Localization algorithms for sensor networks using RF signal strength
CS 252 Class Project

XuanLong Nguyen & Tye Rattentbury

May 9, 2003

Abstract

We propose and investigate: (1) models for RF signal strength and (2) localization algorithms for ad-hoc sensor networks.

1 Introduction

Ad-hoc sensor networks represent a future branch of computer architecture research. As Moore's law continues to drive more and more hardware onto a single chip, small wireless sensor nodes represent the foreseeable horizon for hugely distributed computing. There are important fundamental design issues that must be overcome before such nodes can be truly useful. One of the more important issues is localization. What benefit is a sensor if its readings are not strongly correlated with the real world? The problem of node localization in ad-hoc sensor networks directly addresses the need to place gathered data in context.

To date, there have been many attempts to solve the ad-hoc network localization problem (Hightower and Borriello, 2001b), (Bulusu et al, 2000). None of these attempts has provided a provably good solution to the localization problem. Failed attempts are often attributed to hardware inaccuracies (Whitehouse, 2002). However, as we will show, marked improvements can be made by: first, analyzing the hardware for predictable characteristics on which computational solutions can be built; and second, relying on well-founded mathematical optimization theories when designing computational solutions.

In this paper, we analyze the use of RF signal strength in wireless node distance estimation. Then, based on characteristics of this distance estimation, we develop an optimization technique for solving the node localization problem.

2 Calibration

We are working on the ad-hoc sensor network localization problem using only RF signal strength. Many researchers in this area have indicated how hard this problem is (Whitehouse, 2002), (Seidel and Rappaport, 1992), (Bahl and Padmanabhan, 2000), (Girod and Estrin, 2001), (Savarese et al, 2001), (Doherty et al, 2001), (Savvides et al, 2001), (Ward et al, 1997), (Priyantha et al, 2000). In fact, most of these projects have added ultrasound sensors and transmitters to their nodes to combat the high-error in RF signal strength.

RF signal strength (RSSI) is measured in volts by the node receiving the message. General antenna theory provides the following model given that the sending and receiving antennas are focused on the same radio frequency:

$$RSSI \propto PT \cdot d^n$$

where d is distance, $n \in [-2, \ -0.1]$, and PT is the sending signal voltage (Seidel and Rappaport, 1992).

Unfortunately, this ideal model is very inaccurate due to both macro- and micro-node factors. For example, multipath and ambient EM noise interference are macro-node factors that strongly affect RSSI measurements (Whitehouse, 2002). Similarly, variation in the specific transmit and receive frequencies of node radios, physical antenna orientation, and fluctuations in the power source for the radios are examples of micro-node factors that influence RSSI. Consequently we treat each node's radio as unique and try to estimate parameters that capture all of the important factors in RF propagation.

More specifically, we propose the following model:

$$\frac{1}{RSSI_{i \to j}} \propto PT_i \cdot d_{ij}^n$$

where $RSSI_{i \to j}$ is the RSSI value read at node j when it is receiving a signal from node i, PT_i is node i's transmitting voltage, and d_{ij} is the distance from node i to node j. The reason the RSSI value is inverted is that the ChipCon CC1000 radio outputs inverted RSSI voltage readings.

Filling the model in with parameters, we get:

$$b_j \left(\frac{1}{RSSI_{i \to j}} \right) + c_j = a_i \cdot d_{ij}^n$$

where b_j and c_j handle the proportionality and a_i is supposed to capture PT_i. Since we are actually interested in d_{ij}, this model is not linear in its parameters. However, we can still build a large linear system over all possible node pairs to learn these parameters. To handle the scaling in the system, we force one of the a_i parameters to be 1. Intuitively this normalization corresponds to calculating all parameters relative to one node's transmission voltage. To learn the best value for n, we apply 4-fold cross validation. The data we are using was generated by Kamin Whitehouse. There were two experimental set-ups: one with 16 nodes in a 4x4 grid with 3 meter spacing and one with 24 nodes in a 3x8 grid also with 3 meter spacing. The surface was essentially flat (i.e node positions in R^2.)

We measure error in this model using the following equations:

$$\begin{aligned} TE_{ij} &= |(estimate\ distance)_{ij} - (true\ distance)_{ij}| \\ PE_{ij} &= \frac{TE_{ij}}{(true\ distance)_{ij}} \end{aligned}$$

Given this data and theory-based model, we achieved test-set geometric average PEs around .36. Some test-sets achieved as low as .28 and some as high as .55. In other words, given this model, some distance estimates were as low as half or as much as twice their true distance. The average TEs ranged from 1.7 to 2.5 meters. Clearly, these results are bad. However as you can see in figure 1, the RSSI values for fixed distance not only vary greatly, but the ranges over which they vary overlap. Hence, a perfect model does not exist.

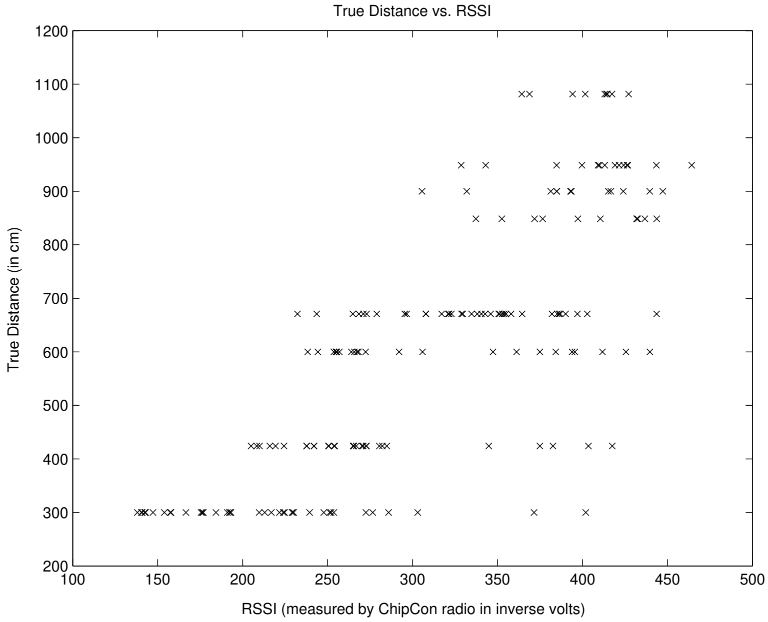

Figure 1: This data is from the fifth data set of the 4x4 grid experiment. Clearly, finding an accurate model (linear or not) for distance estimates from RSSI is impossible.

