# A Distributed Approach to Passive Localization for Sensor Networks

**Rahul Biswas and Sebastian Thrun**
{rahul,thrun}@cs.stanford.edu
Stanford University

### Abstract

Sensors that know their location, from microphones to vibration sensors, can support a wider arena of applications than their location unaware counterparts. We offer a method for sensors to determine their own location relative to one another by using only exogenous sounds and the differences in the arrivals of these sounds at different sensors. We present a distributed and computationally efficient solution that offers accuracy on par with more active and computationally intense methods.

## Introduction

The decreasing cost and increasing popularity of a wide variety of computationally equipped sensing devices has led to the use of large numbers of these devices in aggregate to perform a variety of tasks. Typical deployments include networks of cameras for security, chemical sensors for farming, vibration sensors for geological study, RFID tags for inventory management, and microphones for tracking and surveillance. In most cases, the sensing devices have some computational ability and either wired or wireless network connectivity. The ability to communicate allows the sensors to share information and process it collectively as a so-called sensor network (J. Kahn & Pister 1999; Pottie & Kaiser 2000).

Regardless of the type of device, the location of the sensor, in either a global coordinate frame or in relation to the other sensors, is required for useful operation. For example, cameras tracking people moving through a building need to know their own location to infer the locations of the people. While it is possible to determine the locations of these sensors manually, this rapidly becomes infeasible as the number of sensors grows. Also, the sensors may not be easily or safely accessible to people, e.g. sensors being dropped in a forest to monitor an ongoing forest fire. It is nearly always preferable to use an automated localization method to determine the sensor locations.

The popularity of sensor networks and the importance of localizing them has led to a rich body of localization algorithms. Existing techniques tend to use the sensors themselves or an external device to actively generate typically an acoustic signal that can be commonly observed by each sensor. Such methods tend to be highly accurate but costly in both the type of hardware required and the effort required for deployment. Less expensive methods based on radio signals exist but offer poor results. We take a different approach. We use sounds but instead of generating them ourselves, we rely on exogenous events naturally occurring in the environment. This makes the localization problem harder because we neither know where nor when the events occurred.

We discuss herein a distributed algorithm to leverage the differences of the arrival times of the sounds at each observing sensor to recover relative locations of the sensors to one another. Recovering relative locations can be viewed as solving a non-linear least mean squares (LMS) problem but traditional LMS algorithms converge slowly and are susceptible to local minima. The key to solving the passive localization problem is to find small, overlapping so-called clusters of sensors whose relative location can be determined and then merge these partial solutions into a coherent whole. On both data from Crossbow MICA2 sensors and a sensor network simulator, our algorithm performs comparably to active methods in both robustness and accuracy. Moreover, the computational effort required in localization is distributed over the sensors and scales linearly with the number of sensors and events.

The rest of this paper is organized as follows. First, we discuss existing work in sensor localization. Second, we present a formal definition of passive localization. Third, we present our approach. Fourth, we discuss sensor hardware and present results of how our algorithm performs and how different factors affect these results. Fifth, we conclude with a review of the paper and possible future directions.

## Related Work

### Localization Algorithms

The most widely used localization system is the Global Positioning System (GPS) (B. Hoffman-Wellenhof & Collons 1992). It offers a relatively inexpensive solution for many applications but lacks the relative accuracy required for many applications and the ability to work indoors.

Many schemes use so-called beacon nodes that know their location with respect to a global coordinate frame (Bulusu 2002; N. Priyantha & Balakrishnan 2000; A. Ward & Hopper 1997). While these methods are robust and accurate, the cost and effort to set up the beacon nodes limits their usefulness.

Ranging solutions (Whitehouse 2002) are most similar to our work. In terms of our problem definition, they both collocate the sensors and events and assume knowledge of the time variables. These methods expend extra sensor energy, create unwanted disturbances, and limit the diameter of the sensor network to the audible range of the sensor. They also degrade rapidly with missing information. In a previous paper (Biswas & Thrun 2004), we identified the passive sensor localization problem but the solution presented here both scales better to large networks and is more adept at handling missing information.

