Experimental Results for and Theoretical Analysis of a Self-Organizing Global Coordinate System for Ad Hoc Sensor Networks

JONATHAN BACHRACH, RADHIKA NAGPAL, MICHAEL SALIB and HOWARD SHROBE
{jrb;radhi;msalib;hes}@ai.mit.edu
Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Abstract. We present an algorithm for achieving robust and reasonably accurate localization in a randomly placed wireless sensor network, without the use of global control, globally-accessible beacon signals, or accurate estimates of inter-sensor distances. We present theoretical analysis, simulation results and recent experimental results. The theoretical analysis shows that there is a critical minimum average neighborhood size of 15 for good accuracy, and simulation results show that position accuracy to within 20% of the local radio range can be achieved, even with upto 10% variation in the radio ranges.

Keywords: sensor networks, localization, tracking, acoustic ranging

Introduction

Advances in technology have made it possible to build ad hoc sensor networks using inexpensive nodes consisting of a low power processor, a modest amount of memory, a wireless network transceiver and a sensor board; a typical node is comparable in size to 2 AA batteries [Hill et al., 5]. Many novel applications are emerging: habitat monitoring, smart building reporting failures, target tracking, etc. In these applications it is necessary to accurately orient the nodes with respect to the global coordinate system. Ad hoc sensor networks present novel tradeoffs in system design. On the one hand, the low cost of the nodes facilitates massive scale and highly parallel computation. On the other hand, each node is likely to have limited power, limited reliability, and only local communication with a modest number of neighbors. The application context and massive scale make it unrealistic to rely on careful placement or uniform arrangement of sensors. Rather than use globally accessible beacons or expensive GPS to localize each sensor, we would like the sensors to be able to self-organize a coordinate system.

In this paper, we present an algorithm that exploits the characteristics of ad hoc wireless sensor networks to discover position information even when the elements have literally been sprinkled over the terrain. The algorithm exploits two principles:

(1) the communication hops between two sensors can give us an easily obtainable and reasonably accurate distance estimate, and

(2) by using imperfect distance estimates from many sources we can minimize position error.

Both of these steps can easily be computed locally by a sensor, without assuming sophisticated radio capabilities. We can theoretically bound the error in the distance estimates, allowing us to predict the localization accuracy. The resulting coordinate system automatically adapts to failures and the addition of sensors. In addition, the algorithm can easily incorporate coarse grain inter-sensor distance estimates if available. We also present recent results on applying this algorithm to outdoor localization using Berkeley MICA motes. Our experiments suggest that the algorithm works well in practice. Although described for sensor networks, this algorithm can be applied for localization in many contexts such as smart materials, smart dust, etc.

There are many different localization systems that depend on having direct distance estimates to globally accessible beacons such as the Global Positioning System [Hofmann-Wellenhoff et al., 6], indoor localization [Bahl and Padmanabhan, 1; Priyantha et al., 15], and cell phone location determination [Corporation, 3]. Recently there has been some research in localization in the context of wireless sensor networks where globally accessible beacons are not available. Doherty et al. [4] present a technique based on constraint satisfaction using inter-sensor distance estimates (and a percentage of known sensor positions). This method critically depends on the availability of inter-sensor distance measurements and requires expensive centralized computation. Savvides et al. [16] describe a distributed localization algorithm that recursively infers the positions of sensors with unknown position from the current set of sensors with known positions, using inter-sensor distance estimates. However, theoretical analysis of how the error accumulates with each inference and what parameters affect the error is extremely difficult. The algorithm we present does not rely on inter-sensor distance estimates, is fully distributed, and we can theoretically characterize how the density of the sensors affects the error. Our algorithm is based on a simpler method introduced by Nagpal [12] but also independently suggested in [McLurkin, 10]. Since then several variations of this algorithm have been proposed [Niculescu and Nath, 13; Whitehouse, 17]. The analysis presented here applies to all of these systems and can help theoretically predict achievable accuracy.

Section 1 presents the algorithm for organizing the global coordinate system from local information. We present a theoretical analysis of the accuracy of the coordinate system along with simulation results is presented in section 2. Section 3 reports simulation results that generalize the basic algorithm to include more accurate distance information based on signal strength. Section 4 investigates the robustness of the algorithm to variations in communication radius as well as sensor failures. Finally, section 5 describes experimental results of the algorithm in conjunction with acoustic ranging.

1.   Coordinate system formation algorithm

In this section we describe an algorithm for organizing a global coordinate system from local information. Our model of an ad hoc sensor network is randomly distributed sensors on a two-dimensional plane. Sensors do not have global knowledge of the topology or their physical location. Each sensor communicates with physically nearby sensors

within a fixed distance r, where r is much smaller than the dimensions of the plane. All sensors within the distance r of a sensor are called its communication neighborhood. In the first pass we assume that all sensors have the same communication radius and that signal strength is not used to determine relative position of neighbors within a neighborhood. Later in sections 3 and 4 we relax both of these constraints. We also assume that some set of sensors are "seed" sensors – they are identical to other sensors in capabilities, except that they are preprogrammed with their global position. This may be either through GPS or manual programming of position. The main point is for the seeds to be similar in cost to the sensors, and for it to be easy to add and discard seeds.

The algorithm is based on the fact that the position of a point on a two-dimensional plane can be uniquely described by its distance from at least three noncollinear reference points. The basic algorithm consists of two parts:

(1) each seed produces a locally propagating *gradient* that allows other sensors to estimate their distance from the seed, and

(2) each sensor uses a *multilateration* procedure to combine the distance estimates from all the seeds to produce its own position.

The following subsections describe both parts of the algorithm in more detail.

1.1. Gradient algorithm

A seed sensor initiates a gradient by sending its neighbors a message with its location and a count set to one. Each recipient remembers the value of the count and forwards the message to its neighbors with the count incremented by one. Hence a wave of messages propagates outwards from the seed. Each sensor maintains the minimum counter value received and ignores messages containing larger values, which prevents the wave from traveling backwards. If two sensors can communicate with each other directly (i.e. without forwarding the message through other sensors) then they are considered to be within one communication hop of each other. The minimum hop count value h_i that a sensor i maintains will eventually be the length of the shortest path to the seed in communication hops. Hence a gradient is essentially a breadth-first-search tree [Lynch, 8].

In our ad hoc sensor network, a communication hop has a maximum physical distance of r associated with it. This implies that a sensor i is at most distance $h_i r$ from the seed. However as the average density of sensors increases, sensors with the same hop count tend to form concentric circular rings, of width approximately r, around the seed sensor (figure 1). At these densities the hop count gives a good estimate of the straight line distance which is then improved by sensors computing a local average of their neighbors' hop counts.

