MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Position and Trajectory Learning for Microphone Arrays

Paris Smaragdis, Petros Boufounos

TR2007-001 January 2007

Abstract

In this paper we tackle the problem of source localization by example. We present a methodology that allows a user to train a microphone array system using signals from a set of positions and trajectories and subsequently recall the localization information when presented with new input signals. To do so we present a new statistical model which is capable of accurately describing features from the cross spectra of the microphone signals so as to model the room responses from all positions of interest. We further extend this model to allow modeling of sequences of positions, thereby also enabling the learning and recognition of trajectories. Because of its learning nature this method provides practical advantages in setting up a microphone array, by not requiring favorable room acoustics, careful element positioning or uniformity of sensors. It also introduces an approach to localization which can be extended to other problems requiring models of transfer functions. We present tests on synthetic and real-world data and present the resulting recognition rates for a variety of situations.

IEEE Transactions on Audio, Speech and Language Processing

Position and Trajectory Learning for Microphone Arrays

Paris Smaragdis, *Member, IEEE*, Petros Boufounos, *Student Member, IEEE*

***Abstract*— In this paper we tackle the problem of source localization by example. We present a methodology that allows a user to train a microphone array system using signals from a set of positions and trajectories and subsequently recall the localization information when presented with new input signals. To do so we present a new statistical model which is capable of accurately describing features from the cross spectra of the microphone signals so as to model the room responses from all positions of interest. We further extend this model to allow modeling of sequences of positions, thereby also enabling the learning and recognition of trajectories. Because of its learning nature this method provides practical advantages in setting up a microphone array, by not requiring favorable room acoustics, careful element positioning or uniformity of sensors. It also introduces an approach to localization which can be extended to other problems requiring models of transfer functions. We present tests on synthetic and real-world data and present the resulting recognition rates for a variety of situations.**

***Index Terms*— Localization, phase wrapping, microphone arrays**

I. Introduction

SOURCE localization using microphone arrays is a subject that has received significant attention in the signal processing literature. Such systems are often used, for example, to discover the location of active speakers in a teleconference setting, track vehicles in an outdoor environment, steer surveillance cameras towards suspicious sounds, etc. This type of functionality can be achieved using a variety of techniques depending on the constraints and expectations of the system at hand. One family of approaches takes advantage of the time difference of arrival (TDOA) of a source signal as measured across multiple microphones. These time differences can be estimated using a variety of techniques [1], [2], [3], [4], [5], and once obtained can be used in conjunction with the positions of the microphones to estimate the originating location of the source [6], [7], [8]. Such approaches exhibit the advantage of being fairly efficient and adequately robust for some real-world applications. Another category of source localization algorithms measure the likelihood that the input has originated from a set of locations instead of inferring the location from the input. These algorithms employ a wide variety of computational techniques which involve subspace methods [9], cross spectral measures or beamforming and/or probabilistic formulations [10]. These methods are often less efficient to compute than the TDOA methods, but they provide an increased robustness and can operate more reliably in environments with multiple sources. There have also been some formulations using a learning methodology, but they have been quite ad-hoc [11] and have fallen out of favor.

Regardless of the localization technique used, it is imperative that the room acoustics are accommodating so as to not exhibit confusing reflections, the positions of the sensors are known, and the microphones have similar responses. Non-compliance to any of the above conditions can result in detrimental accuracy in localization estimates.

In this paper we address the source localization issue from a different viewpoint. We will examine the case where the positioning and response of the microphones is unknown, as is the surrounding acoustic environment. We will consider the case where strong room reflections exist in addition to constant background sounds. In order to deal with these issues we will use a learning methodology.

The methodology that we propose has two stages, an unknown array system is trained with sounds emanating from a variety of locations. The response characteristics from each location are used as training features, and subsequently used for recognition. Using this approach spurious reflections, or microphone inconsistencies do not pose a practical issue since they are learned as part of the process.

Obviously, training for the specific acoustic environment where the system is to be used is the price to pay for not having to deal with array calibration. Training, however, is acceptable in many applications involving fixed arrays and in our experience does not pose a significant burden.

The remainder of this paper is divided as follows. In section II we will introduce the features and the statistical model we use for training to discriminate positions and

trajectories. Section III will introduce the training and classification methodology that we propose. Sections IV and V present results from synthetic and real-world data experiments under various conditions.

II. Location Modeling

The problem we set forth in this paper is stated as follows: Given an arbitrary array configuration and a randomly positioned source, we desire to learn the source's position from the observed data, so that whenever another source is placed in that position we can reliably confirm it. We will also consider the dynamic case where the sources are allowed to constantly change positions and follow specific trajectories, which must also be learned and recognized.

In the following sections we will lay out the framework of this work. We will first examine the features needed to perform this task, and subsequently provide a statistical model for modeling positions and trajectories using these features.

A. Location features

In order to have invariance from the nature of the incoming sources and array characteristics, the features that we should employ will have to be relative features between the microphone inputs. Using this approach training will not be influenced by the nature of the inputs, but rather by the cross-microphone relations. To this end we employ the relative magnitude and phase of the spectra of each microphone input.

