

be done by maximizing the energy of a delay-and-sum beamformer over a range of steering directions. The second paradigm is a two stage process that first estimates the time delays between pairs of microphones, and then with those delays solves a system of nonlinear physics equations to estimate where the sound source is that caused that set of delays [3]. Both methods rely on the fact that there is a unique mapping from a set of delays for all pairs of microphones to a spatial position in the room, and they require that the microphone positions be known in terms of a coordinate system for the room. In this work, we explicitly learn this mapping needed by both methods via a calibration process that collects a training set of examples from the mapping. This makes our work very flexible in terms of compatibility with both microphone localization paradigms.

Both localization techniques mentioned above require knowledge of a coordinate system wherein microphone positions are known. For small microphone arrays a coordinate system can easily be found by simply measuring the distances between microphones by hand as in [1]. If we want to be able to localize sounds in a large room accurately, then a large microphone array that spreads throughout the room is beneficial. However, measuring accurately by hand the relative distances now becomes much more difficult and positional errors on the order of 1-5cm can seriously degrade beamforming techniques [4]. Since doing such measurements is often too difficult, especially for arrays with many elements, many techniques have been developed to automatically calibrate the positions of the microphone elements [4, 5, 6, 7, 8]. Most techniques are based on using a carefully designed device to emit a set of special sounds used to accurately measure distances given knowledge of the speed of sound in the room. Typically distances from the device to the microphones, or inter-microphone distances are estimated. With these measurements, either a nonlinear optimization can be solved for the coordinates, or if pairwise distances can be estimated, then traditional multidimensional scaling (MDS) is often used [4, 5, 6, 7, 8]. These techniques are intended for one-time calibration for a system with positionally static elements and do not have self-consistency checks for continued positional ac-

curacy during general usage. Moreover, such calibration techniques are not geared towards camera steering, and as a result give us no insight into methods that let us place and orient a camera in the same coordinate system.

In this work, we avoid the need for solving for microphone positions explicitly and are still able to utilize the benefits of much of the sound localization research community’s work. Our intent is to develop a robust system for pointing a pan-tilt-zoom (PTZ) camera at sound sources in front of an interactive kiosk in a large room. If we were to directly use current sound localization techniques, then we would be required to discover the coordinates of not only the microphones, but also of the PTZ camera. This would require either direct measurement or new calibration methods to locate and orient the camera. Instead, we curtail the need of assigning the camera and microphones spatial coordinates by directly learning the mapping from the set of delays for pairs of microphones to the correct pan-tilt (PT) of the camera so that the sound source is centered in the field of view. We do this by collecting observations consisting of a set of delays between microphones for a fixed source location and the associated PT to center such a source. With this database of samples, we estimate via standard polynomial regression a fixed model for the system. This model describes how PT and delays vary with each other. The result is a function that can map a series of delays between pairs of microphones to a PT directive for our camera. Combining this mapping with known audio localization techniques is then very natural. Together this gives us a real-time implementation that can direct the camera at human speaking subjects. We have implemented such a system as part of the interactive display project at UCSD. A perspective view of the UCSD interactive display and a diagram of it can be seen in Figure 2.

We organize the paper by first briefly discussing the basics of estimating time delays between pairs of microphones in Section 2, and in particular discuss the popular method we use called PHAT based correlation. In Section 3, we discuss our method of creating a mapping between delays and camera PT from a database of examples using a variety of regression techniques. We give experimental results analyzing the bias in our models in Section 4, and afterwards conclude with a few closing remarks.

2. TIME DELAY ESTIMATION

The problem of time-delay estimation (TDE) can be summarized by the diagram in Figure 1. Even though in this work we do not assume knowledge of microphone or camera positions, it is useful to assume they are known and fixed for the discussion that follows. Let $m_i \in \mathbf{R}^3$ be the three dimensional Cartesian coordinates for microphone i . For a sound source located at position s and assuming a spherical propagation model, the direct path time delay between microphone

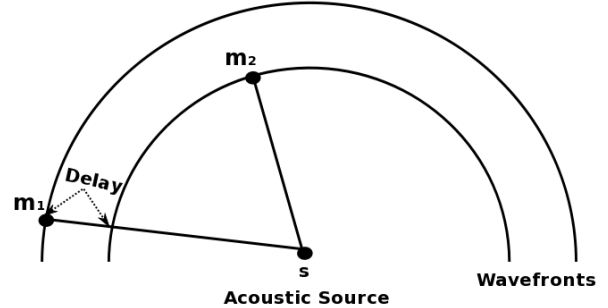


Fig. 1. 2-D Diagram depicting the problem of time delay estimation (TDE). Here m_1 and m_2 are microphones locations in space.

i and j can be calculated as

$$\Delta_{ij} = \frac{\|m_i - s\|_2 - \|m_j - s\|_2}{c} \quad (1)$$

where c is the speed of sound in the medium. Δ_{ij} is often called the *time delay of arrival* (TDOA) between microphone i and j . If f is the sampling rate being used, it is worth noting that largest the TDOA can be in terms of samples is $M = \|m_i - m_j\|_2 f / c$. In other words, Δ_{ij} is always in the range $[-M, M]$ and in practice can only be estimated to the nearest sample. This observation directly reveals the fact that close together microphones cannot have as wide a range of TDOAs as microphones that are spaced further apart. Placing microphones further apart allows for more variability in the feasible TDOAs, and hence, results in a better ability to discriminate between audio source locations in space.