1.2. Multilateration algorithm

After receiving at least three gradient values, sensors combine the distances from the seeds to estimate their position relative to the positions of the seed sensors. In particular, each sensor estimates its coordinates by finding coordinates that minimize the total
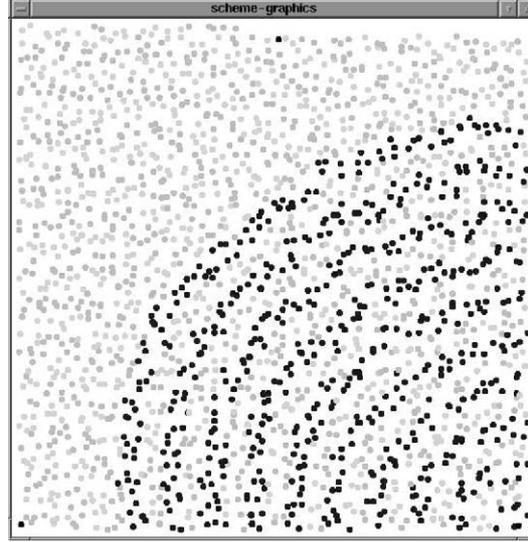
Figure 1. Gradients propagating from a seed. Each dot represents a sensor. Sensors are colored based on their gradient value.

squared error between calculated distances and estimated distances. Sensor j's calculated distance to seed i is

$$d_{ji} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1}$$

and sensor j's total error is

$$E_j = \sum_{i=1}^{n} \left(d_{ji} - \hat{d}_{ji} \right)^2, \tag{2}$$

where n is the number of seed sensors and $\hat{d}_{ji}$ is the estimated distance computed through gradient propagation. The coordinates that minimize least squared error can be found iteratively using gradient descent. More precisely, the coordinate estimate starts with the last estimate if it is available and otherwise with the location of the seed with the minimum estimated distance. The coordinates are then incrementally updated in proportion to the gradient of the total error with respect to that coordinate. The partial derivatives are

$$\frac{\partial E_j}{\partial x_j} = \sum_{i=1}^{n} (x_j - x_i)\left(1 - \frac{d_{ji}}{\hat{d}_{ji}}\right) \quad \text{and} \quad \frac{\partial E_j}{\partial y_j} = \sum_{i=1}^{n} (y_j - y_i)\left(1 - \frac{d_{ji}}{\hat{d}_{ji}}\right), \tag{3}$$

and incremental coordinate updates are

$$\Delta x_j = -\alpha \frac{\partial E_j}{\partial x_j} \quad \text{and} \quad \Delta y_j = -\alpha \frac{\partial E_j}{\partial y_j}, \tag{4}$$

where $0 < \alpha \ll 1$.

1.3. Algorithm properties

This simple algorithm has many advantages. It does not rely on global clocking or globally unique identifiers. The algorithm can easily adapt to the addition of sensors and addition of seeds. It can also adapt to the death of sensors and seeds, provided that sensors can locally monitor their neighbors. As a result it is a very attractive and natural algorithm to use in this setting, and many similar algorithms have been proposed [McLurkin, 10; Butera, 2; Niculescu and Nath, 13; Whitehouse, 17]. One of the key questions however is how good an accuracy can one expect from this algorithm, and in what ways can the quality be improved without losing its desirable properties.

2.   Analysis

In this section we analyze the accuracy of the coordinate system produced by this algorithm. In particular we are interested in the effect of the random distribution of sensors and the average local neighborhood size on the accuracy of the position estimates. We present both theoretical and simulation results.

Accuracy is measured by computing the average absolute error (distance) between the actual physical location and the logical position. The error comes from two sources: (1) errors in the distance estimates produced by gradients and (2) errors produced by combining the distance estimates using multilateration.

For the purpose of analysis, the sensors are assumed to be distributed independently and randomly on a unit square plane. This means that for each sensor we choose a random x coordinate and random y coordinate on the unit square, independently of all other sensors. The probability that there are k sensors in a given area a can be described by a Poisson distribution [Mendenhall et al., 11].

$$Pr(\text{k sensors in area a}) = \frac{(\rho a)^k}{k!}\, e^{-\rho a}.$$

From this formula, we can derive the expected number of sensors in area a to be ρa. ρ is equal to N/S, where N is the total number of sensors and S is the total surface area. The value that we are interested in is the expected number of sensors in a local neighborhood, which we will call n_{local}. A sensor communicates with all other sensors within the communication radius r. Thus the expected local neighborhood n_{local} is $\rho\pi r^2$. This random distribution represents a worst case analysis, where sensors may overlap arbitrarily. In a real system, sensors would probably not arbitrarily overlap, which would reduce the variance in local neighborhood sizes.

2.1. Error in distance estimate

The first source of error in distance estimate arises from the discrete distribution of sensors. A gradient computes the shortest communication path from the source to any sensor. Let the gradient value of sensor i be h_i, then the distance between sensor i and the

source is at least $h_i \times r$. In the ideal case the gradient value is equal to the straight-line distance, which would imply that with each communication hop one moved a distance r closer to the source. However given any two sensors, there may not be enough intermediate nodes for the shortest communication path to lie along the straight-line path between the source and destination. In that case, the gradient value overestimates the actual distance between the sensor and the source. Intuitively this is related to the density of sensors within a local neighborhood.

We can compute a *straightness factor* to compensate for the overestimation, using results derived in the context of random plane graphs and packet radio networks. In these models, receivers are spatially distributed (usually randomly) and each receiver communicates via broadcast with all neighbors within a fixed radius. Shivendra et al. showed that the theoretical expected local neighborhood n_{local} to ensure connectedness is between 2.195 and 10.526 and simulation experiments suggest at least 5 [Philips et al., 14]. Silvester and Kleinrock proved that $n_{\text{local}} = 6$ produces optimal network throughput for randomly distributed receivers [Kleinrock and Silvester, 7]. In the process they derived a formula for how the expected distance covered in one communication hop is affected by the parameters of the random distribution. The expected distance covered per communication hop, d_{hop}, is the physical distance between a pair of sensors divided by the expected number of hops in the shortest communication path. The straightness factor is d_{hop}/r; in the ideal case the straightness factor is 1. Kleinrock and Silvester [7] showed that d_{hop} depends only on the expected local neighborhood n_{local}, not the total number of sensors:[1]

$$d_{\text{hop}} = r \left(1 + e^{-n_{\text{local}}} - \int_{-1}^{1} e^{-n_{\text{local}}/\pi (\arccos t - t \sqrt{1-t^2})} \, dt \right). \tag{5}$$

Using this result, we can characterize the effect of density on the error. In figure 2, we numerically compute and plot d_{hop} for different n_{local} using this formula. From this graph we can see that when the expected number of local neighbors is small, the distance covered per communication hop is small and the percentage of disconnected sensors is large. But as the expected local neighborhood increases, the probability of nodes along the straight-line path increases rapidly until $n_{\text{local}} = 15$, when further increases in local sensor density has diminishing returns. Hence the analysis suggests n_{local} of 15 to be a critical threshold for achieving low errors in the distance estimates.