Methods based on radio signals attempt to leverage both connectivity and received signal strength (RSS) (Bahl & Padmanabhan 2000a; 2000b) and offer an inexpensive solution but are neither robust nor accurate.

## Problem Definition

### Passive Sensor Localization

First, let the sensor network consist of $N$ sensors at locations $S = \{S_1 \ldots S_N\}$. Let $S_i^x$ refer to the x-coordinate of the location of sensor $i$ and let $S_i^y$ and $S_i^z$ refer to the y and z coordinates, respectively. Constraining $S_i^z$ to be 0 suffices to define a 2D version of this problem. Determining these locations constitutes the localization problem.

Second, let there be $M$ acoustic events at locations $E = \{E_1 \ldots E_M\}$. Let $E_i^x$, $E_i^y$, and $E_i^z$ refer to the x, y, and z coordinates, respectively. We do not know these locations.

Third, let the onset times of the events be $T = \{T_1 \ldots T_M\}$. These times are not known to us.

Fourth, let there exist $M \cdot N$ variables $R_n^m$ that specify the arrival time of the sound from event $E_m$ at sensor $S_n$. We observe $P$ of these variables, depending on which sensors actually heard each sound. We assume that sensors do not assign the same label to different events but it is permissible for them to assign different labels to the same event.

### Existence of a Unique Solution

Since no global reference frame exists, we impose the following arbitrary conditions to canonicalize our solution space. $S_1$ is set to be the origin of the coordinate system, i.e. (0,0,0). $S_2$ is set to lie on the positive direction of the y axis, i.e. (0,a,0) where a $> 0$. $S_3$ is set to lie on the xy plane in the positive x direction, i.e. (b,c,0) where b $> 0$. This form can easily provide a global reference if the locations of these three sensors is known with reference to a global frame.

It is possible that $R$ does not provide enough information to determine $S$. Sometimes, this lack of information can be determined a priori given $M$, $N$, and $P$. Each variable $S_n$ has three unknown components as does each variable $E_m$. Each variable $T_m$ introduces an additional unknown component. Each observed variable $R_n^m$ introduces one constraint. Moreover, when $M \geq 3$, the canonicalization of the solution eliminates four unknown variables.

Thus, only if:

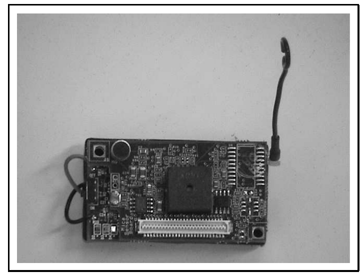

Figure 1: A Crossbow MICA2 Sensor

$$P \geq 3 \cdot M + 4 \cdot N - 4 \tag{1}$$

can there exist a unique solution. In situations where the inequality is strict, we have an overconstrained set of equations. Our algorithm can leverage this additional information for robustness and additional accuracy. Even when the inequality is satisfied, there exist degenerate situations where the information is not sufficient for unique localization. For example, the sensors may not hear enough common events or the true locations may exhibit excessive collinearity.

### Sensor Model

We assume that $R_n^m$ is distributed as follows:

$$R_n^m \quad = \quad N(T_m + \frac{d(S_n, E_m)}{s}, \sigma^2) \tag{2}$$

where $d(a, b)$ is the Euclidean distance between points $a$ and $b$, $s$ is the speed of sound in free space, $\sigma$ is the standard deviation of the error in recording arrival times, and $N(\mu, \sigma^2)$ is a Normal distribution with mean $\mu$ and standard deviation $\sigma$. We assume that the clocks on the sensors are synchronized and that the $R$ variables are thus directly comparable.

## Our Approach

Our algorithm operates in three stages. First, we partition the sensors into a set of overlapping clusters. Second, we localize each cluster individually. Third, we merge the localized clusters. We discuss each step in turn.

### Cluster Selection

In our algorithm, each sensor forms its own cluster with up to $k$ (typically 20) of its neighbors. The sensor $i$ greedily chooses other sensors according to the following criteria:

$$argmax_n \sum_{m=1}^{M} I(\bar{R_n^m} \wedge \bar{R_i^m}) \tag{3}$$

where $I(e)$ is 1 if $e$ is true and 0 otherwise, and $\bar{R_n^m}$ signifies that $R_n^m$ was observed.