An interesting note is that the best values for n ranged from about -0.5 to about 0.2, indicating that our antenna theory-based model, which proposes $n \in [-2, \ -0.1]$, does not accurately approximate our data. Consequently, we changed the signal propagation model to the following:

$$b_j \cdot RSSI_{i \to j} + c_j = a_i \cdot d_{ij}^n$$

Re-estimating these parameters produced the results in tables 1 and 2.

Test Set	Geom. Avg. PE	Avg. TE (m)	Optimal n
1	17.45%	1.74	0.01
2	16.22%	1.49	0.02
3	15.59%	1.30	-0.01
4	16.62%	1.40	-0.02
5	12.98%	1.15	-0.01

Table 1: results from the 4x4 grid.

Although these numerical results demonstrate that the second model we proposed for RF signal propagation is better than the theory-based model (at least on Kamin Whitehouse's data), they do not reveal any information about why the results are of the order that they are. From figure 1, one might conclude that large variance in RSSI readings coupled with the fact that RSSI intervals for a given distance overlap, explains the high error in this model. But there is more that can be said. From figure 2, it is clear that error in our calibration model follows a specific pattern: short distances are overestimated and long distances are underestimated. We will refer to data that follows this pattern as *distorted*. Moreover, these mis-estimations appear to follow a somewhat linear relationship.

Test Set	Geom. Avg. PE	Avg. TE (m)	Optimal n
1	16.67%	1.71	-0.20
2	18.69%	1.93	-0.39
3	21.00%	2.12	0.13
4	15.92%	1.89	-0.08
5	15.93%	1.45	-0.13

Table 2: results from the 3x8 grid.

Table 3 presents the line parameters that best fit the $((true\ distance)_{ij}, (estimated\ distance)_{ij})$ pairs on each test-set.

Test Set	Slope	Intercept (m)
4x4 grid, set 1	0.289	3.06
4x4 grid, set 2	0.410	2.88
4x4 grid, set 3	0.528	3.18
4x4 grid, set 4	0.587	2.90
4x4 grid, set 5	0.781	1.76
3x8 grid, set 1	0.302	3.23
3x8 grid, set 2	0.416	3.44
3x8 grid, set 3	0.304	5.59
3x8 grid, set 4	0.131	5.84
3x8 grid, set 5	0.530	3.25

Table 3: line parameters that indicate fit of RSSI-based distance estimates to true distances in real data sets

For comparison, we ran the data through the calibration scheme developed in (Whitehouse, 2002). This calibration scheme was developed for time-of-flight data, but is applicable because it represent a purely linear model:

$$d_{ij} = (\alpha_i + \alpha_j) \cdot \left(\frac{RSSI_{i\to j} + RSSI_{j\to i}}{2}\right) + (\beta_i + \beta_j)$$

The results are summarized in table 4. Comparing these results with the results in tables 1 and 2 indicates a true non-linearity in the data.

3 Localization algorithm

Our problem setting is as follows: Given n sensor nodes $X_1, \dots, X_n$, some of which are designated as anchor nodes whose positions are exactly known, distance estimates $\{d_{ij}$ of a subset of pairs of nodes $(i, j)\}$, we want to recover the true positions of all sensor nodes in the network.

3.1 A maximum likelihood formulation for non-distorted networks

As we discussed in previous sections, these distance estimates are obtained from RF signal strength and can be extremely inaccurate. Furthermore, the distance estimates are not available for all

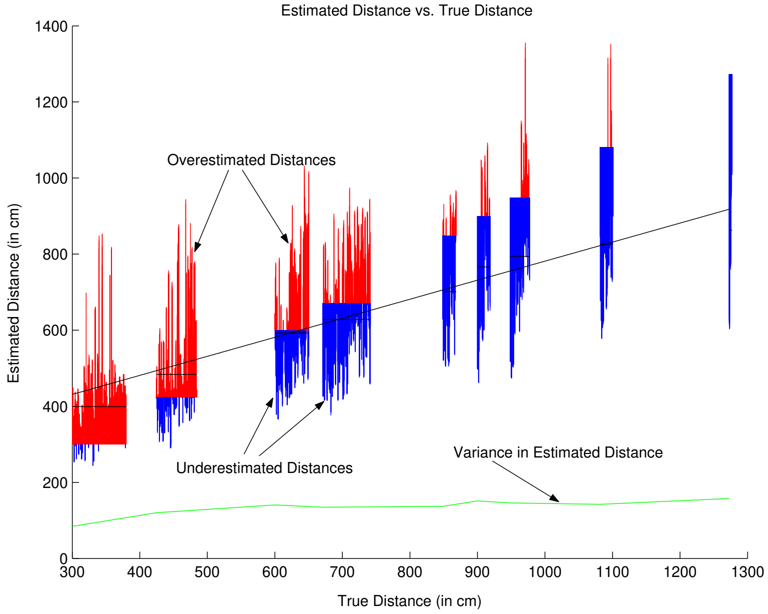

Figure 2: This data is from the fifth data set of the 4x4 grid experiment. Clearly, there is a pattern in the distance estimation error. Because the experiment was performed on a grid, there are actually only a few true distances represented in the data. To help visualize this, estimated distances are drawn in clusters which are spread out horizontally, despite having the same true distance. Average estimated distances for each true distance are represented as black horizontal lines across the data cluster. The sloping black line is the best linear fit to the average distance estimates. Additionally, distance estimates that are overestimates are drawn as a red line from their true distance vertically up to the estimated distance value. Similarly, distance estimates that are underestimated are drawn as blue lines from their true distance vertically down to the estimated values.

pairs of nodes. Usually, for each node, the distances to only nodes that are within a certain radius, depending on their specific orientation, battery power, frequency and other physical properties, can be measured. In this subsection, we consider the non-distorted network first. Distorted networks shall be considered afterward.