In figure 2, we also show the measured value of the average distance covered per hop for different n_{local}, averaged over several simulations of a gradient from a random source. The result confirms that the average distance covered per hop does vary as predicted by Kleinrock and Silvester. The formula slightly under-predicts d_{hop} due to an approximation made in the proof when the source and destination are close. Also,

---

[1] Since n_{local} is proportional to N/S, where N is the total number of sensors, it would seem odd to say that the formula does not depend on the total number of sensors. However if n_{local} is kept constant and N is increased (which implies the total area S must increase), then N has no effect. Hence it is appropriate to say that d_{hop} depends on only n_{local}.

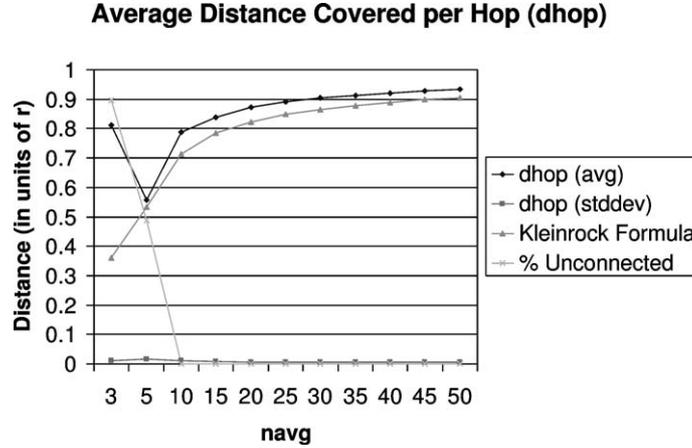**Average Distance Covered per Hop (dhop)**

Figure 2. Theoretical and experimental values for the average distance covered in one communication hop d_{hop}, for different expected local neighborhoods n_{local}. There is significant improvement below $n_{\text{local}} = 15$, after which increasing the neighborhood size has diminishing returns.

the simulation results suggest n_{local} of at least 10 is necessary to significantly reduce the probability of isolated sensors.

2.1.1. Improving the distance estimate through smoothing

Even in the ideal case of infinite density, the distance estimates produced are still integral multiples of the communication radius r. This low resolution adds an average error of approximately $0.5\,r$ to the distance estimates. Therefore we expect the error to asymptote around $0.5\,r$.

The gradient distance estimate is improved by computing a local average. Each sensor collects its neighboring gradient values and computes an average of itself and neighbor values.

$$s_i = \frac{\sum_{j \in nbrs(i)} h_j + h_i}{|nbrs(i)| + 1} - 0.5,\tag{6}$$

where h_i is the gradient value at sensor i (in other words, the integral distance estimate in units of r). $nbrs(i)$ are all the sensors within the communication radius r of sensor i.

Intuitively, sensors can determine if they are on the edge of the band by noticing that a large fraction of their neighbors have an integral distance estimate one lower or one higher than their own. The larger the fraction, the closer they are to the edge. The formula is derived from the effect of smoothing a gradient on a linear array of evenly spaced sensors where it produces the perfect distance (formal derivation in [Nagpal, 12]). However in our model the sensors are not evenly spaced and there are variations in density even within a neighborhood. The variations in density are the main source of error in the smoothing process.

2.1.2. Simulation results on distance error

Figure 3 shows results from simulation experiments that calculate the average absolute error in the integral distance estimates for different values of n_{local}. To vary n_{local}, the total number of sensors N is changed while keeping S and r constant. This keeps the physical diameter of the network (in units of r) constant across all simulations, so that all experiments are equally affected by any errors correlated with distance. In each simulation a gradient is produced by a randomly chosen sensor in the lower left corner. The data point for each value of n_{local} is averaged over 10 simulations. The absolute error for a sensor i is computed as $error_i = h_i d_{hop} - d_i$, where h_i is the gradient value, d_i is the Euclidean distance between sensor i and the source, and d_{hop} is the expected distance covered per hop calculated using formula (5). This takes into account the fact that d_{hop} represents the expected distance traveled in one hop for a given sensor density.

The results confirm our earlier analysis. As the value of n_{local} increases the accuracy of the distance estimate improves, with both the average and standard deviations in error decreasing dramatically. However past $n_{local} = 15$ the error before smoothing asymptotes at $0.4 r$ due to the limited resolution. Further analysis of these simulations shows that the error does not increase significantly with distance from the source because the majority of the per hop error is removed by using Kleinrock and Silvester's formula (5). The error is also not correlated with orientation about the source which is an interesting side-effect of choosing a random distribution versus a rectangular or hexagonal grid where there is anisotropy.

For each of the experiments done for integral gradient values, we also calculated the error in the smoothed gradient value for each sensor. The average error results are also plotted in the same figure. The simulation experiments show that for $n_{local} > 15$ smoothing significantly reduces the average error in the gradient value. Before that the error is dominated by the integral distance error. At $n_{local} = 40$ the average error is as
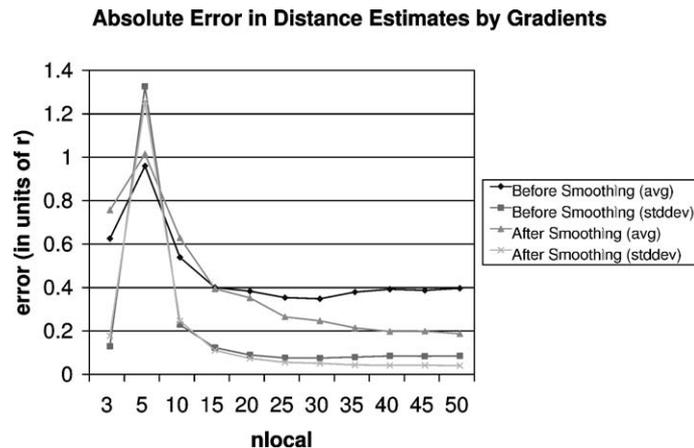

Figure 3. Average error in gradient distance estimates for different n_{local}. Significant improvements are seen in the integral distance estimates for $n_{local} < 15$. Beyond 15 there is improvement when the distance estimates are smoothed.

low as $0.2\,r$. However the error is never reduced to zero due to the uneven distribution of sensors.

2.2. Accuracy of multilateration

The distance estimates from each of the seeds has a small expected error. We combine these distance estimates by minimizing the squared error from each of the seeds using a multilateration formula. Multilateration is a well-studied technique that computes the maximum likelihood position estimation. We use gradient descent to compute the multilateration incrementally.