We start considering a two element array setup. Using two microphones, we receive one signal from each denoted by $z(t)$ and $y(t)$. Assuming local stationarity, we perform short-time spectral analysis to determine the frequency domain counterparts, which we denote as $Z_\omega(t)$ and $Y_\omega(t)$ for each frequency ω at time t. As features we will use the log cross-magnitude and the cross-phase of the two signals at each frequency ω. Both features can be computed simply using one complex logarithm:

$$R_\omega(t) = log\frac{Z_\omega(t)}{Y_\omega(t)} \tag{1}$$

This computation places the log of the ratio of the magnitudes of the inputs in the real part of R_ω and their phase difference in the imaginary part:

$$\begin{aligned} \Re(R_\omega(t)) &= log\frac{\|Z_\omega(t)\|}{\|Y_\omega(t)\|} \\ \Im(R_\omega(t)) &= \angle Z_\omega(t) \cdot Y_\omega(t)^* \end{aligned} \tag{2}$$

The information contained across all R_ω is usually sufficient to discriminate between various positions around the array. Although singular cases exist they are hard to come by in real world setups. The positioning and directionality response of the microphones, as well as the acoustic environment response, are the main factors in defining the discriminatory power of the array. If appropriately chosen it is possible to localize a very wide range of positions using only a few elements (section V).

There are many possible ways to extract equivalent features for arrays with more than two elements. The most straightforward method is to consider the relative magnitude and phase of all pairs of elements and use all of them simultaneously. This has the effect of increasing the dimensionality of our features (and the computational complexity when processing them) by a factor of $\frac{N!}{2(N-2)!}$ for an array of N elements.

B. Location model

Using the aforementioned features we will now construct a model we can train and then use for recognition. A first rudimentary model would be to estimate a complex Gaussian distribution for each R_ω and use that for subsequent classification. However, this approximation is not always appropriate. Although the real part of our features can be adequately modeled by a Gaussian distribution, this is not the case with the imaginary part, which represents a phase value. Phase is estimated in a wrapped form and is bound between $-\pi$ and π. Using a Gaussian distribution to model this data can result in significant estimation errors. To illustrate this issue consider the following example from a real recording of speech from two microphones. Figure 1 displays on the left the histogram of relative phase estimates at around 6300 Hz. We can see that they can be described using a Gaussian model. However consider the relative phase distribution around 7800 Hz, as shown in in the right plot of figure 1. Due to the phase being wrapped around $\pm\pi$ the result is a bimodal distribution that is poorly fit by a Gaussian model. Even when the wrapping effect is not that severe, the mean of the estimated Gaussian will be biased towards zero, as compared to where the distribution modes truly are.

Therefore we need to consider a different model for the phase angle so that we can estimate likelihoods with better accuracy. To address this we model the distribution of the relative phase as a Gaussian wrapped around the interval $[-\pi, \pi]$. This means that the phase data is assumed to be normally distributed had we not had wrapping. By looking at the histograms in figure 1, we can see that this is a better model. The addition of the wrapping

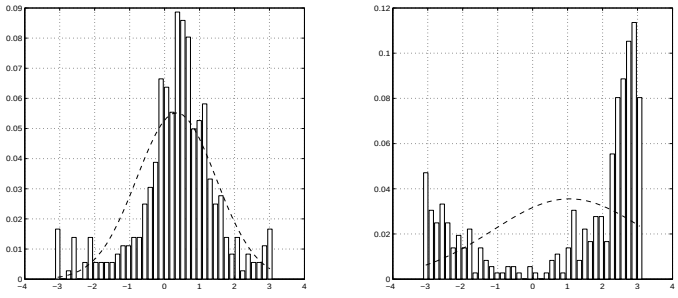


Fig. 1. Histograms and corresponding Gaussian fits of relative phases from two different frequencies. Due to phase being bound from $-\pi$ to π a Gaussian model is not sufficient to model the data.

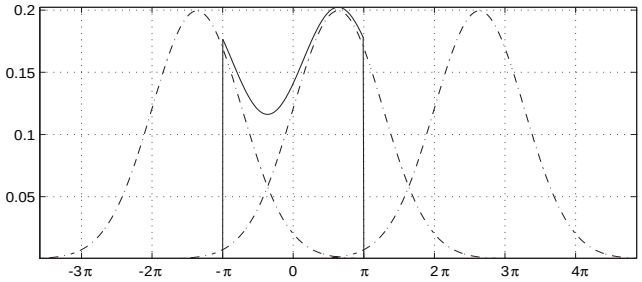


Fig. 2. The wrapped Gaussian model. The dotted line represents the Gaussians to be summed and the solid line their addition in the $[-\pi, \pi]$ interval.

of the distribution is meant to mirror the wrapping that phase undergoes. The resulting distribution is defined as:

$$P_{\Im(R_\omega)}(x) = \begin{cases} \sum_{k\in\mathbb{Z}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu+k2\pi)^2}{2\sigma^2}} & x \in [-\pi, \pi] \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Although k ranges from $-\infty$ to ∞, in practice a range from -2 to 2 is often an adequate approximation (and the one we have used in all our experiments). We demonstrate how this model works in figure 2. Assuming the data in the right plot of figure 1 we use $k \in [-1, 1]$ to obtain three Gaussians which are shown with the dashed lines. The distribution that they approximate, which is only defined between $-\pi$ and π, is shown with the solid line. We can see that it corresponds to the bimodal nature of the data much better than a single Gaussian.