Given n microphones there are $\binom{n}{2}$ unique pairs of Δ_{ij} that can be calculated. We let $\vec{\Delta} = (\Delta_{ij})_{i < j} \in \mathbf{R}^{\binom{n}{2}}$ be the vector that contains each of these unique TDOAs for a given audio source location. We will often call $\vec{\Delta}$ the *TDOA vector*. For a $\vec{\Delta}$ that corresponds to a true audio location there are many linear dependencies between the components because

$$\Delta_{ij} = \Delta_{kj} - \Delta_{ki} \quad \forall i, j, k$$

Therefore there are only $n - 1$ linearly independent coordinates of each $\vec{\Delta}$. In other words, the pairwise delays to just a single reference microphone uniquely determine the delays for all pairs.

When given a fixed Δ_{ij} for a pair of microphones, we can deduce from (1) that the set of feasible s positions that could have resulted in Δ_{ij} form one half of a two-sheeted hyperboloid in space. It follows that for a fixed $\vec{\Delta}$, the possible audio source locations that could have generated such a TDOA vector can be determined through finding the intersection among all such hyperboloids. We only require three TDOAs (4 microphones) to uniquely determine where in space the sound source is under idealized TDOAs. However, in practice we can only estimate each Δ_{ij} from the underlying

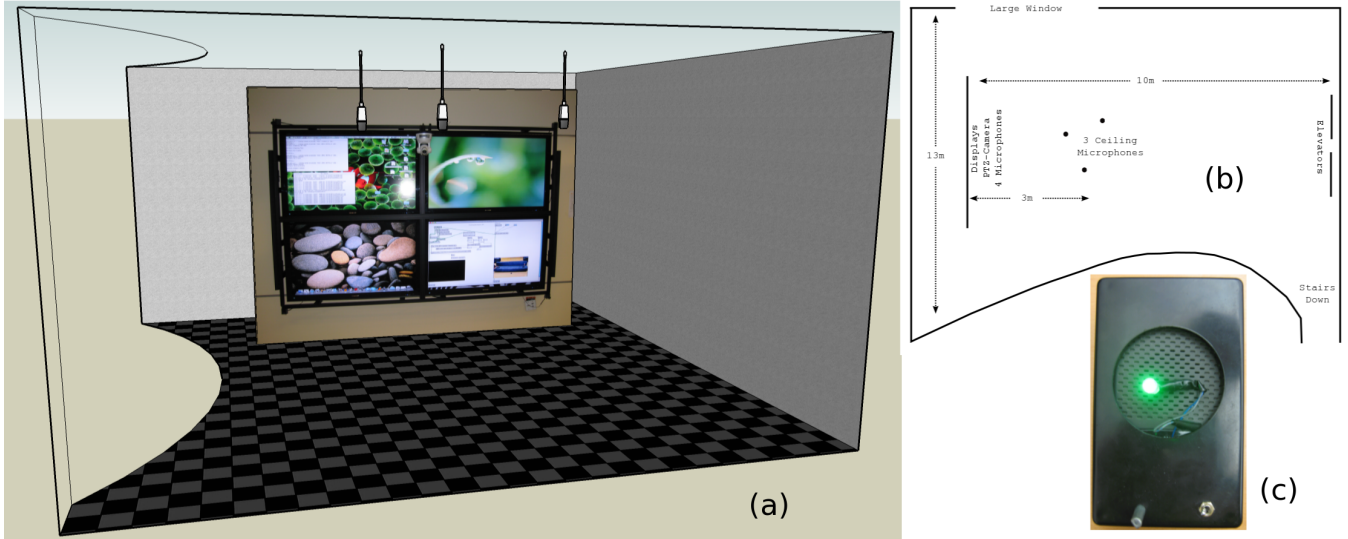


Fig. 2. (a) Perspective drawing (with photo) of the interactive display and lobby. (b) Diagram of lobby containing the display and (c) the device used to collect calibration data.

audio signals. This leads not only to imprecision from a finite sampling rate, but also from signal corruptions such as multipath reflections and reverberations. Often when using these estimated TDOAs we find that the intersection set of all hyperboloids is empty. Finding a source that is as close to this intersection as possible has been the focus of many researchers and a discussion can be found in [3]. Including more microphones in the array adds a redundancy to the information content of $\vec{\Delta}$, which can be leveraged to make a more robust TDE and localization system.