However, the seed placement has a significant effect on the amount of error in the position of a sensor. As explained in section 2.1, we can treat the error in the distance estimate from a single seed as radially symmetric and invariant with distance. However, when the distances from multiple seeds are combined, the error varies depending on the position of the sensor relative to the seeds. In figure 4 the concentric bands around each seed represents the uncertainty of the distance estimate from that seed; the width of the band is the expected error in the distance estimate. The intersection region of the two bands represents the region within which a sensor "may" exist – the larger the region, the larger the uncertainty in the position of the sensor. Hence the error in position of a sensor depends not only on the error in the distance estimates, but also in the position of the sensor relative to the two sources. Let ϵ be the expected error in the distance estimates from a seed, and θ be the angle $\angle$ASB. The overlap region between two bands can be approximated as a parallelogram.

Theorem 1. The expected error in the position of a sensor S relative to two point sources A and B is determined by the area of the parallelogram with perpendiculars of length 2ϵ and internal angle θ. The area is $(2\epsilon)^2 / \sin\theta$.
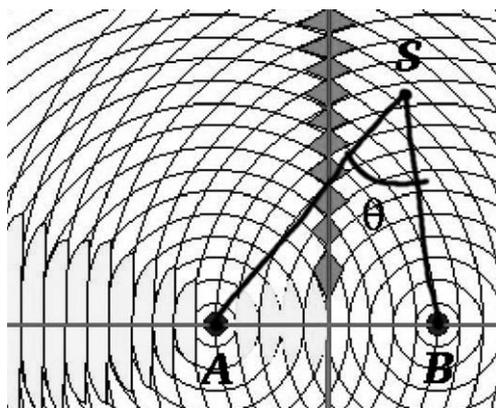

Figure 4. Error in position relative to two seeds can be approximated as a parallelogram. The area of this parallelogram depends on the angle θ. When θ is 90 degrees the error is minimized, however in certain regions θ is very small resulting in very large error.

The area of the parallelogram is minimized when θ is 90 degrees (square) and when θ is very large or very small the bands appear to be parallel to each other resulting in very large overlaps and hence large uncertainty.

This theoretical result has several consequences. First, increasing the accuracy of distance estimates decreases the width of a band, but can not eliminate the effect of seed placement. Second, by adding more seeds, the areas of uncertainty will decrease because there will be more bands intersecting. If placed correctly the intersecting regions can be kept small in all regions. This analysis suggests first placing seeds along the perimeter to avoid the large overlaps regions behind seeds. However if seeds are inexpensive then another possibility is simply to place them randomly.

2.2.1. Simulation results on position error

The simulations presented here are motivated by an actual scenario of 200 sensors distributed randomly over a square region $6r \times 6r$. This gives a local neighborhood size of roughly 20, which we know from our previous analysis to give good distance estimates. We investigate two seed placement methods: (1) all seeds are randomly placed and (2) four are hand placed at the corners and the rest are randomly placed. Figure 5 shows the location estimation accuracy averaged over 100 runs with increasing numbers of seeds.

We can see that location accuracy is reasonably high even in the worst case scenario with all randomly placed seeds. Accuracy improves with the hand placement of a few. However, the accuracy of both strategies converge as the number of seeds increases and
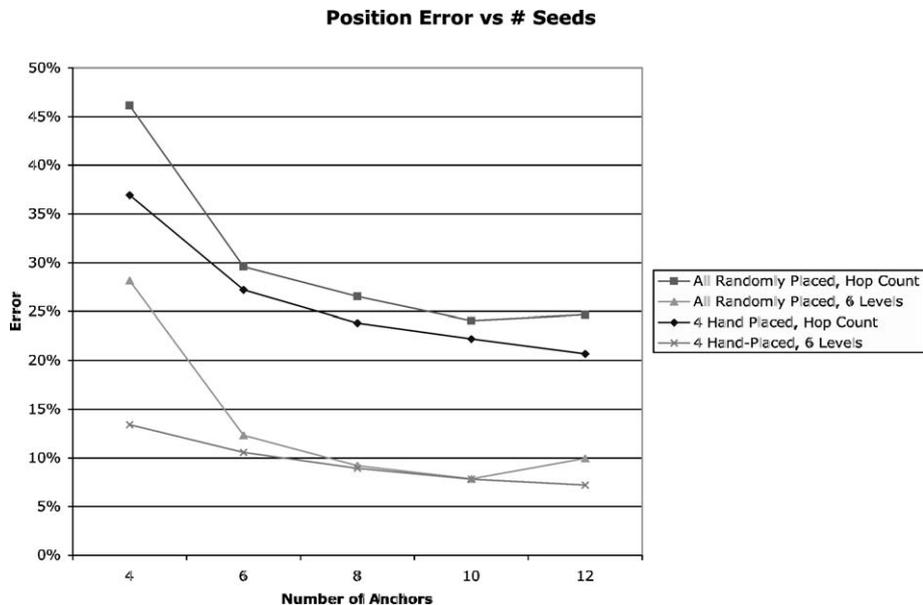

Figure 5. Graph of position error versus number of seeds for two different seed placement strategies. Position error for smoothed hop count and 6 level radio strength distance estimates are shown.

the improvement levels off at about ten seeds. These results suggest that reasonable accuracy can be achieved by carefully placing a small number of seeds when possible or using a large number of seeds when one is unable to control seed placement.

2.3. Theoretical limit on resolution

There is a fundamental limit to the accuracy of any coordinate system developed strictly from the topology of the sensor graph no matter how many seeds we use. We can think of each sensor as a node in a graph, such that two nodes are connected by an edge if and only if the sensors can communicate in one hop, i.e. they are less than r distance apart. It is possible to physically move a sensor a nonzero distance without changing the set of sensors it communicates with, and thus without changing any position estimate that is based strictly on communication. The old and new locations of the sensor are indistinguishable from the point of view of the gradient. The average distance a sensor can move without changing the connectivity of the sensor graph gives a lower bound on the expected resolution achievable.

Theorem 2. The expected distance a sensor can move without changing the connectivity of the sensor graph on an amorphous computer is $(\pi/4n_{\text{local}})r$.