We will now develop an algorithm to estimate a complex Gaussian model in which the imaginary part is wrapped in the interval $[-\pi, \pi]$ and the real part is not, as is the case with R_ω. We can treat the sum of Gaussians in the imaginary domain as a constrained Gaussian mixture and adapt the parameters as such [12]. To do so we will find the mean μ and variance σ^2 of our model using an Expectation-Maximization approach. Therefore having a complex random variable $x \in \mathbb{C}$, with the aforementioned properties, we use the following steps to iteratively update μ and σ:

1) Start with initial estimates $\mu = 0$ and $\sigma = 1$.
2) Compute the distance of the sample set from the unwrapped Gaussians using:

$$D_k(x) = x - \mu + k2\pi\sqrt{-1} \tag{4}$$

3) Compute the posterior probabilities of the sample set for each of the unwrapped Gaussians using:

$$\begin{aligned} Q_k(x) &= \tfrac{1}{\pi\sigma^2} e^{-\frac{D_k(x)^2}{\sigma^2}} \\ P_k(x) &= \tfrac{Q_k(x)}{\sum_k Q_k(x)} \end{aligned} \tag{5}$$

4) Update the variable μ as a mean weighted by the posteriors:

$$\mu \leftarrow \mu + \left\langle \sum_k D_k(x) P_k(x) \right\rangle \tag{6}$$

where $\langle\cdot\rangle$ denotes sample expectation. Furthermore ensure that the imaginary part is wrapped around $[-\pi, \pi]$ by setting:

$$\Im(\mu) \leftarrow [\,(\Im(\mu) + \pi)\ mod\ (2\pi)\,] - \pi \tag{7}$$

5) Likewise update the variable σ using:

$$\sigma \leftarrow \sqrt{\left\langle \sum_k D_k(x)^2 P_k(x) \right\rangle} \tag{8}$$

6) Repeat from step 2 until convergence.

Convergence is rapid and usually concludes to a satisfactory solution by the 10th iteration. For numerical reasons it is best if step 3 is performed in the log domain to reduce underflow effects due to the product operation.

III. Learning to Localize

In this section we will show how we can employ the model we just introduced in order to learn and subsequently identify positions and trajectories.

A. Learning positions

The methodology for learning to localize a position is fairly straightforward now that we have a model. During the training phase the features from each location are extracted using estimates of $R_\omega(t)$ by applying equation 1 on the short time Fourier transforms of the microphone inputs. For each position we compute the model we just introduced at each ω and obtain a series of μ_ω and σ_ω. To localize an unknown input we can

extract the features and evaluate the likelihood of the learned models for each position using:

$$P(x) = \prod_{\omega} \sum_{k} \frac{1}{\pi \sigma_{\omega}^2} e^{-\frac{\|(x_{\omega} - \mu_{\omega} + k2\pi\sqrt{-1})\|}{\sigma_{\omega}^2}} \quad (9)$$

The position model that provides the highest likelihood reveals the most likely position that the input has originated from. This process makes the assumption that each position has a unique set of relative magnitude and phase. Although this is not strictly true for all configurations, diligent use of microphone directional responses and environmental reverberation can help in minimizing any location ambiguities.

One issue that rises with this approach is the spectral consistency between training and testing sounds. An estimate of R_{ω} can be unreliable when the source used for training has little energy at frequency ω. If that is the case then classification will be poor since it will be contrasted with excessively noisy data. To remedy this we can keep track of the frequency content of the training data and perform classification by evaluating equation 9 on only a few of the most prominent frequencies ω. This also provides a computational advantage, significantly reducing the operations required for classification. To obtain a good classification estimate it is also important that the training source spectrum and the source to be classified have non-negligible energy in overlapping spectral areas. It is easy to satisfy this constraint by choosing the training source to be either a wideband signal or of similar type to the source to be classified.

B. Learning trajectories

Learning trajectories is somewhat more complicated. Identifying a trajectory involves having temporal knowledge of the series of positions that the source has gone through. A straightforward method to include temporal information to our training is to employ a Hidden Markov Model and Viterbi training [13].

As before we will extract the features of each time point using the features introduced in section II-A and model each state with the model introduced in section II-B. This model will be incorporated in a Viterbi training loop to learn and recognize sequences of positions as outlined in the steps below.