All localization algorithms can be characterized in terms of 3 key features:

1. How the position estimates for X_i are initialized.
2. What is the underlying objective function that it tries to optimize to obtain X_i.
3. What is the algorithm for solving the aforementioned optimization algorithm. Is it centralized or distributed? How is the network topology exploited?

By investigating some of the existing algorithms in the literature in terms of these 3 features, we will be able to shed light into the implicit assumptions that they make and to come up with our own algorithms.

Test Set	Geom. Avg. PE	Avg. TE (m)
4x4 grid, set 1	20.38%	1.72
4x4 grid, set 2	15.10%	1.42
4x4 grid, set 3	16.03%	1.39
4x4 grid, set 4	19.06%	1.40
4x4 grid, set 5	13.23%	1.19
3x8 grid, set 1	31.44%	1.77
3x8 grid, set 2	37.61%	1.95
3x8 grid, set 3	31.37%	2.16
3x8 grid, set 4	31.59%	1.89
3x8 grid, set 5	29.71%	1.92

Table 4: calibration results for the purely linear model proposed in (Whitehouse, 2002)

3.1.1 Initialization

Several algorithms use a simple step called "Bounding Box" method (Simic, 2002) to obtain the initial estimate of all X_i. The idea is to exploit inequality constraints given by the the distance estimates. For each pair (i, j) such that d_{ij} is available, we have:

$$\begin{aligned} X_{j1} - d_{ij} &\leq X_{i1} \leq X_{j1} + d_{ij} \\ X_{j2} - d_{ij} &\leq X_{i2} \leq X_{j2} + d_{ij} \end{aligned}$$

Solving this set of inequality constraints in a distributed manner gives a bounding range for each node X_i (excluding the anchor nodes). Each X_i's position is then initialized to be the center of the bounding box.

Doherty et al (2001) employ a related inequality which bounds locations using the L_2 norm (as opposed to the L_1 norm). Doherty et al assumes that all pairs of connected nodes must be uniformly bounded within a some distance R – and poses these constraints as a convex optimization problem. This approach is more computationally expensive since the system of equations must be solved in a centralized manner (i.e. each node cannot perform its own bounding box calculation in Doherty et al's scheme), and useful only for isotropic networks where each node has a significant number of neighbors strandling the entire distance R boundary. Clearly, realistic ad-hoc sensor networks do not demonstrate these characteristics as was demonstrated in the previous section.

In principle, any algorithm can be used to provide initial estimates if further accuracy can be assured and computational cost increase is not a concern.

3.1.2 Objective function

Most existing algorithms in the literature attempt to minimize the least square error of the distance estimate without precisely describing *what* the underlying objective function is and *why* it is the right thing to minimize on. This impreciseness and ambiguity make it very difficult to evaluate and understand the strength and weakness of each algorithm.

Implicit in these approaches is a key assumption that the distance estimate d_{ij} are Gaussian distribution with mean $||X_i - X_j||$, and that the estimates for all pairs (i, j) are statistically independent.

The independence assumption about pairwise estimates is clearly not true, but may be reasonable for reducing computational costs. Let E be the set of all pairs, and E_o be the set of pairs of nodes whose distances were measured (through calibration in the previous section). Assume that for each pair (i, j), d_{ij} is a normal random variable with mean $||X_i - X_j||$ and variance σ_{ij}. Coupled with the independence assumption, we are able to write down the log likelihood of observations $(d_{ij})_{i,j}$ as follows:

$$\begin{aligned}\ln P(d|X_1, \ldots, X_n, \sigma) &= \ln \prod_{(i,j)\in E_o} \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} \exp -\frac{1}{2\sigma_{ij}}(d_{ij} - ||X_i - X_j||)^2 \\ &= -\sum_{(i,j)\in E_o} \frac{1}{2\sigma_{ij}}(d_{ij} - ||X_i - X_j||)^2 + \text{ constant}\end{aligned}$$

Applying the maximum-likelihood principle, one can find $X_1, \ldots, X_n$ by minimizing the weighted sum of square error:

$$L_{E_o}(X) = \sum_{(i,j)\in E_o} \frac{1}{\sigma_{ij}}(d_{ij} - ||X_i - X_j||)^2.$$

This objective function forms the basis for a host of algorithms such as the Multilateration algorithm family by Savvides et al (2001, 2002), APS (Niculescu and Nath), and Resolution of Force (Whitehouse, 2002). However, due to subtle algorithmic differences, it is clear that these algorithms are actually minimizing a closely related, but identical, objective function.

3.1.3 Optimization procedure

This is the step where most algorithms differ in details.

The family of Multilateration algorithms (Savvide et al, 2002) is most faithful to the maximum-likelihood framework presented above. For example, Iterative Multilateration initially considers pairs $\tilde{E} \subseteq E_o$ of nodes in which at least one node has to be an anchor. Then all the neighbor nodes of the anchors are updated using least squared error minimization, and they themselves become anchor nodes. This updating process progates through the whole network. Collaborate Multilateration constructs a more selective *collaborate subtree* that contains only nodes whose possible positions are "well-defined." It is through these subtrees that the new positions get updated.