Accurate and robust TDE is the key to many types of localization systems. Background noise, multipath reflections and room reverberations complicate the estimation process. There has been much research on TDE in a variety of fields, and a review of many techniques can be found in [3]. Nevertheless, one very intuitive way to estimate a TDOA would be to calculate the cross-correlation of a pair of microphone signals and find the maximum. Unfortunately, due to corrupting factors of the signal this often gives maxima that are not near the true TDOA. One of the most popular TDE techniques, and the method used in this work, is a generalized cross-correlation (GCC) technique that utilizes the phase transform (PHAT). PHAT is very robust to noise and reverberations compared to other correlation based TDE techniques [3, 9]. Let $X_k(\omega)$ be the Fourier transform of microphone k . The GCC between microphone l and m is

$$R_{lm}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Psi(\omega) X_l(\omega) X_m^*(\omega) e^{j\omega\tau} d\omega$$

where $\Psi(\omega)$ is a weighting function for the GCC and $*$ denotes complex conjugation. The PHAT weighting of the GCC

is of the form

$$\Psi(\omega) = \frac{1}{|X_l(\omega) X_m^*(\omega)|}$$

The PHAT weighting has a whitening effect by removing amplitude information in the signals. The result is often a large spike in the GCC at the true TDOA. Hence the PHAT method for TDOA estimation is to let

$$\Delta_{ij} = \arg \max_s R_{ij}(s)$$

One of the primary competing forces in TDE is between placing microphone elements near to each other or far away. Placing elements near each other gives incoming signals that are very similar and hence result in more accurate TDE. However, placing elements farther apart allows for both more positional discrimination and coverage of a room. The system designer must weigh these competing forces in conjunction with the number of microphone elements available when constructing such an audio localization system.

In this work we are interested in ad-hoc placement of both the microphones and the camera. We do not assume knowledge of any m_i or the location and orientation of the PTZ-camera. Instead we would like to learn from a training set how to point the camera for a given $\vec{\Delta}$. If we plan to fit a regression function that describes how the pan and tilt of the camera changes with variations in $\vec{\Delta}$, then it would be worthwhile to understand some features of this variation so that we can reasonably select a regression model that can capture such behavior. Notice that when a speaker is close to the camera, small deviations in position correspond to large deviations in pan and tilt when compared to the same sized movements when far from the camera. Therefore, it would be reasonable

to believe that the predictive function from TDOAs to pan and tilt is most nonlinear for TDOAs that correspond to locations close to the camera. Moreover, notice that when nearby a pair of microphones small movements in position correspond to large changes in the TDOA relative to the changes in TDOA when far away. From this discussion, it's likely to believe that a linear model will not capture all the variation between these two sets of variates, but in areas where the variations in each match, a linear model may be very accurate. Moreover, in other areas that are non-linear it is not clear that the same linear model wouldn't be a good approximation. It is unclear at a surface level how much nonlinearity to expect. This motivates the inquiry into very simple regression models to fit this variation and then the examination of how and where these models perform below expectations.

3. REGRESSION CHOICES

In this section we describe the regression models we decided to use for describing the variation between $\vec{\Delta}$ and pan and tilt. For what follows assume that a training set of size N is given with observations of the form $y_i = (\theta_i, \psi_i)$, for pan and tilt respectively, paired with an estimated TDOA vector $x_i = \vec{\Delta}_i$ of length $p = \binom{N}{2}$. We organize the training set into matrices $Y \in \mathbf{R}^{N \times 2}$ and $X \in \mathbf{R}^{N \times p}$ where each observation is a row vector. In what follows, we briefly remind the reader of least squares linear regression and a variation called principal components regression. Further information on both can be found in [10].

3.1. Least Squares Linear Regression

For each column of Y , denoted Y_i , we fit a separate linear regression model. The linear regression model has the form

$$f(X) = \beta_0 + \sum_{j=1}^p X_j \beta_j$$

where X_j is the j^{th} column of X and β is the vector containing the coefficients in the linear model. The least squares (LS) solution to linear regression chooses the model that minimizes the residual sum of squares (RSS)

$$RSS(\beta) = \sum_{i=1}^N (y_i - f(x_i))^2$$

When X is full rank the LS solution can be written in closed form as $\beta = (X^T X)^{-1} X^T Y_i$. It is known that if the true model of data generation is linear, then the LS estimator is the minimum variance unbiased estimator of β .