Proof. Let Z be a continuous random variable representing the maximum distance a sensor p can be moved without changing the neighborhood. The probability that Z is less than some real value z is

$$F(z) = Pr(Z \leqslant z) = 1 - e^{-\rho A(z)}$$

which is the probability that there is at least one sensor in the shaded area $A(z)$ (figure 6). The area $A(z)$ can be approximated as $4rz$ when z is small compared to r and we expect z to be small for reasonable densities of sensors. The expected value of Z is

$$E(Z) = \int_0^\infty z \dot{F}(z) \, dz \tag{7}$$

$$= \int_0^\infty \rho 4rz e^{-\rho 4rz} \, dz \tag{8}$$

$$= -ze^{-\rho 4rz} \Big|_0^\infty + \left(-\frac{1}{\rho 4r} \right) e^{-\rho 4rz} \Big|_0^\infty \tag{9}$$

$$= -\left(z + \frac{1}{\rho 4r} \right) e^{-\rho 4rz} \Big|_0^\infty \tag{10}$$

$$= r \left(\frac{\pi}{4n_{\text{local}}} \right), \tag{11}$$

where equation (9) is by the product rule.[2] □
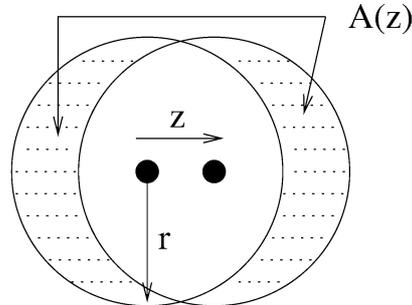
[2] Proof courtesy of Chris Lass.

Figure 6. A sensor can move a distance z without changing the connectivity if there are no sensors in the shaded area.

Hence, we do not expect to achieve resolutions smaller than $\pi/(4n_{\text{local}})$ of the local communication radius r on an amorphous computer. Whether such a resolution is achievable is a different question. For $n_{\text{local}} = 15$. this implies a resolution limit of $0.05\,r$, which is far below that achieved by our algorithm.

3.   Improving estimates using inter-sensor distance measurements

One virtue of our algorithm is that it can function in the absence of direct distance measurements. At the same time, our algorithm can be easily generalized to incorporate direct distance measurements if available. For example, suppose that sensors are able to estimate the distance of neighboring sensors through radio strength, then these estimates can easily be used in place of r, or one hop.

In this section, we show how the error in position estimates changes as we allow multiple levels of quantization. By quantization we mean coarse distance estimates, e.g., a sensor with 2 levels of quantization can determine whether its neighbor is within one mini hop or two mini hops. Figure 5 shows simulation results for the position error, given six levels of quantization, for both seed placement strategies. We see that six levels of quantization information gives a substantial improvement in accuracy over smoothed hop counts, more than what is achieved by the addition of seeds.

Figure 7 shows the effect of different amounts of signal strength information on location estimation accuracy for eight seeds. We see that position accuracy increases with increased levels of quantization. Beyond 7 levels there are diminishing returns. Our original position estimates based on hop count with no quantization yield a position accuracy between 2 and 3 levels of quantizations. This is because we use local averaging to improve the distance estimates. We plan to investigate how smoothing could be used in conjunction with quantization in the future.

We get very high position accuracy with six levels of quantization: error less than 10%. At this level of accuracy with a radius of 20 feet, we could discern locations within 2 feet, which is comparable to commercial GPS.
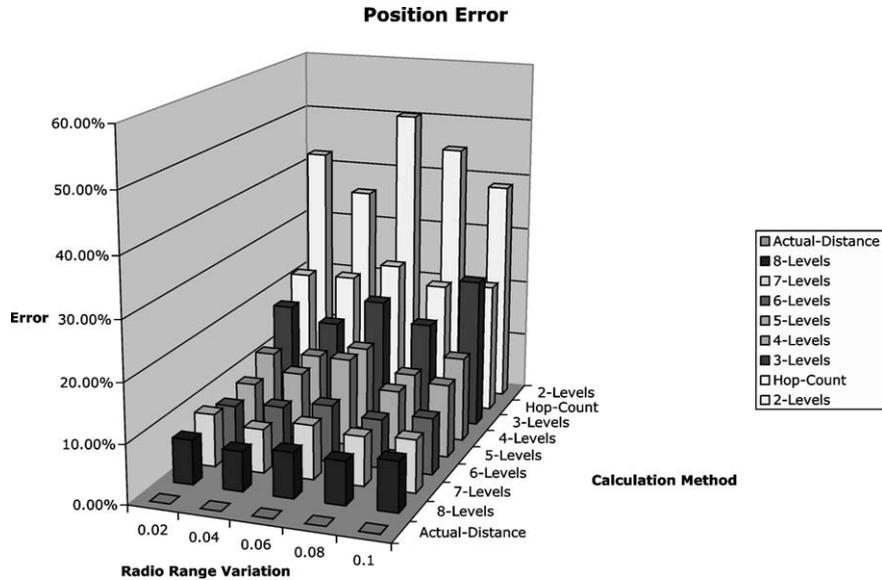
Position Error

Figure 7. Graph of the effect of 0–10% communication radius variation and different levels of signal strength quantization on location estimation accuracy.

4.   Robustness

Up to this point we have assumed that each of our sensors had the same communication radius r. In a real-world application we would expect to see variations in radio range from sensor to sensor. Our algorithm can also tolerate variations in communication radius. In figure 7 we show the error in distance estimate and position estimates when we allow up to 10% random variation in the communication radius. As we can see, the position estimates are reasonably robust to variation in sensor communication radius, tolerating up to 10% variation in range with little degradation.

The algorithm can also adapt automatically to the death and addition of sensors and seeds. If sensors are added, they can locally query neighbors for gradient values and broadcast their value. If this causes any of their neighbors distances estimates to change then those changes will ripple through the network. As a sensor receives new gradient values it can just factor that into the multilateration process. New seeds simply initiate gradients and any sensor that hears a new seed can then incorporate that seed value into the multilateration process. Prior location estimates will serve as good initial locations for multilateration ensuring fast convergence.

If we assume that sensors randomly fail, then the accuracy is not affected unless the average density falls below 15. If enough sensors in a region die then distance estimates will be affected since gradients must be propagated around the perimeter of the "dead zone", leading to overestimates. However, regional failures can be easily corrected by randomly sprinkling new sensors in that area.

The effect of seed failure depends on their placement strategy. Random placement would be more statistically robust in the face of seed failure. Other placement strategies would be more fragile. In these regimes, sensors have to recognize that seeds have failed and then exclude them from multilateration. This could be achieved with active monitoring of neighbors' aliveness to produce active gradients.

Our algorithm can tolerate a certain amount of message loss, because there are multiple redundant paths from seeds to sensors and therefore distance estimates are repeated many times. In general, the error caused by occasional message loss is unlikely to be anywhere close to the error caused by the random distribution of sensors.

5. Experimental results

In this section we evaluate the performance of our localization algorithm on real sensor network hardware. We explore the use of acoustic ranging as our source of direct inter-sensor distance measurements.

Our hardware platform is comprised of 100 Berkeley MICA2 motes [Hill et al., 5] each the size of two AA batteries. The MICA2 mote consists of an Atmel 128 micro-controller and a Chipcon CC1000 radio transceiver operating at 433 MHz. Attached to the mote is a sensor board with a Sirius PS14T40A 4.3 kHz sounder and a Panasonic WM62A microphone. Figure 8 shows a picture of a single MICA2 mote.