This criterion has the effect of creating clusters where the participating sensors will tend to have heard the same sounds. Duplicate clusters are eliminated. To keep this time linear, we limit the search to sensors within radio range of one another.

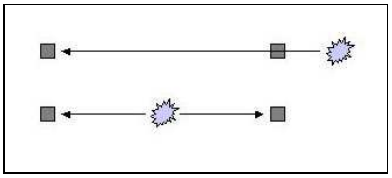

Figure 2: Different Interpretations of TDOA

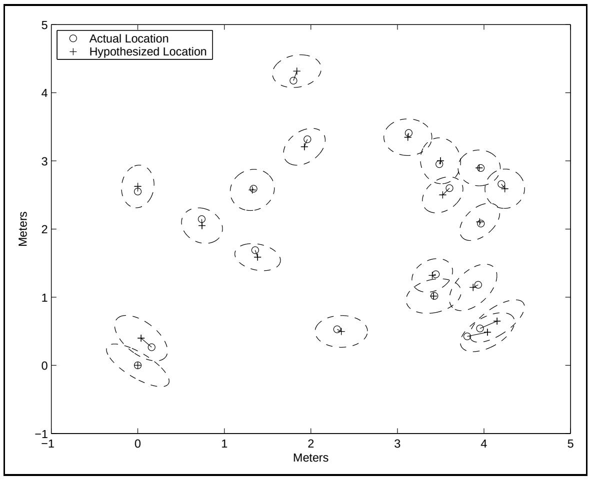


Figure 3: Uncertainty Estimates of Location

## Cluster Localization

**Estimating Distances between Sensors** In Figure 2, we see the two extreme cases for how differences in arrival times of a single sound at two sensors can be interpreted. In the top view, we see an acoustic event on the right (represented by a star) propagating to two sensors (represented by squares). The angle of arrival is the same for both sensors and the time difference of arrival is the propagation time from one sensor to the other. In the bottom view, we see the opposite case. The angle of arrival is diametrically opposite and the time difference of arrival is 0.

In general, we have:

$$0 \leq (R^m_{n1} - R^m_{n2}) \cdot s \leq d(S_{n1}, S_{n2}) \tag{4}$$

Moreover, note that:

$$\lim_{M \to \infty} \max_{m \in M} (R^m_{n1} - R^m_{n2}) \cdot s = d(S_{n1}, S_{n2}) \tag{5}$$

We thus approximate the distances between sensors and events:

$$\widehat{d(S_{n1}, S_{n2})} = \max_{m \in M} (R^m_{n1} - R^m_{n2}) \cdot s \tag{6}$$

This approximation works well in practice. This method only requires a constant number of sounds irrespective of the number of sensors involved.

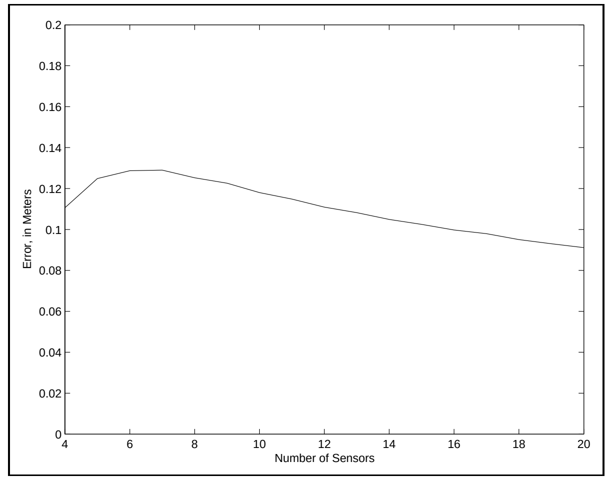


Figure 4: Localization Error for a Small Set of Sensors

**Recovering Locations from Approximate Distances** To recover locations from distances, it suffices to form an $N$ by $N$ matrix populated with these distances, take its eigendecomposition, and use the eigenvectors corresponding to the three largest (two, if we want a 2D projection) eigenvalues as the coordinates of the sensors.