3.2. Principal Components Regression

Often in regression it's advantageous to trade a small amount of bias for a large reduction in variance. Principal components

regression (PCR) attempts to describe the $k < p$ orthogonal directions in the feature space that preserve most of the variance in X . After centering X , PCR projects the data onto the k -dimensional subspace spanned by these directions, and learns a LS linear regression model to predict Y in this reduced space. Although the resulting model is slightly biased if the true underlying model is linear in the original feature space, typically the reduction in parameters by PCR results in a dramatically smaller variance in estimating its parameters and hence a corresponding smaller RSS. In addition, an attractive feature of PCR is its denoising properties of X , which is a major motivating factor in choosing it and of which we discuss further below.

Calculating the top principal components is achieved through principal components analysis (PCA). The first principal component v_1 is defined as the direction in the feature space that gives the projections of X onto it the highest sample variance. Subsequent principal components have the property that when X is projected onto them they have the next largest sample variance subject to being orthogonal to all previous principal components. Solving for the principal components can be shown to be solved by an eigendecomposition of the covariance matrix $X^T X = V D V^T$. The column of V with corresponding largest eigenvalue is the first principal component, and the eigenvector with next largest eigenvalue is the second, and so on. The percentage of variance in X explained by a principal component is the ratio between its corresponding eigenvalue and the trace of D . Typically one retains the top k principal components that describe most of the variance and discards the remaining deeming them as observation noise.

3.3. Higher Order Polynomial Fits

We can fit general polynomials using the LS approach by simply extending X to contain higher-order combinations of features. For example, in the quadratic regression (QR) analysis used in the experiments that follow, appending to X the squares and cross-terms of features and applying the LS method gives the desired parabolic fit. The same procedure can be repeated for the data matrix X used in PCR.

4. EXPERIMENTAL SETUP

The UCSD interactive display consists of four 52 inch 1080p LCD TVs arranged in a 2 by 2 grid. A Macintosh G5 with 2.66 GHz Dual-Core Intel Xeon processor and 2 GB of RAM drives the displays. The G5 is connected via firewire to a MOTU 896HD that preamplifies and digitizes the audio channels. The display has an array consisting of seven CAD CM100 overhead choir microphones. The microphones are unidirectional condenser microphones that have a frequency response range of 40 Hz to 20 kHz. One microphone is placed at each corner of the 2 by 2 grid of TVs, and 3 microphones are placed in a roughly triangular pattern hanging from the

ceiling approximately 3m away from the display (4m from the floor) and approximately 1m apart from each other. A plastic arm is used to mount a Sony SNC-RZ30N networked PTZ-camera directly along the vertical central axis of the 2 by 2 grid slightly above standing height level. The camera records at 640×480 resolution at 30 frames per second and can take commands via the network through a CGI interface for pointing, zooming and many other operations. The lobby is approximately $13m \times 10m \times 4m$ in dimensions. A perspective view of the display including a photo taken directly facing the display is shown in Figure 2.

The device used to collect all the data in the experiments to come is shown in Figure 2c. It consists of a simple radio and a green LED attached to a 9V battery with a switch and dimmer all in a plastic encasing. We will call this the *calibration device* from here on. The radio component of the calibration device can be tuned to a nonexistent station that emits noise that is very close to white. This random noise typically has the most consistent TDOA vector estimates from the PHAT technique. A simple color thresholding detector was written to find the LED in the camera’s field of view using Max/MSP and Jitter [11]. The result is a real-time control of the PTZ-camera to keep the LED centered in the field of view, and a constant white noise to calculate TDOAs for. The calibration device is used to collect samples of TDOA vectors in unison with where the camera is pointing to center the green LED in its field of view. The camera can be queried as to what pan and tilt it is currently pointed at whenever a TDOA vector is collected to append this information as a complete data observation.

The calibration procedure we examine in this work is to take the calibration device and collect several $(\vec{\Delta}, \theta, \phi)$ observations throughout the room as a training set for a simple regression function. We walked slowly around the display lobby with the LED on and the radio emitting noise. We use 100ms audio windows with a 75ms overlap, giving a TDOA, pan and tilt observation every 25ms. The first data collection phase for training is aimed to be as simple as possible, and required a researcher to walk around the room with the calibration device for 10 to 15 minutes while the light detector was running to continually keep the LED centered in the camera’s field of view. Another utility takes as input the 7 audio channels and calculates the TDOA vectors and continually queries the camera for updates on its current pan and tilt positions to append them for complete observations.