Hop-Terrain/Refinement (Savarese et al, 2001) attempts to minimize a a weighted sum of squared error, where the weights are "confidences" on node positions. Each time a node's position is updated, its weight is updates as the mean of the weights on all neighboring nodes. This algorithm is known not to converge in some cases. To combat the cases that do not converge, the authors apply various ad-hoc rules to rule out position updates (Savarese et al, 2002). Also, nodes are randomly set to bad locations to improve local optima convergence. These confidence weights are not well-founded. Although they are intuitive, the update equations for the weights does not follow from the objective function. A more justified weighting scheme relies on estimated variances σ_{ij}. Weight of this sort have been used in robotics localization algorithms (Lu and Milios, 1997).

APS is a distributed algorithm that updates the position estimate through a correction factor that helps minimize the sum of square of the difference between the estimated Euclidean distances (given current estimates) and the multi-hop distances computed through propagation from anchor nodes to the rest of the network.

Whitehouse's Resolution of force algorithm advocates an APS variant that miminizes the sum of absolute value of the aforementioned differences (instead of sum of square), because he observed that the multi-hop distances obtained through propagation follow a Laplacian distribution instead of Gaussian. The problem with both approaches is that the multi-hop distances do not necessarily reflect the metric distances, whose errors were minimized in their algorithms.

3.2 Our algorithm: A regularized gradient-based least square approach

Given our formal framework presented above, we shall present a derivation of our algorithm in both a mathematical and intuitive manner.

3.2.1 Motivations

First, we would like to decide the weights $1/\sigma_{ij}$ in the objective function $L(X, \sigma)$. Initially, we thought that the variance in distance estimates should increase with the true distance, following the intuitive argument that radio signal strength is degraded more the longer it travels. However, after calibrating our RSSI-to-distance model, we found that variance fluctuated but did not follow a general trend (increasing or decreasing). For example, observe the variance line in Figure 2. So we have decided to have constant variance, which effectively removes σ_{ij} from the formula for $L_{E_0}(X)$.

Second, while minimizing the squared errors for the measured distances in E_o appears reasonable, doing so disregards the distances of other pairs in $E - E_o$. After all, the maximum-likelihood only truely makes sense when all d_{ij} are statistically independent, which is clearly not true. To account for the errors in non-measured distances resulting from our position estimates, we want to minimize the sum not only for pairs E_o, but also for pairs in $E_h := E - E_o$. That is:

$$L(X) := L_{E_o} + L_{E-E_o} = \sum_{(i,j)\in E_o} \frac{1}{\sigma_{ij}}(d_{ij} - ||X_i - X_j||)^2 + \sum_{(i,j)\in E_h} \frac{1}{\sigma_{ij}}(d_{ij} - ||X_i - X_j||)^2$$

Let us call L_{E_o} the *observed error* and L_{E_h} the *hidden error*. Our algorithm will involve updating the position estimates for each node in a distributed manner using gradient-based method. Specifically, we wish to update all X_i according to the gradient

$$\partial L/\partial X_i = \partial L_{E_o}/\partial X_i + \partial L_{E_h}/\partial X_i.$$

While $\partial L_{E_o}/\partial X_i$ can be computed easily, the problem is that we do not have the measured distances d_{ij} for $(i, j) \in E_h$.Hence, $\partial L_{E_h}/\partial X_i$ cannot be computed analytically.

Our solution is to perform gradient-descent on direction $\partial L_{E_o}/\partial X_i$ plus a corrected factor that aims to minimize the hidden error L_{E_h}. To compute this corrected factor, we need to understand the behavior of the hidden error. A key observation that came out of our investigation of RSSI calibration models is that the observed error usually involves short distances (i.e nodes that are connected), while hidden error usually involves nodes that are far away.

3.2.2 On the centering effect of our initialization with Bounding Box method

Fortunately, the behavior of the hidden error L_{E_h} can be captured by the position estimates obtained by the bounding box method. Empirically, we observed that the position estimates given by bounding box method are always pulled toward the center of the network [1] (See Figure 3). What gives rise to this phenomenon?

It turns out that this effect is controlled by the hidden error L_{E_h}. Indeed, the bounding box method constrains the position of each node through multi-hop (i.e. hidden) distances from that node to all anchors in the network. For a node whose position is estimated to be near the boundary region of the network, there are more constraints on the distances with nodes that are far away (across the center of the network) and less constraints from the boundary region. By taking the center of the bounding box as a concensus, the node's estimates tend to be pulled toward the center

[1] This phenomenon for Bounding Box and APS, which uses initial estimates given by the Bounding Box method, were also noted in Whitehouse's thesis.