**Recovering Sound Locations and Times** While the acoustic event locations are typically not important, it is possible to recover them if they are needed. For each event, we seek to minimize:

$$\sum_{n=1}^{N} |d(S_n, E_m) - (R^m_n - T_m) \cdot s| \tag{7}$$

where $E_m$ and $T_m$ are free variables. Gradient descent with the closest sensor and its observation time as a starting point suffices to solve this.

## Merging Clusters

To combine the results of individual clusters, we repeatedly merge the two clusters with the lowest inter-cluster cost. We define the cost between clusters $C_a$ and $C_b$ as:

$$\sum_{n=1}^{N} I(S_n \in C_a) \cdot I(S_n \in C_b) - q \cdot (|C_a| + |C_b|) \tag{8}$$

where $q$ is a penalty constant, and $|C_i|$ refers to the number of sensors in cluster $C_i$. This technique is similar to Kruskal's minimum spanning tree algorithm except that the edges ending at the newly joined clusters change in between steps of this algorithm. To keep this time constant, we limit the search to clusters stemming from sensors within radio range of one another.

To merge a pair of clusters, we optimally align (Besl & McKay 1992) a constant subset of sensors found in both clusters and use the resulting transformation to align the remaining sensors. It is necessary to attempt merging with

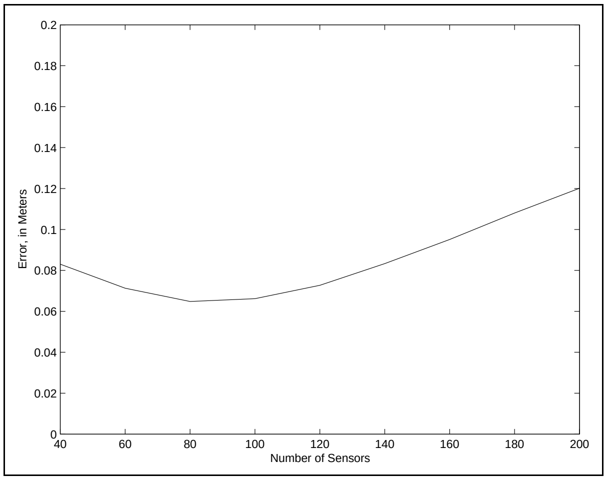


Figure 5: Localization Error for a Large Set of Sensors

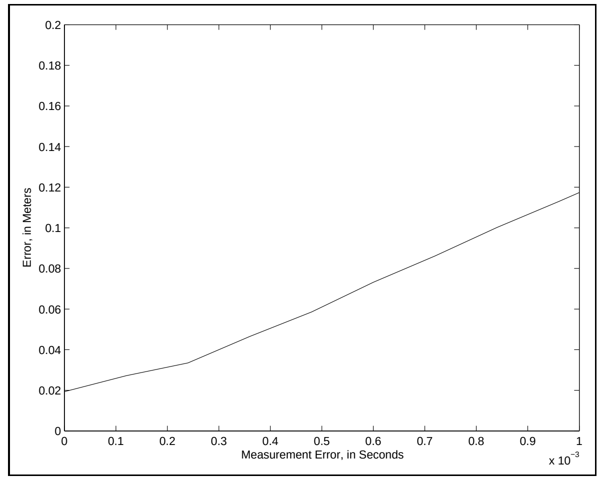


Figure 6: Localization Error as Measurement Error Grows

both the original cluster as well as the original cluster with the x-coordinates negated (as if the sensors were viewed through a mirror) to prevent merging failure due to lack of cluster canonicalization.

The total number of merges is $N - 1$. Computing the optimal alignment takes constant time and aligning one of the clusters takes time up to $N$ but typically much less. The average-case running time is $O(N \cdot lgN)$ for a reasonable merge order.

### Uncertainty Estimation

We estimate the uncertainty regarding the location of each sensor by fitting a multivariate Gaussian to the posterior probability of the sensor location given its recorded measurements and the acoustic event locations and onset times:

$$P(S_n|R_n^1 \ldots R_n^M, E_1 \ldots E_M, T_1 \ldots T_M)$$

$$= \prod_{m=1}^{M} P(S_n|R_n^m, E_m, T_m) \quad (9)$$

$$\propto \prod_{m=1}^{M} P(R_n^m|S_n, E_m, T_m) \quad (10)$$

$$= \prod_{m=1}^{M} N(T_m + \frac{d(S_n, E_m)}{s}, \sigma^2). \quad (11)$$

Figure 3 shows an example.