5. RESULTS

The result of the training set collection is a dataset of close to 28k observations. We noticed that when a estimate for Δ_{ij} was incorrect, it typically had a very large deviation from what was often consistent. To remove such noisy observations, we performed some simple outlier removal by thresholding the magnitudes of the $\vec{\Delta}$ projections onto the bottom

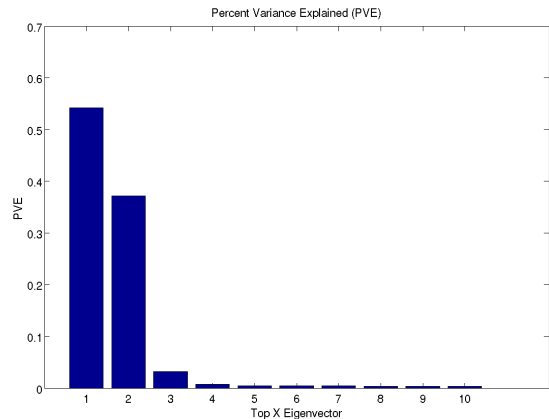


Fig. 3. Percentage of variance explained by top X eigenvector. The top 3 eigenvectors dominate and the rest are noise.

global PCA eigenvectors (orthogonal space) leave approximately 20k observations remaining as our training set. We then did a PCA analysis of just the $\vec{\Delta}$ parts of this training set. Figure 3 shows the percentage of variance explained by the addition of each eigenvector. It’s clear that the top two eigenvectors dominate most of the variance explained, and that the 3rd eigenvector seems to have a significant advantage over the remaining ones. The total percent variation captured by the top 3 eigenvectors is nearly 90%. This follows from the fact that there are 3 spatial degrees of freedom that were examined during the training data collection. Moreover, two of these spatial directions had much more spatial variance than the third, ceiling-to-floor, spatial direction. The room is simply much larger in width and breadth than the variance in observation heights, which matched typical heights that human speakers could appear at. This analysis lead us to choose three eigenvectors for the principal components regression analysis that follows.

From this training set with outliers removed we have nearly 20k observations. From here we learn a simple linear least-squares regression (LS) model, a linear principal components regression (PCR), a quadratic least-squares regression (QLS) and a quadratic principal components regression (QPCR). We would like to analyze how the bias-variance trade-off of these simple models behaves as function of physical position of the sound source in the lobby. In other words, in what areas do these simple models perform well, and where does the inherent nonlinearity of the problem cause large bias? With these questions in mind we collect a test set of data in a similar fashion to the training set. We place the calibration device at a fixed height (approximately 1m from the floor) and roll it along straight lines using a rolling chair. We repeat this process for each of the 13 lines in the grid depicted in Figure 4b. This results in a variety of observations that cover a representative set of the spatial variability in the room relevant for

human speakers. Moreover, using white noise as our sound source will simulate the behavior of our model under conditions where TDE is highly optimized. This gives us insight into isolating the effects of the model assumptions. Each of the 13 grid lines was traversed back and forth two times during the collection phase. This test set of observations collected along the grid will be used to analyze the predictive power of each of the regression models. It is worth noting that although using the light detector will not give a “ground truth” comparison, it is nevertheless very close and thus a fair comparison. The light detector observations for pan and tilt are very consistent and stable when the calibration device is stationary. The camera was directed to recenter the LED whenever the center of color thresholded pixels exited a small 20 by 20 pixel box in the center of the image. Therefore, these pan and tilt observations from this test set should be considered as very close to having the sound source centered, which is the ultimate goal of this camera pointing system.

Figure 4a depicts the embedding of the TDOA vector components of the entire grid test set onto the top 2 eigenvectors from the PCA learned from the training set. The zoomed in portion depicts lines 9-13 in red and lines 1-6 in blue in the same orientation as the diagram in Figure 4b. The curved nature of each line can be observed from such plots. Even though the spatial location of the sound source is varying along a straight line in space, the corresponding location in the TDOA vector space corresponds to slightly curved trajectories. It is clear that a linear model for spatial location is not going to fully capture all the variation, but nevertheless the grid structure is still very recognizable in even just the top 2 eigenvectors indicating that a linear model may not be a poor approximation in these regions.

Another thing to observe, especially in the smaller full plot of the entire embedding, is the noisy nature of the observations themselves. Although, the majority of the TDOA vectors are estimated along trajectories, there is quite a bit of noise. This is attributed to the noisy nature of time delay estimation; although the majority of TDOAs consistently follow a fixed trajectory, observations are occasionally noisy due to channel corruptions or reflections from room surfaces. This noisiness highly depends on location. For example, the observation noise from lines 1 - 6 increases as a function of distance from the display, and hence the microphones. It is also interesting to see that lines 7 and 8 are particularly noisy. This is most likely due to the fact that these lines are still in front of the microphones on the ceiling, and the data was collected while the radio faced the display. This causes the direct path to these ceiling microphones to be not as strong as reflections of the floor or front wall. This resulted in a variety of TDOA estimates for pairs involving the ceiling microphones at these particular locations.