In our experiments we used 35 motes evenly distributed in a 5×7 grid across a $11' \times 16'$ area with a 83 cm node to node grid spacing. The motes were placed outside on a large cement sidewalk similar to the one shown in figure 9.

5.1. Acoustic ranging

Acoustic ranging is a technique for measuring the distance between a sender and receiver using the time of flight difference between radio and audio signals. A sender simultaneously sends out a radio message and an audio signal. Receivers timestamp
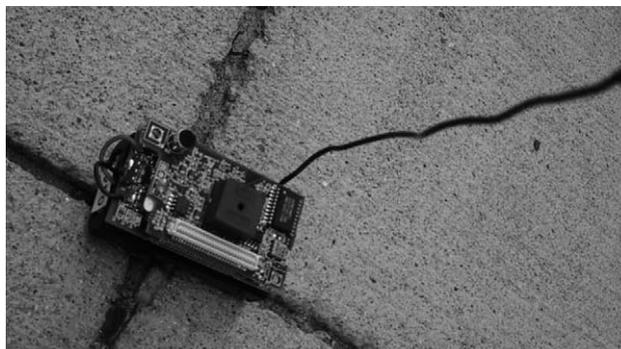

Figure 8. A Berkeley MICA2 mote with sensor board.

Figure 9. Motes placed on a 5 × 7 grid aligned with an outside cement sidewalk.

incoming radio messages and then measure the delay to the receipt of the subsequent audio signal. Finally, the distance is calculated by multiplying this delay by the speed of sound.

We employed MICA2 acoustic ranging software developed by Vanderbilt University [Maroti, 9]. Their algorithm uses a train of 16 12.5 msec 4.3 kHz pulses separated by 50–72 msec pauses. Both the sender and receiver know the exact pulse train timings. The receiver digitally records the incoming signal during the active periods of the pulse train. This buffer of samples is then passed through a 4–5 kHz band pass filter. The beginning (and end) of the pulse train is found when the average absolute value of the signal goes above (or below) the total average absolute value of the signal. The signal is recognized if the length of the pulse train is between minimum and maximum bounds. If recognized then the distance is calculated using the delay between receipt of the radio signal and the beginning of the pulse train.

In order to measure acoustic ranging performance, we collected distance measurements between motes in the grid. Each mote was given a chance to launch 48 successive ranging experiments. We then compared the average of the measurements received from these experiments against their actual ground truth distances. Figure 10 compares ranging estimates against the corresponding ground truth distances. Figure 11 shows the ranging information for one node, node 119. The mean values of the range data are shown as circles centered around the neighbor nodes. Zero error measurements would correspond with all the circles intersecting at node 119.

Out of $35^2 - 35 = 1190$ possible ranging measurements, 664 measurements were successfully gathered. The error for these 664 measurements had a mean of 15.8 and standard deviation of 25.7, which is around 19 percent of the 82.5 cm node to node grid spacing. Our results suggest that, beyond several meters, acoustic ranging is not successful. However, within that distance the mean error in ranging measurements remains constant, as shown in figure 10. In our experimental system, each node has, on average, ranging estimates for 19 other nodes. This large neighborhood size is an artifact of the small area within which we conducted our experiments; if we had placed our motes further apart, we would expect to see much smaller neighborhood sizes.

The large neighborhood sizes observed here reduce the applicability of these results to real world sensor networks, few of which will be able to boast such excellent ranging connectivity. In addition, the high degree of acoustic connectivity fails to ade-
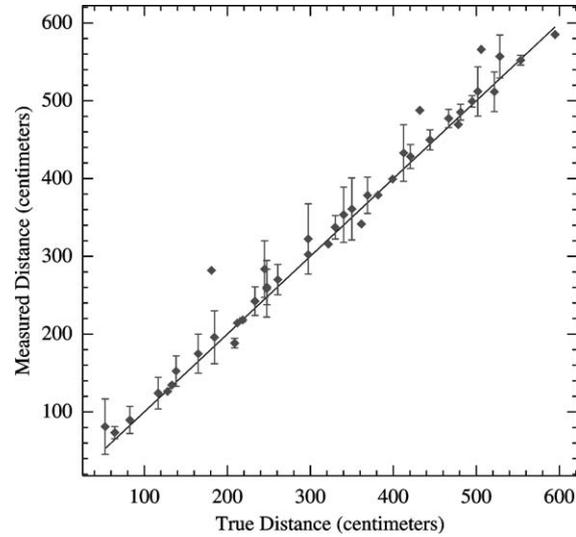
Figure 10. Acoustic ranging measurements. The solid line indicates the performance of an ideal distance estimator. Points indicate the mean of multiple range measurements made at the same distance while the error bars indicate the standard deviation of those measurements.
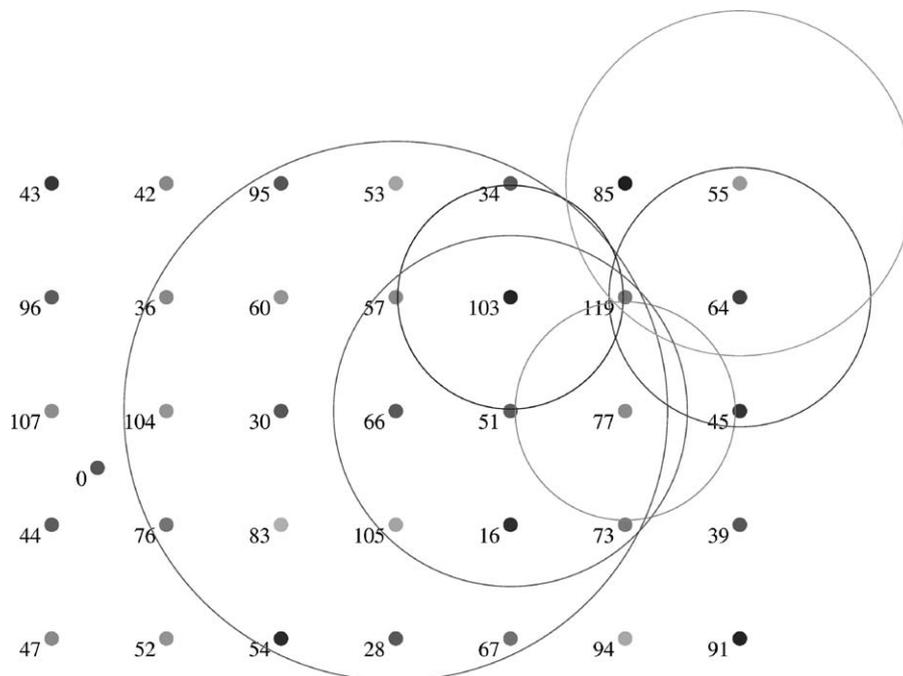

Figure 11. Ranging information for node 119. Ranging estimates are only shown for nodes less than two hops away. Node 119 has ranging information for nodes 66, 103, 55, 64, 77 and 51.

quately stress the localization algorithm; with sufficient acoustic connectivity, all nodes can simply perform multilateration calculations based on direct ranging estimates from the seeds. To remedy both these problems, we introduced the notion of a *ranging limit parameter* in our experiments. When such a limit is set, nodes refuse to communicate with other nodes that are farther away than the limit value. This constraint effectively reduces the communications radius and thus the neighborhood size. By setting the limit value appropriately, our experimental results far better model real world sensor networks with sparse node distributions. We conducted the same localization experiment with limit values of ∞ and 125 cm.