## Experimental Results

### Crossbow MICA2 Sensors

We use commercially available Crossbow MICA2 sensors (see Figure 1) for our experiments. These sensors are capable of detecting tones at 4 KHz, detecting light, measuring temperature, flashing colored lights, interfacing with other devices, and communicating wirelessly with one another. We discuss our experience with time synchronization, the acoustic range and consistency, and the accuracy in recording times.

Our algorithm relies on somewhat precisely synchronized clocks to be able to calculate accurate differences in arrival time. Only local accuracy is important, i.e. the sensors need to be able to compare times amongst themselves, not with a global reference such as an atomic clock. We use the Reference Broadcast System algorithm presented in (Elson, Girod, & Estrin 2000). It does a more than adequate job of providing the accuracy we need to localize our sensors with differences resulting from sound wave propagation.

The acoustic range of these sensors is limited and varies dramatically based on angle of arrival, remaining battery power, and the type of sound. Also, the sensors do not hear all sounds even when generated with a tone generator at the correct frequency. A model assuming that sensors hear nearby sounds and not faraway ones is not reasonable for these sensors and we do not assume one.

Lastly, we have empirically determined that the standard deviation in arrival time is approximately 0.6 milliseconds, in which time a sound wave would travel 20 centimeters.

### Sensor Network Simulator

We also tested our algorithm with a sensor network simulator that strives to replicate the behavior of the Crossbow sensors. It places both sensors and sounds uniformly at random within a square. The size of the square is varied to maintain a sensor density of one sensor per square meter. The sound density is fixed at three sounds per square meter. The acoustic range of the sensors is set to 10 meters and the standard deviation in recording time is set to 0.6 milliseconds.

### Performance

Figure 4 shows the performance of the algorithm as the number of sensors varies from 4 to 20. The error is fairly constant at 10 cm but goes down slightly as the number of sensors increases. We believe this occurs because the system

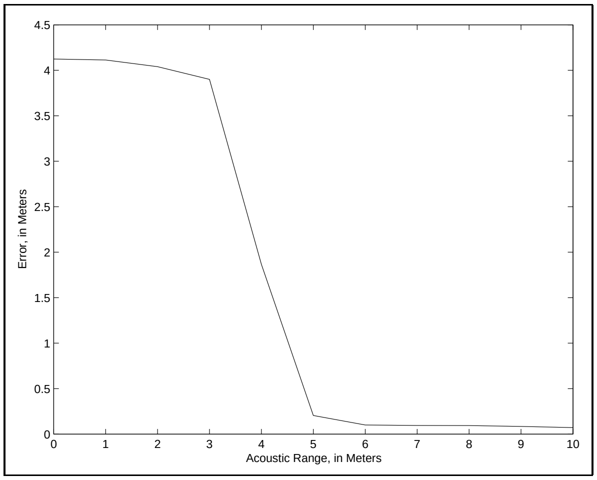


Figure 7: Localization Error as Acoustic Range Increases

of equations becomes more constrained. The uptick at the beginning is an artifact of canonicalization.

Figure 5 shows the performance of the algorithm as the number of sensors varies from 40 to 200. The average error remains at 10 cm and the error grows linearly with the number of sensors albeit at a very slow pace. This increase is due to the limited audible range of the sensors. The difficulty in capping global error is akin to the open loop mapping problem in robotics and has defied solution there as well.

Figure 6 shows the performance of the algorithm on 120 sensors as the measurement error grows. At the empirically determined measurement error for Crossbow MICA2 sensors, the error is 7 cm and grows linearly with $\sigma$, the standard deviation of the measurement times. It is interesting to note that the end of the graph, at 1 ms, corresponds to 34 cm of measurement error, but only 12 cm of localization error. This suggests that the algorithm is highly robust in the face of measurement noise due to the overwhelmingly overconstrained nature of the problem.