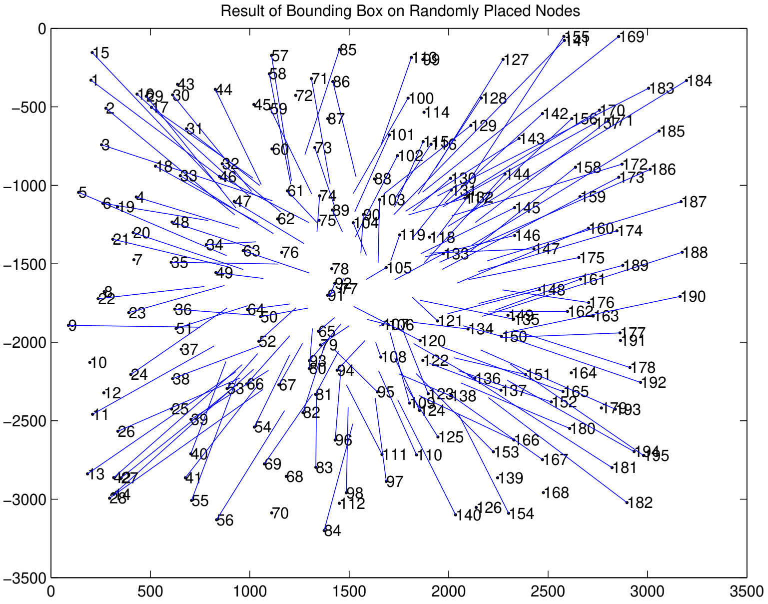

Figure 3: This data is from a randomly generated set of nodes. The connectivity was set iid with probability 0.75. The distance estimates were generated randomly according the error model seen in figure 2. Nodes labels are placed at true positions. Blue lines are drawn from true node positions to the estimated positions generated by the Bounding Box algorithm. Clearly, there is a general over-estimation toward the center of the node distribution.

of the network. This observation is specially revealing when we think about distorted networks, in which case, the longer distances are underestimated and the shorter distances are overestimated (more on this in the subsection on distorted networks). Because these hidden distances tend to be underestimated, they would pull any nodes in the boundary region toward the center of the network more than they should.

3.2.3 Our regularized objective function

The motivations and observations described in the preceeding subsections led us to come up with a new objective functin as follows. Starting from the bounding box's estimates, we would update X_i that minimize the following objective function:

$$L_\lambda(X) = \sum_{(i,j)\in E_o} \frac{1}{\sigma_{ij}}(d_{ij} - ||X_i - X_j||)^2 - \lambda \sum_{i=1}^{n} ||X_i - C||^2,$$

where C is a center of the initial position estimates. $\lambda > 0$ is called a *regularization parameter*. By minimzing L_λ, we hope to minimize the observed error L_{E_o} by pushing the pulled-in nodes back to the boundary. It is this "pushing-out" movement that helps reduce the hidden error term L_{E_h}. Now we shall derive the algorithm analytically.

3.2.4 Algorithm for non-distorted networks

Taking derivative of L_λ with respect to the first coordinate X_{i1} of $X_i = (X_{i1}, X_{i2})$, and let it be zero, we have:

$$\partial L_\lambda / \partial X_{i1} = \sum_{j:(i,j)\in E_o} 2(||X_i - X_j|| - d_{ij}) \frac{X_{i1} - X_{j1}}{||X_i - X_j||} - 2\lambda(X_{i1} - C_1).$$

Rearranging gives the following update equation:

$$X_{i1} = \frac{1}{\deg(i)} \left(\sum_{j:(i,j)\in E_o} X_{j1} + d_{ij} \frac{X_{i1} - X_{j1}}{||X_i - X_j||} \right) + \lambda(X_{i1} - C_1) \tag{1}$$

where $\deg(i)$ denotes the number of nodes j such that $(i,j) \in E_o$. A similar updating equation for the second coordinate of X_i:

$$X_{i2} = \frac{1}{\deg(i)} \left(\sum_{j:(i,j)\in E_o} X_{j2} + d_{ij} \frac{X_{i2} - X_{j2}}{||X_i - X_j||} \right) + \lambda(X_{i2} - C_2) \tag{2}$$

These updates can be interpreted as being a sum of 3 terms, the first assign the position estimate of a give node to be the center of its connecting neighbors, the second term is a correction term, which aims at minimizing the observed error L_{E_o}, while the third correction term minimizes the hidden error L_{E_h}. Our algorithm is simple, distributed, and parallel, as we summarize as follows.

1. Initialize $X = \text{BoundingBox}(d_{ij}, i, j = 1, \ldots, n)$.
2. Let $C = \frac{1}{n}(X_1 + \ldots + X_n)$.
3. Repeat until L_{E_o} stops decreasing, in which case go to 5.
4. For each node $i = 1, \ldots, n$ update X_i according to equation 1 and 2.
5. Return.

The parameter λ has the interesting role of meditating the tension between the the hidden error and the observed error. In our experiment, we usually choose λ to be between 0 and 1. The smaller λ is the slower the convergence, but the accuracy of position estimates is higher. When λ is sufficiently large, the algorithm stops after only a few iterations. Hence λ can be tuned according to the accuracy requirement and the power constraints of the sensors in the network.

Finally, for a non-distorted network both observed error L_{E_o} and hidden error L_{E_h} behave very nicely with one another. Both of these terms tend to go down together and go up together. Therefore, L_{E_o} reaches local minima at about the time $L_{E_o} + L_{E_h}$ does in our updating procedure. This feature, however, is not shared by distorted networks.

3.2.5 Simulation results for non-distorted networks

We consider a dataset of non-distorted networks randomly generated. (See Figure **??** for an example). The size of the area is scaled to be approximately 1 in both dimensions. There about 150 sensor nodes randomly distributed (uniformly within sufficiently small grids), and approximately 10-12 anchor nodes such that all sensor nodes are within the convex hull formed by the anchors.