1) Define the number S of states to use for describing a training trajectory and assign each time point to a random state.
2) Train the model of each state using the features of the time points assigned to it using the process in section III-A.
3) Estimate vector $\mathbf{P}$ of initial probabilities of each state and the matrix $\mathbf{A}$ of the probabilities of transitions between states. P_i is the probability that state i will be the first to appear, and $A_{i,j}$ is the probability that state i will be succeeded by state j. Their estimation is performed in a straightforward manner by noting the initial states and then counting the subsequent state transitions.
4) Use $\mathbf{A}$ and $\mathbf{P}$ for Viterbi decoding in conjunction with the state models to find the most likely state of each time point in the training data.
5) If most likely state assignments differ from the ones we had from before, go to step 2 and repeat. Otherwise terminate training and return $\mathbf{A}$, $\mathbf{P}$ and the state models.

Once we have obtained a set of state models and the initial and transition probabilities $\mathbf{P}$ and $\mathbf{A}$, we can use Viterbi decoding on an arbitrary input to estimate its similarity to the trained sequences, thus performing classification with respect to trained models.

IV. Synthetic Experiments

In this section we will present the results from synthetic simulations. We will examine two cases where we will learn and identify positions and trajectories. All examples were generated using a source image model of a two-dimensional square room [14]. The room size was $10m \times 10m$, we estimated the 24 most significant room reflections and to model the walls we used a sound absorption coefficient of 0.15. The sampling rate of the experiments was $44100Hz$. Two virtual cardioid microphones were placed in the room, the leftmost pointing towards the left side of the room, the rightmost towards the right side. Their magnitude response was $-4dB$ at $\pm 180^{\circ}$ and linearly scaled to $0dB$ at 0°.

To generate training examples we positioned the sound of a shaker— exhibiting a fairly wideband spectrum from $3500Hz$ to $13000Hz$—to the positions and trajectories we wanted to learn. To generate the testing examples we used the sound of a male speaker counting from one to five, and placed it in slightly different points as compared with the training positions. Had identical positions been used the classification results would be 100%; by introducing this slight deviation we construct a more realistic scenario and examine the tolerances of this approach.

A. Synthetic position example

To test the ability of this model to learn static positions we generated a pool of ten random positions, and then randomly offset them by up to $20cm$ to generate the testing positions. The two microphones were positioned at $(4.95m, 5m)$ and $(5.05m, 5m)$. The entire setup is

shown in figure 3(a). In order to perform feature extraction we used an FFT size of 1024 points with no overlap and no zero padding and employed a Blackman window. Each spectral frame was used to extract a set of features which were then used to train the relative magnitude and phase model. In the classification stage each set of features from each spectral frame was classified by assigning it to the position model for which it exhibited the highest likelihood. Table I presents the confusion table for the frame level classification. Each row shows how many frames from each test example were classified as belonging to each model (each column being a model). The same data is also displayed in histogram form in figure 4, where each subplot corresponds to a row in table I. The position tests are ordered in positions from 1 to 10, so the diagonal elements of the table should contain the higher numbers, and the ith bar in the ith position subplot should be the tallest one. Numbers off the diagonal of the table contain the misclassified frames.

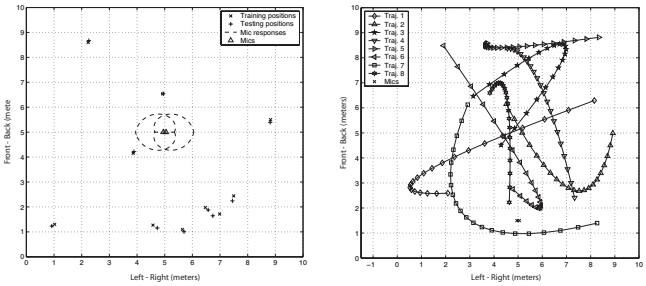
TABLE I

SYNTHETIC POSITION ESTIMATION CONFUSION TABLE

		Estimated Position								
	68	0	0	3	0	0	0	0	0	0
	0	65	6	0	0	0	0	0	0	0
	0	6	65	0	0	0	0	0	0	0
	2	0	0	69	0	0	0	0	0	0
	0	0	0	0	70	0	0	1	0	0
	0	0	0	0	0	68	0	3	0	0
	0	0	0	4	0	0	68	0	0	0
	0	0	1	0	0	0	0	70	0	0
	0	0	0	0	0	0	0	0	63	9
	0	0	0	0	0	0	0	0	0	71

Actual Position

Overall recognition at the frame level was 95.3%. The position model which claimed the most frames from each sound example was used to deduce the position that the sound was coming from, in which case accuracy was 100%. Repeated simulations yielded the same results so long as two positions did not exhibit the same relative phase and magnitude features. This would be the case when positions would be in the same angle of attack towards the sensors. This problem is easily resolved by using more microphones or by strategically positioning the two microphones and taking advantage of their directional responses (see section V-A). Additional simulations with up to 50 positions in the same virtual environment yielded results no worse than 90% recognition at the frame level (and 100% at the entire sound level).