Figure 7 shows the performance of the distributed localization algorithm on 120 sensors as the acoustic range of the sensors grows. The error is very large until about 5 meters, at which point the sensors hear enough sounds to be able to localize well. If the density of sensors increased, the dropoff would occur at a lower range.

Figure 8 shows a diagram of 100 sensors that have been localized. This diagram shows an interesting point that Figure 5 does not. As the number of sensors increases and they are distributed over a wider range, the relative localization error grows sub-linearly. Also, the local error is superior to the global error, which is useful for most sensor network applications.

Figure 9 shows the empirically measured time required to localize a network as the number of sensors varies. While the time increases linearly and has a low constant factor, running this algorithm as-is on a sensor network would require only constant time as the bulk of the time is spent in localizing each cluster and each sensor would be responsible for localizing its own cluster in parallel with the other sensors.

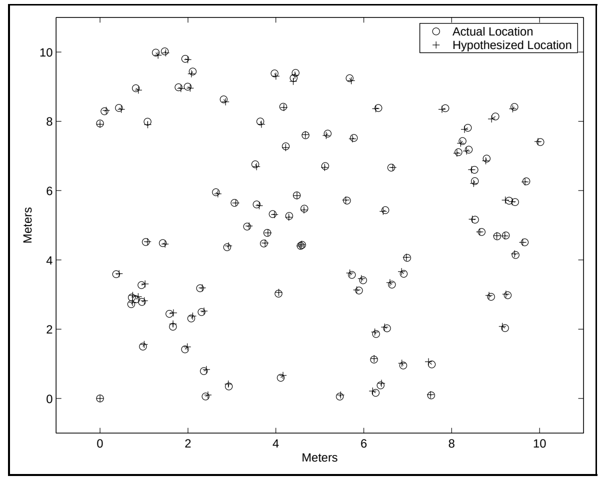


Figure 8: One Hundred Localized Sensors

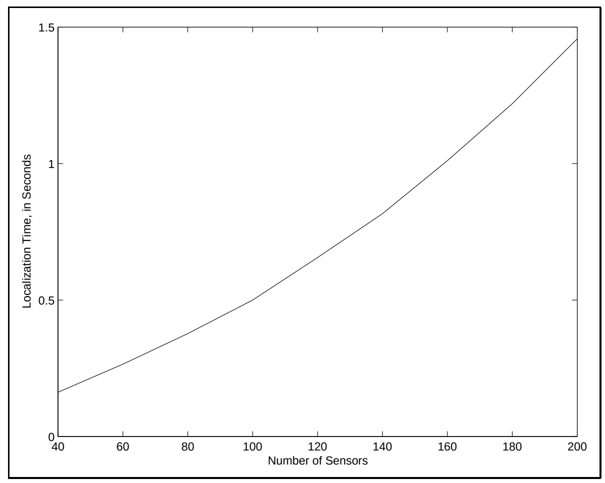


Figure 9: Localization Time as Number of Sensors Increases

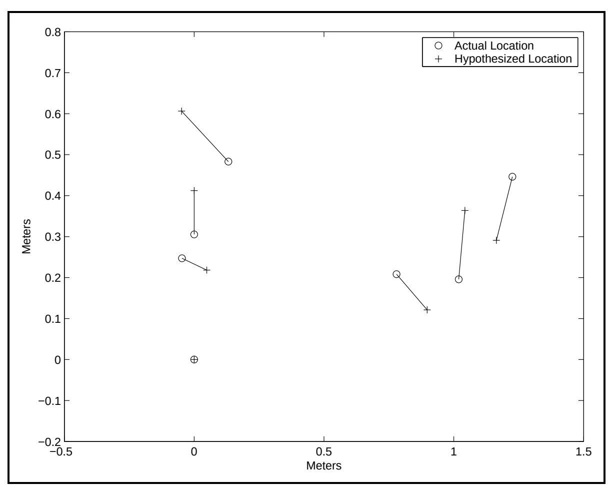


Figure 10: Localization Results for Crossbow MICA2 Sensors

Figure 10 shows localization results for seven Crossbow MICA2 sensors. The sensors were placed on a large sheet of graph paper, from which their actual positions were taken. This dataset unfortunately has only eight sounds, an unrealistically low number that leads to a poor distance approximation.