Dataset	BBox	Res-of-force	$\lambda = 0$	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.5$
1	0.044	0.034	0.035 (12)	0.027 (21)	0.029 (12)	0.036 (3)
2	0.052	0.051	0.042 (19)	0.038 (15)	0.038 (10)	0.041 (4)
3	0.058	0.064	0.047 (47)	0.037 (25)	0.034 (12)	0.037 (3)
4	0.050	0.045	0.040 (26)	0.039 (16)	0.039 (9)	0.044 (3)
5	0.043	0.075	0.035 (15)	0.026 (16)	0.027 (7)	0.032 (2)
6	0.072	0.085	0.063 (23)	0.055 (15)	0.054 (13)	0.060 (3)
7	0.041	0.062	0.029 (25)	0.024 (13)	0.024 (8)	0.036 (4)
8	0.040	0.067	0.036 (11)	0.031 (17)	0.030 (8)	0.034 (2)
9	0.051	0.069	0.043 (34)	0.039 (16)	0.040 (8)	0.044 (2)

Table 5: Average error of position esimates in 9 non-distorted networks with perfect distance estimate. Each network has about 150 nodes, 1500 connections. The size of the network is scaled to approximately cover an area of 1x1 metric unit square. The error is reported in the scaled metric unit. The numbers in parentheses are the numbers of updating iterations that our algorithm runs before it stops. Our algorithm was run with three values of regularized parameter λ and also a non-regularized case with $\lambda = 0$. BBox denotes the bounding box algorithm, and Res-of-force denotes Whitehouse's Resolution of force algorithm. The corresponding plot of this table is Figure 4.

The graph is very sparse – there are approximately 1500 connections. In this data set, the distance estimates for connected edges are exact.

Table 5 reports the error of position estimates returned by our algorithm (with different λ), the bounding box algorithm, and Whitehouse's Resolution of Force algorithm. The reported error is averaged over the error of all (non-anchor) nodes in the network:

$$\text{error}(i) = |X_{i1} - \text{estimate}(X_{i1})| + |X_{i2} - \text{estimate}(X_{i2})|.$$

In the second set of data, we consider the same network positions, but add noise to the distance estimates: All distance estimates are true estimates plus a Gaussian error of mean 0 and standard deviation 0.05. In all cases, we see that our regularized algorithms give more consitently more accurate position estimates that both the bounding box and Resolution of Force algorithm. The magnitude of improvement is usually in the range of 20-40% (see figures 4 and 5).

Our algorithm is fairly robust with respect to λ. In the simulation results, the 3 different values $\lambda = 0.05, 0.1, 0.5$ give comparable estimate errors. Several observations are in order, however. When the distance estimates are perfect, smaller λ gives more accurate results, although the algorithm takes more iterations. When the distance estimates are noisy, it appears that the hidden error assert stronger force. Hence, larger λ appear to be more effective. The best value of the regularization parameter λ can be found using cross-validation simulation, given the number of nodes and size of the network.

3.3 Localization algorithm for distorted networks

As discussed in the calibration section, in a distorted network the distance estimates based on RF signal strength tend to overestimate short distances and underestimate long distances. Given a relatively densely connected network (e.g. greater than 50% connectivity), the line parameters (such as those in table 3) can be estimated. First, between anchor nodes, and then between nodes that have been localized. To estimate these parameters, be just perform a simple regression

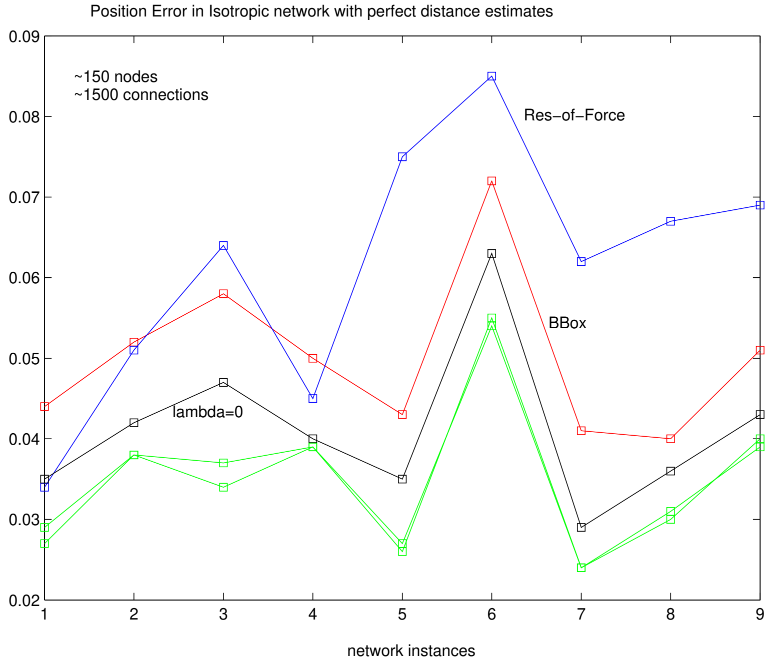

Figure 4: Average position estimate error for non-distorted network with perfect distance estimates. We compare our algorithm with 3 different values of λ (0, 0.05, 0.1) and the Bounding Box, and the Resolution of Force algorithm.

between true distances (as measure by locations) versus the estimated distances (as measured by RSSI conversion). Once the line parameters have been learned, the linear relationship between estimated and true distances can be inverted and distance estimates can be improved by rescaling. Given these improved distance estimates, we can apply the non-distorted network assumptions and use our previously described algorithm.

3.3.1 Simulation results for distorted network

Because of the small number of nodes in our real data sets (16 and 24) we required some simulated data to test out our algorithm for distorted networks. To generate simulation data, we perform the following steps:

1. Randomly generate nodes inside of a rectangular region.
2. Randomly connect nodes iid with probability p (we chose p to be 0.75).
3. Randomly choose initial anchor nodes iid with probability q (we chose q to be 0.1).
4. Randomly choose line parameters that satisfy: slope $<$ 1 and intercept $>$ 0.
5. For connected nodes, set the distance estimations between nodes to be: (slope)·(true distance) + (intercept) + (gaussian noise).

This simulation procedure gave rise to the data which Bounding Box operated on in figure 3. Applying our line parameter learning to improve distance estimates first, and then applying our algorithm from the non-distorted section, we get the results in figure 6.