Figure 4c compares the predictions from the simple linear LS model to the pan and tilt recorded from the light detector. The dots in black are the predicted pan (or tilt) from the model

for each TDOA vector observation. The green line depicts the pan (or tilt) from the light detector. Finally the red line depicts an exponential moving average (EMA) of the model predictions over time. In other words, the EMA prediction, p_t , at time t is calculated with update $p_t = (1 - \alpha)p_{t-1} + \alpha f(\Delta_t)$, where $f(\Delta_t)$ is the prediction of the raw observation at time t . We chose $\alpha = 0.1$. The EMA line should give us a sense of what the true model predictions are by smoothing out the observation noise. In doing so, we can compare the light detector observations to the EMA line and get a sense for the bias in our model. Remember that for each grid line we collect data along two round trips across the line, which is why we see the periodic nature in the data. It’s also worth noting that the light detector observations are slightly lagged from the truth. This is because the camera only recenters after the light detections exit a 20×20 pixel box in the center of the image. It’s easy to see the bias of the model due to the nonlinear nature of some of the variation in these plots. For example, in the line 3 plot for tilt, the slope of the EMA line does not match the rate of change from the light detector. The attempt to capture the portions of tilts that occur when a sound source is close to the display is modeled to closely to constant when this is clearly not the case. However, this bias shrinks for lines that are further from the display. It is also worth noting that as you move further from the display the TDOA vectors themselves become more noisy, which can be observed in the plots for line 13. Nonetheless, the bias in tilt is still the most dramatic for most grid lines, since this straight line motion in space does not correspond to linear changes in tilt. On the other hand, for pan the changes are very near to linear.

Table 1 gives the root-mean squared error (RMSE) between the EMA of the model predictions and the observations from the light detector for each of the regression models. Surprisingly, the PCR methods do not show any advantage over their LS counterparts, and in fact are significantly worse. There is no advantage for trying to remove additional noise from the observations. The variance in estimating the LS model is most likely very low because of the large quantity of observations collected in the training set. Moreover, the bias in the both the LS and PCR models should be similar because they are both linear approximations, giving the LS model an advantage in total RMSE. The poorer performance of PCR can only be attributed to the fact that some signal is being removed by projecting down to only the top 3 eigenvectors and not the entire space.

We also provide information about quadratic models of both. The quadratic models do show a slight improvement over the linear model in almost every line, which indicates that significant improvement with nonlinear models is possible. The fact that line 13 tilt in a QLS model performs worse than the LS-tilt might be indicative of poor boundary extrapolation effects from the QLS model, but it’s not enough to conclude that here. Because of inaccuracies from the lag of the light detector the RSME results here should be treated

Model	Grid Line Number													avg
	1	2	3	4	5	6	7	8	9	10	11	12	13	
LS-pan	4.31	3.34	2.77	2.26	2.22	4.33	5.99	6.54	3.56	3.95	3.20	3.83	3.96	3.87
PCR-pan	7.30	5.08	4.67	5.07	4.78	5.60	6.85	6.26	4.54	4.86	4.99	8.24	8.04	5.87
QLS-pan	4.25	3.12	2.50	2.08	2.10	3.47	4.32	5.46	3.12	3.50	2.09	3.03	3.35	3.26
QPCR-pan	7.55	4.49	3.95	4.41	4.27	4.81	5.47	5.76	3.53	4.47	4.30	8.07	7.42	5.27
LS-tilt	5.15	5.74	7.57	7.67	7.50	5.48	3.33	8.40	5.63	4.74	3.90	5.15	4.48	5.75
PCR-tilt	4.02	6.83	9.13	9.23	9.27	6.17	2.65	11.12	6.61	5.21	3.03	2.29	3.37	6.07
QLS-tilt	4.43	4.40	4.32	4.32	4.47	4.23	3.06	3.14	3.32	3.23	2.17	4.59	6.13	3.99
QPCR-tilt	5.21	5.62	6.47	6.25	6.53	5.95	3.83	7.66	5.72	4.61	2.43	3.75	5.72	5.36

Table 1. RMSE (in degrees) of different regression models for each grid line.

as an upper bound of the L1 error in pan and tilt. With a QLS model we can get almost uniformly throughout the room within four degrees error in both pan and tilt. This kind of uniform bound implies that predictions made for distances that are farther from the display have higher spatial error than of predictions for sources that are closer. To get specialized accuracy in regions a more piecewise model should be explored. Nevertheless, it is still a nice result that such a simple model of the variation can work well enough for many practical applications.

6. CONCLUSION AND FUTURE WORK

We present in this work a microphone coordinate-free way of predicting where to point a camera from a TDOA vector. It involves a brief, yet simple, calibration phase, and then learning a simple regression function. Although, this model is biased, as expected, the amount of bias is not overwhelmingly large and the model is surprisingly accurate in many locations of interest. In addition, since the model does not assume that the microphone elements are synchronized, we can allow for a misalignment that is fixed. The need for examining how to further represent the surface more accurately is needed for future work, and using kernel methods or piecewise models such as splines and regression trees must be examined.