(a) (b)

Fig. 3. On the left subfigure (a) are the training, testing and microphone positions used to synthetically evaluate static position learning. On the right subfigure (b) is the set of trajectories used for training to perform trajectory classification. The position of the microphones in the room is shown by the two $\times$ marks around position $(5m, 1.5m)$.

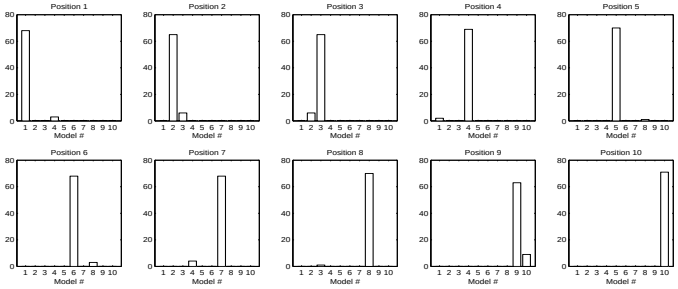


Fig. 4. Classification results for each position. Each plot is a histogram of the frame-level classification result of each position tested.

B. Synthetic trajectory example

In this example the training data consisted of a set of eight different trajectories (shown in figure 3(b)), and for testing data we generated eight more trajectories which randomly deviated from the training set by up to $20cm$ at each point. The microphones were positioned at $(4.95m, 1.5m)$ and $(5.05m, 1.5m)$. We performed training using the Viterbi algorithm described in section III-B and used six states to model the trajectories. The FFT frames were 1024 points. Figure 5 presents the results of this classification. Each plot displays the likelihood of each test trajectory as evaluated by all trained models. All trajectories are exhibiting a maximum likelihood at the appropriate model.

V. REAL-WORLD EXPERIMENTS

In this section we present results that we obtained using real-world recordings. The recordings were performed in an office measuring $3.80m \times 2.90m \times 2.60m$. The room features many reflective surfaces most important of which being two glass windows amounting to about $3m^2$, and a large whiteboard. The reverberation T_{60} of the room was estimated to be 0.45 sec.

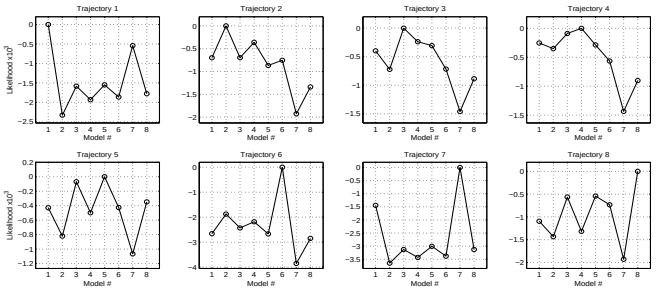


Fig. 5. Likelihoods of each test trajectory from each model.

Fig. 6. Recording apparatus used for the real recordings.

Background noise, such as air conditioning, and room ambience amounted to a $-12dB$ noise floor as compared to the speech levels used in the evaluation recordings. The recordings were made using a Technics RP-3280E "ambience microphone", which is a dummy head binaural recording device (figure 6). Its microphones were substituted by two Behringer ECM-8000 microphones. The head-like shape of the enclosure and the pinnae that are part of the sound path leading to each microphone ensure that sounds from almost all locations have distinct relative magnitude and phase values (this is the same feature that allows humans to localize sounds with little ambiguity in three dimensions using only two ears [15]).

Just as before we generated training examples by using the aforementioned shaker in various positions and trajectories, and then performed classification on male speech counting numbers. The sampling rate was $44100Hz$.

A. Position example

To test position recognition, approximately 3 sec training examples were generated using the shaker from eight uniform positions around the microphone at $0°$, $45°$, $90°$, $135°$, $180°$, $225°$, $270°$, and $315°$. To generate the testing data, one of the authors counted from one to six from approximately the same positions. Using analysis frames of 1024 samples we estimated the likelihoods of each frame and assigned it to the classifier which reported the highest likelihood. The results are reported

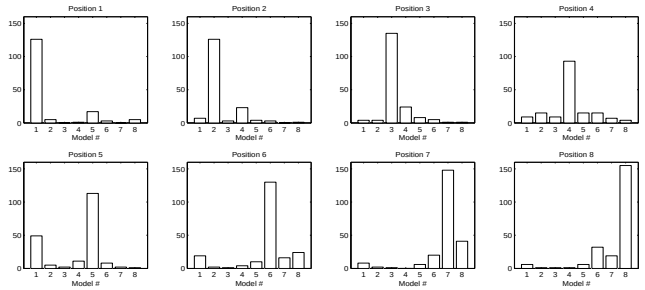


Fig. 7. Results of position classification using a dummy head microphone. Note how due to symmetry of the head the $0°$ and $180°$ positions are easily confused (just as in human hearing). Also note how the directional responses allow us to localize from $0°$ to $360°$ with only two elements.

as a confusion table shown in table II also displayed in figure 7.