Dataset	BBox	Res-of-force	$\lambda = 0$	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.5$
1	0.120	0.209	0.111 (9)	0.106 (8)	0.101 (8)	0.094 (4)
2	0.114	0.206	0.099 (8)	0.092 (8)	0.092 (7)	0.100 (4)
3	0.136	0.278	0.131 (7)	0.127 (7)	0.124 (7)	0.114 (4)
4	0.099	0.154	0.080 (9)	0.076 (9)	0.073 (9)	0.073 (5)
5	0.090	0.216	0.083 (6)	0.080 (6)	0.077 (5)	0.071 (4)
6	0.135	0.513	0.128 (7)	0.123 (7)	0.117 (7)	0.096 (5)
7	0.107	0.336	0.096 (8)	0.093 (7)	0.091 (7)	0.090 (4)
8	0.100	0.293	0.084 (9)	0.081 (8)	0.079 (8)	0.084 (4)
9	0.122	-	0.092 (12)	0.086 (11)	0.084 (10)	0.087 (5)

Table 6: Error of position esimates in a non-distorted network with Gaussian distance estimate error with standard deviation 0.05. The same network topology are used as in Table 5. (-) presents some numerical error in the algorithm. The corresponding plot of this table is Figure 5.

Dataset	Estimated Dist. Standard Dev.	Avg. TE (m) Bounding Box	Avg. TE (m) Our algorithm	Avg. TE (m) Force-resolution
1	15	4.061	.1308	2.396
2	30	5.617	.1704	3.948
3	45	6.716	.3696	5.148
4	60	5.579	.8592	3.662
5	75	5.809	.5826	4.843
6	90	4.540	.4867	3.055
7	105	6.530	2.095	5.531
8	120	5.584	.8403	2.785
9	135	4.984	1.713	2.991
10	150	5.402	1.728	3.058

Table 7: Algorithm results on distorted simulation data. All data was generated according to the procedure described above, with the indicated standard deviations applied to the estimated distances.

Figure 6 makes our algorithm look amazing. However, there are test cases that it still has trouble with (e.g. test set 7 in Table 7 where the average absolute location error was 2.095m). Nevertheless, Table 7 shows that in this distorted network setting, the Bounding Box method is extremely inaccurate. The Resolution of Force reduces Bounding Box's by half, while our algorithm improve the error of Resolution Force from 10 upto 20 times!

For comparison, we also include figure 7, which is the result of the force-resolution algorithm on the same data that was inputted in figure 3 and in figure 6. To be fair, however, our algorithm is built to handle the error distribution described in the calibration section whereas the force-resolution algorithm is not.

4 Conclusion and Future Work

This paper described two contributions to the problem of RF-based localization in ad-hoc sensor networks. First, we reveal and analyze the nature of RSSI-based distance estimates. Some inter-

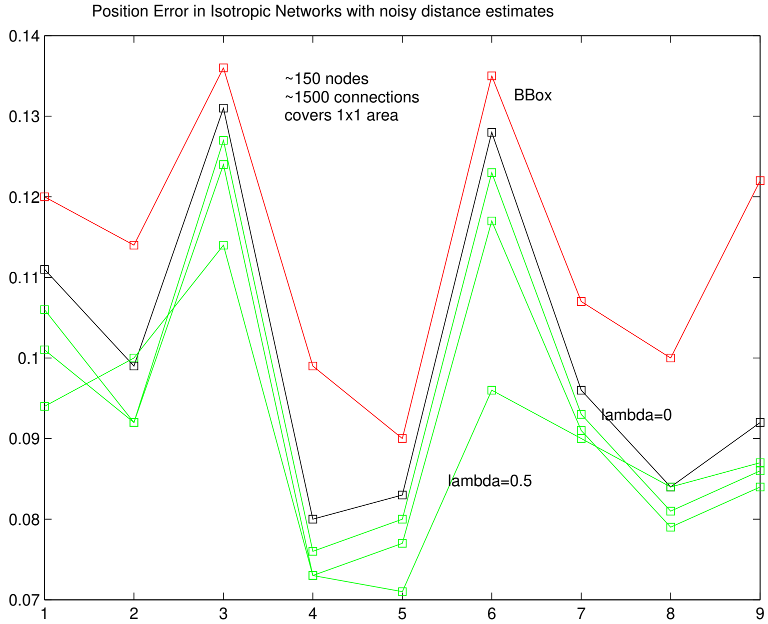

Figure 5: Average position estimate error for non-distorted network with noisy distance estimates. We compare our algorithm with 4 different values of λ (0, 0.05, 0.1, 0.5) and the Bounding Box algorithm. The error of Resolution of Force algorithm is about more than twice worse than the Bounding Box algorithm in this data set and is not ploted here.

esting characteristics of RSSI-based distance estimation are its consistent overestimation of short distance, its consistent underestimation of long distances, and its relatively constant variance in distance estimates regardless of the true distance. Second, we provide a well-founded justification for existing algorithms as well as for our new algorithm (which incorporates regularization to handle error in hidden information). This justification is based on maximum likelihood estimations of observed distances between nodes.

There are interesting issues that remain in this problem domain. Most importantly, more data needs to be generated from real ad-hoc networks. Although the data provided by Kamin Whitehouse was enough for this project, a more in-depth analysis of how RSSI varies with true distance would be facilitated by non-grid experiments. Also, a complete description of what objective functions were being optimized by existing algorithms would be helpful in understanding why they work, and why they do not work.

Ackowledgements

We are indebted to Kamin Whitehouse for providing a huge foundation of data and code for our disposal.

References

[SI02] S. Simic, "A distributed algorithm for localization in random wireless networks", Submitted to Discrete Applied Mathematics, 2002.