5.2. *Empirical localization results*

In this section, we present acoustic ranging based localization results produced with the same 5×7 grid. The 664 acoustic range estimates were distributed through the network using the gradient propagation algorithm and positions were calculated using multilateration. The four corner nodes were selected as seeds.

   We present results for two different limit values, ∞ and 125 cm. For each of those cases, we show localization results calculated with and without straightness factor compensation. As discussed in section 2.1, gradients in sparse sensor networks overestimate
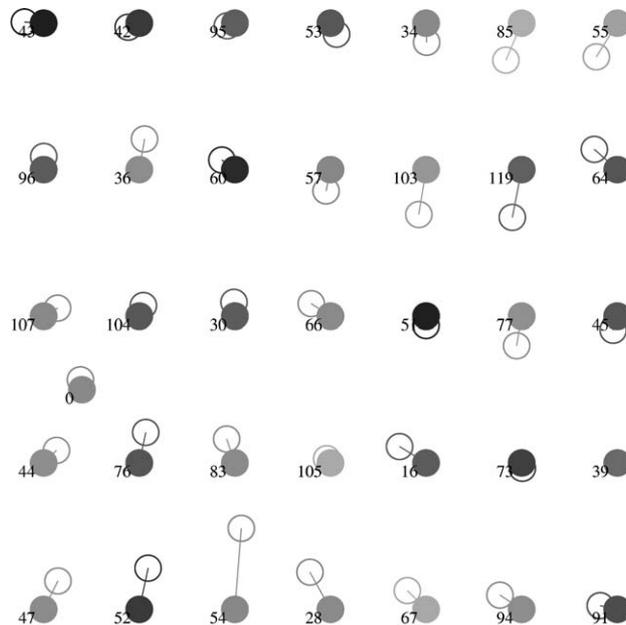

Figure 12. Ground truth and estimated positions for a 5×7 grid with *limit* $= \infty$ and no straightness factor compensation. The four corners are the seed nodes. Filled circles represent the ground truth positions while open circles represent position estimates derived from multilateration. The distance error has a mean of 16 cm and a standard deviation of 9 cm.
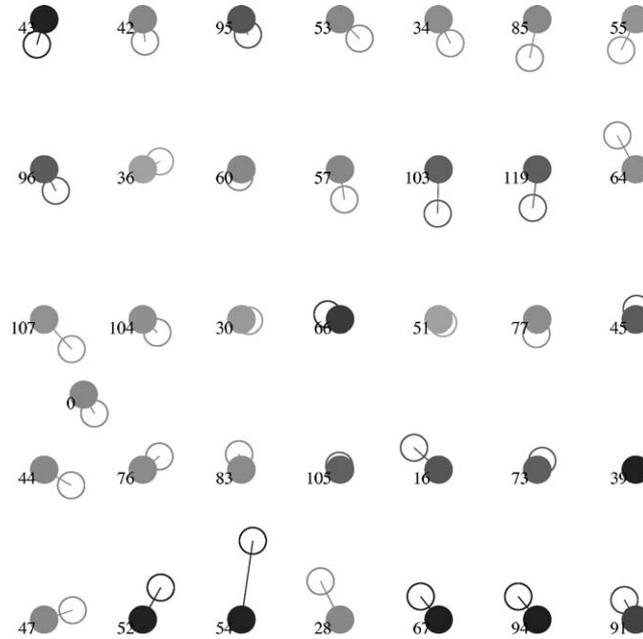
Figure 13. Ground truth and estimated positions for a 5 × 7 grid with *limit* = ∞ and straightness factor compensation. The distance error has a mean of 17 cm and a standard deviation of 8.6 cm.

the distance between a sensor and a seed. For randomly distributed sensors, formula (5) can be used to calculate the straightness factor, which when multiplied by the distance estimate compensates for the overestimation. In our experiments, the seeds directly calculate a straightness factor from the existing sensor distribution.

Each seed uses the gradient distance and actual distance between itself and other seeds to compute a set a straightness factors. Each seed averages these straightness factors and propagates that average throughout the network. Before performing multilateration, nodes multiply their gradient distance to each seed by that seed's corresponding straightness factor. Niculescu and Nath [13] propose a slightly different algorithm where nodes correct a seed's gradient distance by using the straightness factor of the seed closest to them rather than the straightness factor of the seed whose distance they are trying to correct.

Figures 12–15 show the estimated positions for 35 motes overlaid on top of the ground truth positions. Figures 12 and 13 show results with and without straightness factor compensation with infinite ranging limit set. Figures 14 and 15 show results with and without straightness factor compensation with the ranging limit set to 125 cm (just over one hop distance). When simulating a high density sensor network (by setting the range limit to ∞), our localization algorithm estimated positions with a mean error of 16 cm without using straightness factor compensation and 17 cm with it. When modeling low density networks (by setting the range limit to 125 cm), our algorithm estimated
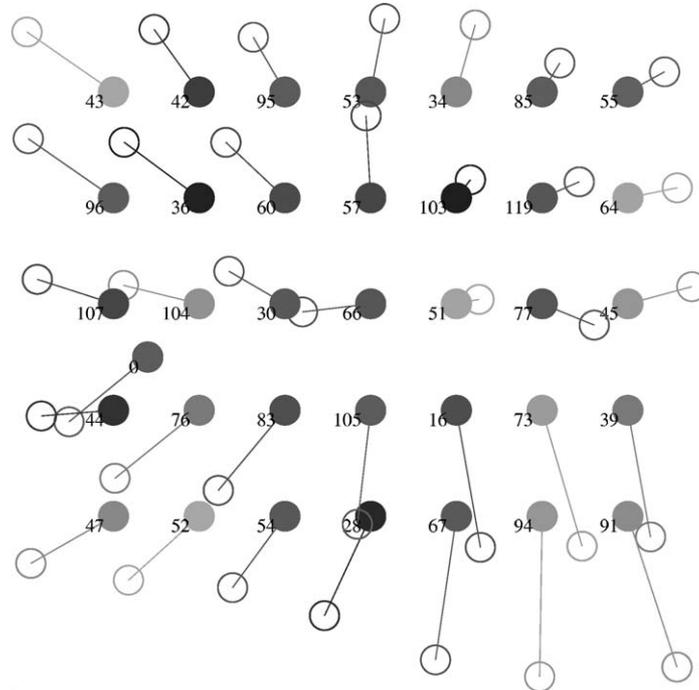
Figure 14. Ground truth and estimated positions for a 5×7 grid with *limit* $= 125$ cm and no straightness factor compensation. The distance error has a mean of 75 cm and a standard deviation of 27 cm.

positions with a mean error of 75 cm and 37 cm without and then with straightness factor compensation, respectively.