Moreover, our system has the attractive attribute that as long as new observations can be collected, the system can be further calibrated and refined. With the help of a face detector, we could collect observation points from humans interacting with the system. This would only require synchronizing face detections with TDOA vectors from real users. This would allow for a lifelong learning model that makes the system more accurate the more that it is used. Such a continuous stream of observations could be fed into a self-updating prediction model. We could also examine with this feedback information in what regions of space is the system poorly calibrated. This would give us insight into how to further refine the prediction model. Another direction of research could be how to most quickly recover an accurate prediction model when a few microphone elements have been jostled in position slightly. A

lifelong learning model may be able to recover from small movements in the microphone positions and overcome prediction biases. All such directions should be fruitful for future research.

7. REFERENCES

- [1] H. Wang and P. Chu, "Voice source localization for automatic camera pointing system in videoconferencing," in *ICASSP '97: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97) -Volume 1*, Washington, DC, USA, 1997, p. 187, IEEE Computer Society.
- [2] Yiteng Huang, J. Benesty, and G. W. Elko, "Passive acoustic source localization for video camera steering," in *ICASSP '00: Proceedings of the Acoustics, Speech, and Signal Processing, 2000. on IEEE International Conference*, Washington, DC, USA, 2000, pp. II909–II912, IEEE Computer Society.
- [3] M. Brandstein J. DiBiase, H. Silverman, *Robust localization in reverberant rooms. In M. Brandstein and D. Ward Microphone Arrays.*, Springer-Verlag, 2001.
- [4] J.M. Sachar, H.F. Silverman, and W.R. Patterson, "Microphone position and gain calibration for a large-aperture microphone array," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 1, pp. 42–52, Jan. 2005.
- [5] V.C. Raykar and R. Duraiswami, "Automatic position calibration of multiple microphones," *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 4, pp. iv–69–iv–72 vol.4, 17–21 May 2004.
- [6] S.T. Birchfield and A. Subramanya, "Microphone array position calibration by basis-point classical multidimensional scaling," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 1025–1034, Sept. 2005.

- [7] I. McCowan, M. Lincoln, and I. Himawan, "Microphone array shape calibration in diffuse noise fields," *Audio, Speech, and Language Processing, IEEE Transactions on [see also Speech and Audio Processing, IEEE Transactions on]*, vol. 16, no. 3, pp. 666–670, March 2008.
- [8] E. Hörster, R. Lienhart, W. Kellermann, and J.-Y. Bouguet, "Calibration of visual sensors and actuators in distributed computing platforms," in *VSSN '05: Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, New York, NY, USA, 2005, pp. 19–28, ACM.
- [9] P. Svaizer, M. Matassoni, and M. Omologo, "Acoustic source location in a three-dimensional space using crosspower spectrum phase," in *ICASSP '97: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97) -Volume 1*, Washington, DC, USA, 1997, p. 231, IEEE Computer Society.
- [10] J. Friedman T. Hastie, R. Tibshirani, *The Elements of Statistical Learning*, Springer-Verlag, 2001.
- [11] "Max/msp website," <http://www.cycling74.com>.