### Other Approaches

We tried other approaches as well but they did not fare as well. We discuss three in particular – an approximate constraint satisfaction problem (CSP) formulation, belief propagation (BP) (Pearl 1988), and other eigendecomposition techniques. For the CSP, we used a lower limit on the probability of the measurements given a partial, potential solution to curtail search. While this method is reasonably effective, it is slow and scales poorly.

For BP, a proper junction tree would involve all the variables in a single clique and inference would reduce to an exponential search for a likely solution. We chose a pairwise Markov Network approximate junction tree instead but inference on this network closely resembled a sort of stochastic hill climbing trying to optimize a poor solution via local changes. Even when applied to the solution our algorithm produced, BP was unable to improve the solution further because of the poor correlation of the posterior probability of the data with the actual localization error, irrespective of norm. We also tried Expectation Propagation (Minka 2001) with both Gaussian and mixtures of Gaussians approximating distributions but that behaved similarly.

More complicated eigendecompositions involving sounds sometimes offered higher accuracy but since the distance between a sound and a sensor is sensitive to the sound onset time, these alternative methods turned out to be somewhat fragile.

## Discussion

We have presented herein a passive localization algorithm that relies only on inexpensive microphones and exogenous sounds. Experimental results are provided for both simulated sensor networks and Crossbow MICA2 sensors. The results presented are robust and offer accuracy on par with far more expensive, active methods.

We are excited about our results and there are a few important directions in which we plan to extend this work. First, we are setting up a camera array with built-in microphones to evaluate our algorithm on a large array of real sensors. The Crossbow MICA2 platform was not designed for and does a poor job at detecting and time stamping the onset of sounds. An inexpensive USB microphone does a vastly superior job.

Second, our algorithm lacks a means to uniquely identify sounds received at multiple sensors. Giving the same label to two different sounds would lead to a highly inaccurate distance estimate. While it is difficult to detect mislabeled sounds a priori, the mistake should be evident following localization. We feel that Expectation Maximization with hidden correspondence variables may provide a useful and mathematically sound solution to this problem.

Third, we want to include both frequencies and amplitude of sound in our analysis. The difference in amplitude does not provide a distance estimate between sensors but does provide a constraint on it. The frequency is helpful for uniquely identifying sound and for leveraging a moving sound source.

## References


A. Ward, A. J., and Hopper, A. 1997. A new location technique for the active office. In *Proc. IEEE Personal Communications Magazine*.

B. Hoffman-Wellenhof, H. L., and Collons, J. 1992. *Global positioning system: Theory and practice, 2nd ed.* Springer-Verlag.

Bahl, P., and Padmanabhan, V. 2000a. Enhancements to the radar user location and tracking system. In *Technical Report MSR-TR-2000-12, Microsoft Research, Redmond, WA*.

Bahl, P., and Padmanabhan, V. 2000b. Radar: An in-building user location and tracking system. In *Proceedings of the IEEE Infocom*.

Besl, P., and McKay, N. 1992. A method for registration of 3-d shapes. In *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Biswas, R., and Thrun, S. 2004. A passive approach to sensor network localization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Bulusu, N. 2002. *Self-Configuring Localization Systems*. Ph.D. Dissertation, Univeristy of California, Los Angeles, Los Angeles, CA.

Elson, J.; Girod, L.; and Estrin, D. 2000. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of Operating Systems Design and Implementation*.

J. Kahn, R. K., and Pister, K. 1999. Mobile networking for smart dust. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*.

Minka, T. 2001. *A family of algorithms for approximate Bayesian inference*. Ph.D. Dissertation, MIT Media Lab, Cambridge, MA.

N. Priyantha, A. C., and Balakrishnan, H. 2000. The cricket location support system. In *Proceedings of ACM MobiCom*.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Mateo, CA: Morgan Kaufmann Publishers.

Pottie, G., and Kaiser, W. 2000. Wireless integrated network sensors. In *Proceedings of the ACM Communications*.

Whitehouse, K. 2002. The design of calamari: an ad-hoc localization system for sensor networks.