The overall classification for each case is correct. An interesting observation is that classification for the fifth position, which corresponds to $180°$, is strongly confused with the first position which corresponds to $0°$. This confusion is a well documented effect in human localization known as the front/back ambiguity. This rises due to the fact that the relative magnitude and phase between two human ears are the same across the $0°$ to $180°$ meridian. Since our recording apparatus is modeled after the human head, it exhibits the same ambiguity which we find in our results. However the proper classification prevails since the room response (which was also implicitly learned) imposes slightly different responses in these two positions. Frame level classification is about 68% (predominantly due to front/back confusion), classification over an entire testing sound is 100%.

TABLE II
POSITION ESTIMATION CONFUSION TABLE

		Estimated Position							
		$0°$	$45°$	$90°$	$135°$	$180°$	$225°$	$270°$	$315°$
Actual Position	$0°$	126	5	0	1	17	3	0	5
	$45°$	7	126	3	23	4	3	0	1
	$90°$	4	4	135	24	8	5	1	1
	$135°$	9	15	9	93	15	15	7	4
	$180°$	49	5	2	11	113	8	2	1
	$225°$	19	2	1	4	10	130	16	24
	$270°$	8	2	1	0	6	20	148	41
	$315°$	6	1	1	1	6	32	19	155

B. Trajectory example

In this example the training data consisted of seven distinct trajectories within the recording room. The trajectories featured two passes across the long dimension of the room in each direction, two passes from each end to the center of the room and back, two passes across

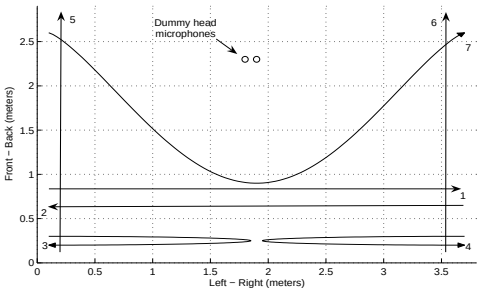


Fig. 8. The trajectories and microphone positions used for the real-world trajectory test. The numbers near the end arrows indicate the numerical designation of each trajectory. Trajectories 1 to 4 were along the same line on the front-back axis, but are shown slightly separated in the figure for better legibility.

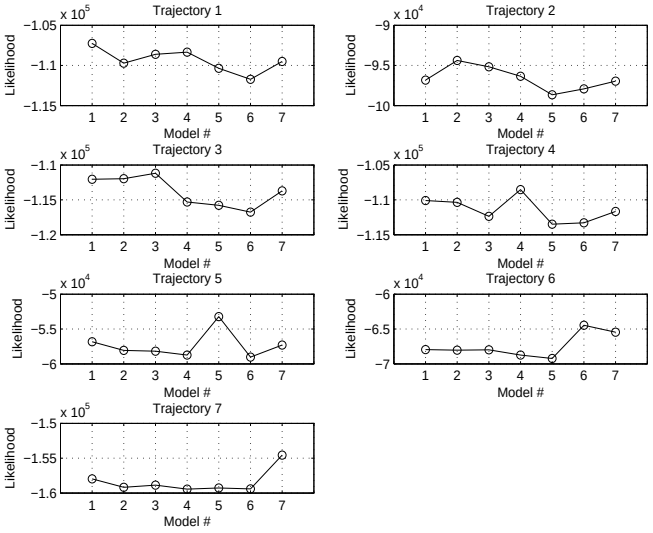


Fig. 9. Likelihoods of each test trajectory from each model.

the short dimension of the room in either side of the microphones and an arching trajectory starting from one side of the room to the other. The dummy head was placed in the center of the office close to one of the walls. Figure 8 graphically displays the trajectories and the microphone placement. As in the previous section the training examples were generated by an author producing sound with a shaker along these trajectories, and the testing examples were generated likewise with the author counting from one to twenty in English. Using the same feature settings and training as in the previous section we obtained the results shown in figure 9. The correct trajectory was classified for all cases, yielding 100% classification.

C. Training and testing environment mismatching

Using the trained models from the static positions experiment in section V-A, we also tested accuracy under different conditions to evaluate the limits of this approach.

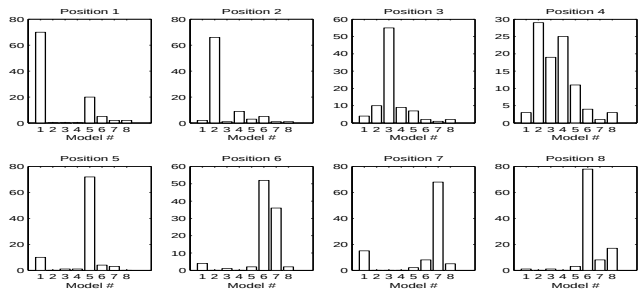


Fig. 10. Results of position classification using a dummy head microphone rotated 90° after training to simulate an acoustic environment change. Comparing the results with figure 7 we see that the margin of classification has worsened and that we have two misclassifications.