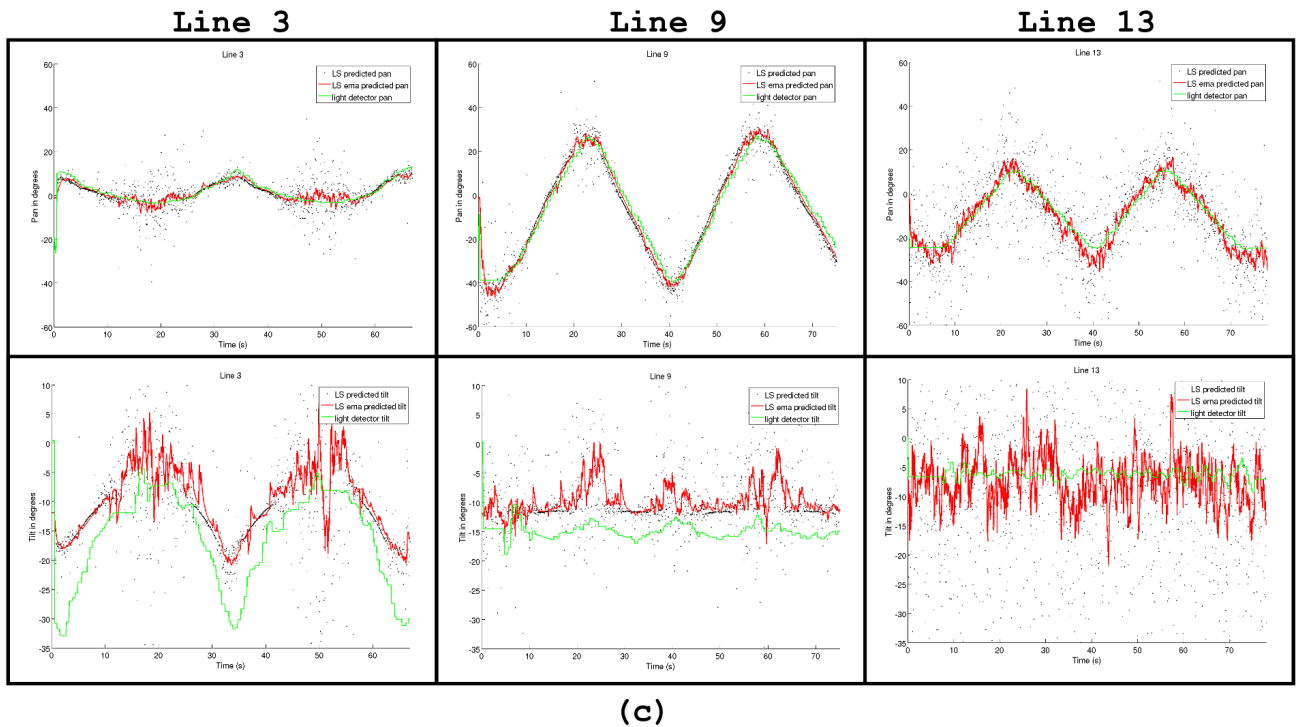
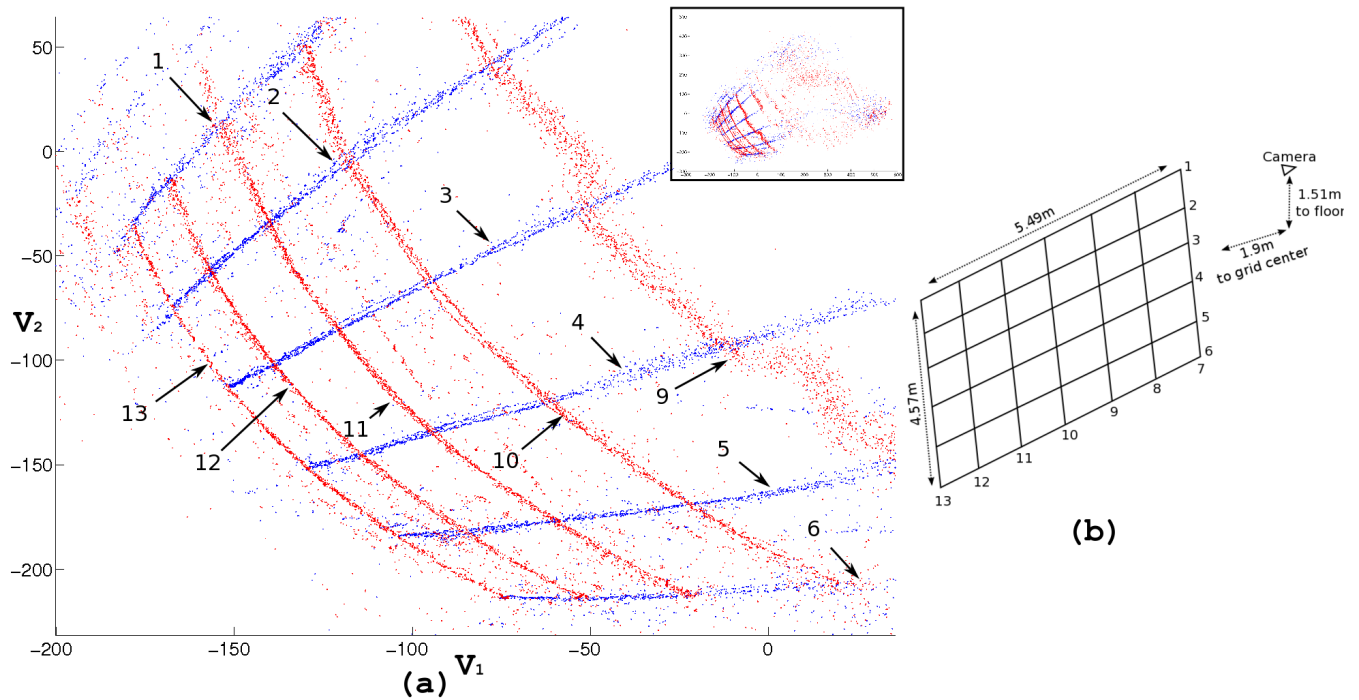


Fig. 4. (a) Embedding of the TDOAs collected from the grid onto top 2 eigenvectors. The entire embedding is shown small in the upper right corner and a zoomed in portion of the same embedding is shown larger. (b) To the right is a diagram of the equispaced grid over which data was collected. (c) Below are 3 selected lines and the LS predicted value for each TDOA collected. Also depicted in red is an exponential moving average of the predictions ($\alpha = 0.10$), and in green where the camera was pointing to center the LED.