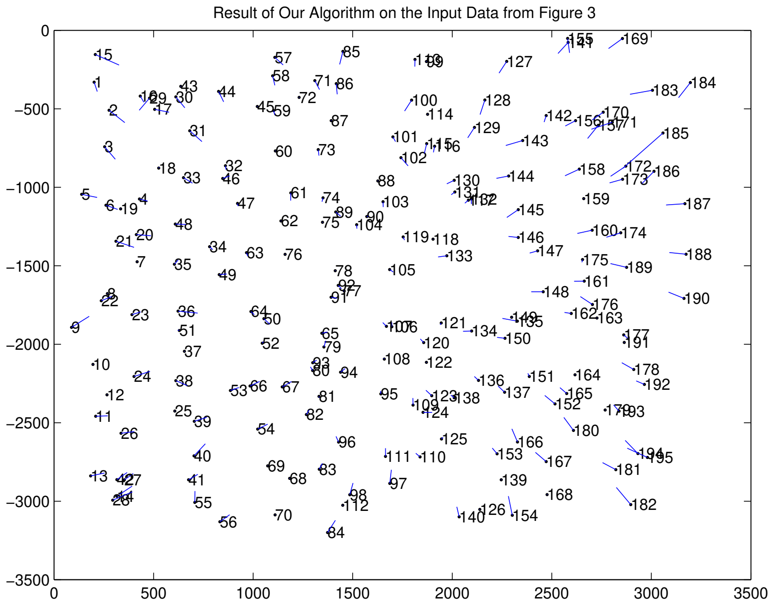

Figure 6: This data is the result of our algorithm applied to the data from Figure 3. Clearly, our algorithm overcomes the errors that plagued the bounding box algorithm. This data corresponds to test set five in table 7. Nodes are plotted in the same fahsion as in figure 3.

[S92] Seidel, A. and Rappaport, T. 1992. "914MHz path loss prediction models for indoor wireless communications in multi-floored buildings", in IEEE Transactions on Antennas and Propagation, vol. 40, no. 2.

[B00] Bahl, P. and Padmanabhan, V. 2000. "RADAR: an in-building RF-based user location and tracking system" in IEEE Infocom.

[H01a] Hightower, J. and Borriello, G. 2001a. "Real-time error in location modeling for ubiquitous computing" in Location Modeling for Ubiquitous Computing - Ubicomp 2001 Workshop Proceedings.

[G01] Girod, L. and Estrin, D. 2001. "Robust range estimation using acoustic and multimodal sensing" in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.

[S01] Savarese, C., Rabaey, J., and Beutel, J. 2001. "Locationing in distributed ad-hoc wireless sensor networks" in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[S02] Savarese, C., Rabaey, J., and Langendoen, K. 2002. "Robust positioning algorithms for distributed ad-hoc wireless sensor networks" in USENIX Technical Annual Conference, Monterey, CA.

[D01] Doherty, L., Pister, K., and El Ghaoui, L. 2001. "Convex position estimation in wireless sensor networks" in Proceedings of IEEE Infocom.

[W97] Ward, A., Jones, A., and Hopper, A. 1997. "A new location technique for the active office" in IEEE Personnel Communications.

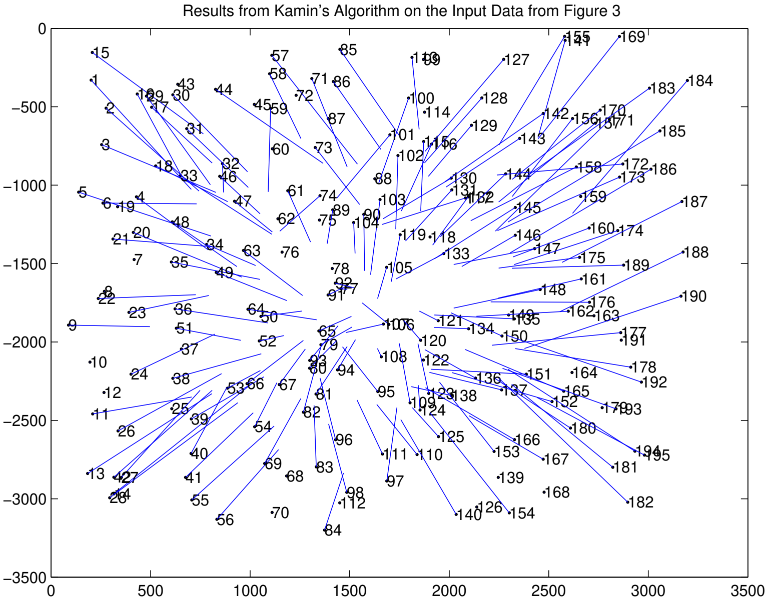

Figure 7: This data is the result of force-resolution algorithm applied to the data from figure 3. Nodes are plotted in the same fashion as in figure 3.

[P00] Priyantha, N., Chakraborty, A., and Balakrishnan, H. 2000. "The cricket location-support system" in ACM International Conference on Mobile Computing and Networking, Boston, MA.

[SA01] Savvides, A., Han, C. and Srivastava, M. 2001. "Dynamic Fine Grained Localization in Ad-hoc Sensor Networks", in Proceedings of Mobicom 2001.

[SA02] Savvides, A., and Srivastava, M. 2002. "Distributed Fine-Grained Node Localization in Ad-hoc Networks." Submitted to IEEE Transactions of Mobile Computing.

[W02] Whitehouse, C. 2002. "The design of Calamari: an Ad-hoc Localization System for Sensor Networks." Master thesis, EECS UC Berkeley.

[H01b] Hightower, J. and Borriello, G. 2001b. "Location systems for ubiquitous computing," in IEEE Computer.

[B00] Bulusu, N., Heidemann, J., and Estrin, D. 2000. "GPS-less low cost outdoor localization for very small devices" Technical report 00-729, University of Souther California, CS Department.

[L97] Lu, F. and Milios, E. 1997. "Globally Consistent Range Scan Alighent for Environment Mapping."