Several interesting observations can be made from these experiments. First, there is the radial pattern in estimated node positions seen when using uncompensated localization with a small ranging limit value, as shown in figure 14. Second, the straightness factor compensation has little if any benefit in high density networks (ranging limit of ∞), but provides a dramatic increase in localization accuracy when used in networks with lower neighborhood density (ranging limit of 125 cm).

Both of these results can be explained by the fact that as network density decreases, the rate at which gradient path length overestimates true distance increases. As a result, in low density networks without straightness factor compensation, seed distances are overestimated in proportion to their actual distance. When combined with a seed placement policy that dictates placement on the network perimeter, the localization algorithm produces a radial error pattern as shown in figure 14. In higher density networks, the gradient path length and the true distance are close enough together so as not to contribute significant systematic error for the straightness factor compensation to rectify. Another perspective is that since the total change introduced by the straightness factor is proportional to the number of hops, it becomes less significant as the number needed to traverse a path in the network decreases.
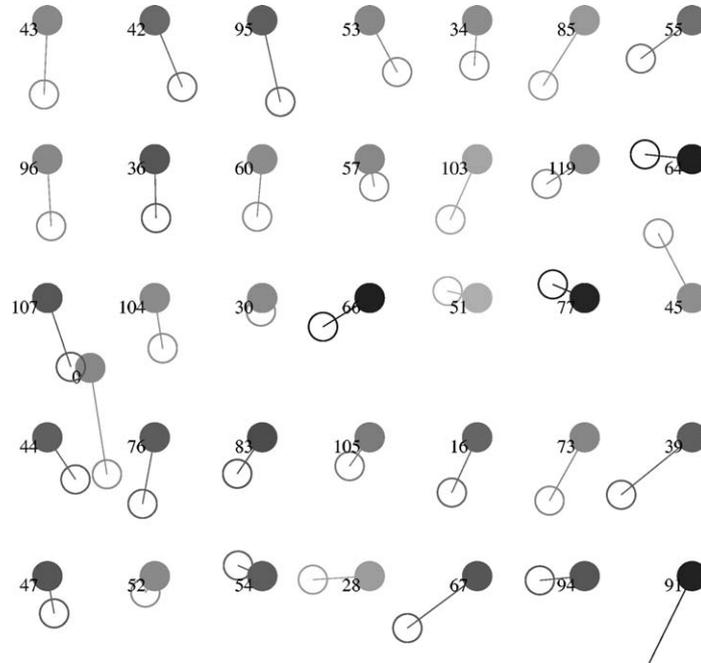
Figure 15. Ground truth and estimated positions for a 5×7 grid with *limit* $= 125$ cm and straightness factor compensation. The distance error has a mean of 37 cm and a standard deviation of 13 cm.

6.   Conclusions and future work

In this paper, we present an algorithm to self-organize a global coordinate system on an ad hoc wireless sensor network. Our algorithm relies on distributed simple computation and local communication only, features that an ad hoc sensor network can provide in abundance. At the same time it is able to achieve very reasonable accuracy and the error is theoretically analyzable. The algorithm gracefully adapts to take advantage of any improved sensor capabilities or availability of additional seeds. Experimental results indicate that our localization algorithm combined with acoustic ranging works well in practice. Given that so much can be achieved from so little, an interesting question is whether more complicated computation is useful in these simple scenarios. More work is needed however to develop accurate coordinate systems in the face of large obstacles, such as buildings and or walls, that prevent sensors from being uniformly distributed.

Acknowledgements

References

[1] P. Bahl and V.N. Padmanabhan, RADAR: An in-building RF-based user location and tracking system, in: *Proc. INFOCOM 2000*, 2000, pp. 775–784.

[2] W. Butera, Programming a paintable computer, Ph.D. thesis, MIT Media Lab. (2002).

[3] U.W. Corporation, http://www.uswcorp.com/USWCMainPages/our.htm.

[4] L. Doherty, L.E. Ghaoui and K.S.J. Pister, Convex position estimation in wireless sensor networks, in: *Proc. INFOCOM 2001*, 2001.

[5] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler and K. Pister, System architecture directions for networked sensors, in: *Proc. of Conf. on Architectural Support for Programming Languages and Operating Systems IX (ASPLOS-IX)*, 2000.

[6] B. Hofmann-Wellenhoff, H. Lichtennegger and J. Collins, *Global Positioning System: Theory and Practice*, 4th ed. (Springer, Berlin, 1997).

[7] L. Kleinrock and J. Silvester, Optimum tranmission radii for packet radio networks or why six is a magic number, in: *Proc. of National Telecommunication Conf.*, 1978, pp. 4.3.1–4.3.5.

[8] N. Lynch, *Distributed Algorithms* (Morgan Kaufmann, Wonderland, 1996).

[9] M. Maroti, Acoustic ranging on MICA2's, Private communication (2003).

[10] J.D. McLurkin, Algorithms for distributed sensor networks, Master's thesis, UCB (1999).

[11] W. Mendenhall, D. Wackerly and R. Scheaffer, *Mathematical Statistics with Applications* (PWS, Boston/Kent, 1989).

[12] R. Nagpal, Organizing a global coordinate system from local information on an amorphous computer, AI Memo 1666, MIT (1999).

[13] D. Niculescu and B. Nath, DV based positioning in ad hoc networks, Telecommunication Systems 22(1) (2003) 267–280.

[14] Philips, Shivendra, Panwar and Tatami, Connectivity properties of a packet radio network model, IEEE Transactions on Information Theory 35(5) (1998) 1044–1047.

[15] N.B. Priyantha, A. Chakraborty and H. Balakrishnan, The cricket location-support system, in: *Proc. of MOBICOM 2000*, 2000, pp. 32–43.

[16] A. Savvides, C. Han and M. Strivastava, Dynamic fine-grained localization in ad-hoc networks of sensors, in: *Proceedings of ACM SIGMOBILE*, 2001, pp. 166–179.

[17] C. Whitehouse, The design of Calamari: An ad-hoc localization system for sensor networks, Master's thesis, University of California at Berkeley (2002).