First we rotated the recording apparatus by 90° anti-clockwise, thereby changing the already learned room characteristics. Since the apparatus was located in the center of a non-square room the basic structure of the acoustic environment was severely changed. In addition to that, since one side of the office had a large whiteboard and the opposing side contained a cluttered desk and bookshelves, the reflection characteristics were now significantly different from the training case. In addition to the rotation, the blinds covering one of the windows were drawn to additionally change reflectance properties. The results are shown in figure 10. Most positions were properly classified, although by a notably less clear margin as compared to figure 7. The two misclassifications were off by 90°, with the correct answer being the second most likely model. The models for these positions apparently relied on the environment's acoustics more than the cross-microphone relationship. Less drastic changes in the room, such as moving furniture around, did not pose as serious a disruption proving some degree of robustness against changing conditions between training and testing.

We also placed the recording apparatus in a corner of the room. It was positioned to face towards the center of the room which resulted into strong reflections coming from the rear (especially from the rear-left, since one side of the corner was a glass window). Due to this placement we could only evaluate three positions, 0°, 45° and 315°. Individual frame classification results are shown in table III. Note how the frames from the two side positions resulted, by majority, in proper classification, but the front position was misclassified. In the case where the sound was coming from the front there are two factors that contributed to the misclassification. First we would expect to have strong reflections from the rear (and mostly the rear-left where the glass surface was), second there is the problem of front-rear confusion we

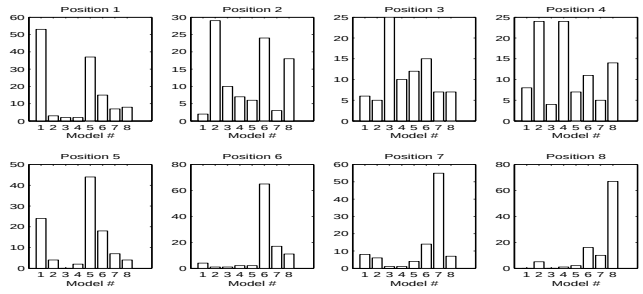


Fig. 11. Results of position classification using an unknown noise source. Although the margin of certainty is lower, the correct classification is made for all cases.

came across in section V-A. These are factors that are reflected in the results where we see most frames being classified as coming the the rear or the rear-left.

TABLE III
ROOM CORNER POSITION ESTIMATION CONFUSION TABLE

		Estimated Position							
		0°	45°	90°	135°	180°	225°	270°	315°
Actual Position	0°	7	7	14	16	29	5	4	5
	45°	1	64	25	1	5	1	1	3
	315°	3	0	0	1	0	19	73	12

Finally to test noise tolerance we evaluated classification under the same positioning as the trained data, but performed the speech recordings with music at $-8dB$ relative to the speech levels and coming from an untrained position. The results are shown in figure 11. We see the correct classification for all cases but at a tighter margin.

D. Use of additional array elements

In this section we consider the case where we have an array of more than two elements and see how that can change the localization results. For this setting we used a four microphone linear array with the microphones spaced $10cm$ apart. We trained localization models for four distinct positions. The training and testing methodology was the same as in the preceding sections. We evaluated the results using only two, three or all four microphones by extracting the appropriate features as described in section II. The results are shown in figure 12. The overall frame-level classification results were 81.5% correct for using two microphones, 89.5% for three microphones and 91.5% for all four microphones (taking the frame majority vote all testing sounds were properly classified in all cases). We note that each time we added an extra microphone we observed an improvement in classification accuracy, which was expected since the additional information in training helps disambiguate cases in which two microphones would be inadequate.

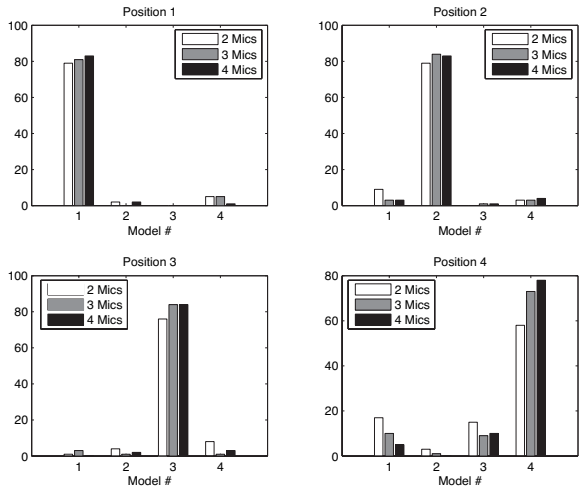


Fig. 12. Results of position classification using two, three or four microphones. The frame-based classification results are shown side by side for each position, white bars are for two microphones, grey bars are for three microphones, and black bars are for four microphones.

VI. CONCLUSIONS

In this paper we developed a statistical model which can model magnitude and phase responses and can be used to classify transfer functions. We have employed this model for the task of sound localization using microphone arrays. We have tested this model on both synthetic and real-world situations under a variety of settings and obtained satisfactory classification results.

This is a different paradigm from the one usually employed on arrays and it comes with its own set of advantages and disadvantages. The main differences are that there needs to be a training stage, that we recognize a discrete set of positions, and that the system is somewhat tied to the acoustical environment during training.

The existence of training complicates an installation by requiring that initial stage. However it frees the array designer from having to ensure meticulous setup and array uniformity that traditional approaches require. Since locations are learned from cross-element responses and not deduced from propagation hypotheses, there is no need to have a system that is well approximated by theory. The learning part also takes care of adverse acoustic conditions. This frees the array designer from reverberation considerations, since any acoustic environment peculiarities can be absorbed by the learning process. The only requirement is that each learned position exhibits a unique transfer function relating pairs of microphones. Although this is a difficult requirement to ensure, it is most often the case in reasonable acoustical settings. An added advantage to this feature is that otherwise ambiguous positions can now be discernible

due to unique reverberation patterns even though their direct path features are the same (unless of course the acoustic environment mirrors the response symmetry of the array).

The downside is that we might end up learning so much of the acoustic environment that an environmental change might require relearning. Like all learning-based methods, this is a highly context dependent issue. If the room response is a dominant element in discriminating locations, then this will be an issue and a change in the acoustic environment would be detrimental to performance. However, if this is the case, a more traditional localization approach would have failed before we even changed the environment, given that reverberation would provide more location information than the direct signal. In section V-C we explored the tolerance of training and deployment environment mismatch and noted that our approach would start to fail under severe mismatching. Minor environmental changes such as moving furniture, slightly displacing the array elements and adding noise, did not have a particularly adverse effect in classification.

Finally the fact that this is a system that is not recognizing a continuum of positions, but rather a discrete set also provides a level of robustness by eliminating certain ambiguities we often see in localization systems. If ambiguous positions are not simultaneously part of the training set, then there is no difficulty in recognizing them.

These differences place our approach not as a competitor to other localization approaches, but rather as an alternative. Depending on the limitations and requirements of an array deployment one approach can be better than the other. The solution we present is geared towards scenarios which require the surveillance of a specific set of locations/trajectories under adverse acoustical conditions and array morphology.

Although we only presented this in the context of localization, this work can be extended to model transfer functions in general and potentially be employed for other system modeling tasks where wrapping is an issue.

ACKNOWLEDGMENT

The authors would like to thank Bhiksha Raj of Mitsubishi Electric Research Laboratories for fruitful discussions, and the assigned reviewers for this paper who provided invaluable feedback in improving it.

REFERENCES

[1] C.H. Knapp and G.C. Carter, "The generalized correlation method for estimation of time delay", in *IEEE Transactions of Acoustics and Speech Signal Processing* ASSP-24, 320–327 1976.

[2] "Special issue on time-delay estimation", *IEEE Transactions on Acoustics and Speech Signal Processing*, vol. ASSP-29, June 1981.

[3] M.S. Brandstein, J.E. Adcock, and H.F. Silverman, "A practical time delay estimator for localizing speech sources with a microphone array", in *Computer Speech and Language*, vol. 9, pp. 153–169, Apr. 1995.

[4] Benesty, J. "Adaptive eigenvalue decomposition algorithm for passive acoustic source localization", in *Journal of the Acoustical Society of America*, vol. 107, pp. 384–391, Jan. 2000.

[5] Omologo, M. and Svaizer, P. Acoustic event localization using a cross power spectrum phase based technique, *in the proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1994.

[6] R. Schmidt, "A new approach to geometry of range difference location", in *IEEE Transactions of Aerospace and Electronic Systems*, vol. AES–8, pp. 821–835, Nov.1972.

[7] J. Smith and J. Abel, "Closed-form least-squares source location estimation from range-difference measurements", in *IEEE Transactions on Acoustics and Speech Signal Processing*, vol. ASSP-35, pp. 1661–1669, Dec. 1987.

[8] J. Smith and J. Abel, "The spherical interpolation method for closed-form passive source localization using range difference measurements", in *Proceedings of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 1987.

[9] T. Pham and B.M. Sadler, "Wideband array processing algorithms for acoustic tracking of ground vehicles". US Army Research Laboratory, report. Available at: http://www.arl.army.mil/sedd/acoustics/reports.htm

[10] S.T. Birchfield and D.K. Gillmor, "Fast bayesian acoustic localization", in *the proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2002

[11] G. Arslan, F.A. Sakarya, and B.L. Evans, "Speaker Localization for Far-field and Near-field Wideband Sources Using Neural Networks", *IEEE Workshop on Nonlinear Signal and Image Processing*, 1999.

[12] Dempster, A.P., N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", in *Journal of Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977

[13] Rabiner, L. R. "A tutorial on hidden Markov models and selected applications in speech recognition", Proceedings of the IEEE, 77(2):257-286.

[14] Borish, J. "Electronic Simulation of Auditorium Acoustics", Ph.D. thesis, Center for Computer Research in Music and Acoustics, Department of Music, Stanford University, CA, (1984).

[15] Begault D.R. "3-D Sound for Virtual Reality and Multimedia", Academic Press, 1994.