Optimism in Localization

Kamin Whitehouse Computer Science
387 Soda Hall #1776
UC Berkeley
Berkeley, CA 94720
Class Project
CS294, Deeply Embedded Wireless Networks

December 16, 2003

Abstract

This paper presents the experience of a real deployment of Calamari, an ad-hoc localization system for sensor networks. It provides a practical evaluation of many ideas from the literature, serving as a proof-of-concept in some cases and revealing tacit assumptions or unexpected problems in others.

1 Introduction

An Ad-hoc localization system can position nodes without requiring each one to be in contact with at least three anchor nodes. In the past several years, many ad-hoc localization algorithms have proved effective in simulation [6, 9, 8, 10] and several ranging technologies have been empirically shown to provide reasonable distance estimates [7, 9, 11, 5]. However, complete systems that successfully combine the two are scarce. This paper first demonstrates this disconnect by using real ranging data in a simulation to show that results are worse than those obtained with simulated ranging data. Then, the concept of *effective range* is proposed to explain this phenomenon and methods of altering the effective range are discussed.

2 Empirical Ranging Measurements

The ranging estimates were collected using ultrasound technology developed for Calamari [10], an ad-hoc localization system designed specifically for sensor networks and based on the Berkeley mica hardware platform [4]. Ultrasound is the default ranging technology for those localization systems that do exist [7, 9]. It was chosen in this experiment because radio signal strength [3] does not match the noise requirements assumed by the majority of existing ad-hoc localization algorithms and RF time of flight [5] is not practical to implement on a small scale ad-hoc network.

2.1 Ultrasound Ranging

The ultrasound ranging board uses a 45 degree reflective cone was placed above each ultrasonic transducer to transform its normal 20 degree conical emittance pattern into a 10 degree radial emittance. This board used a combination of a Prowave 25KHz ultrasonic transducer [1] and an Atmel Atmega8 1MHz microcontroller. The output of the transducer was wired to the analog comparator on the microcontroller for detecting a signal via a simple threshold, which could be controlled in software. The input of the transducer was wired to a PWM line on the Atmega8, which directly keyed the 25KHz signal through software. Both devices resided in a daughter board that was mounted on a Mica2Dot, which conists of a ChipCon CC1000 FSK 433Mhz radio and a Atmel Atmega128 4MHz microcontroller. The two microcontrollers could communicate through a UART interface and a single PW line that was wired from one of the Atmega128's PWMs to a hardware interrupt on the Atmega8 [2].

When the Atmega128 wanted to transmit, it would send a UART signal with a *transmit* command to the Atmega8. This would put the Atmega8 in transmit mode. When the Atmega128 sent a radio packet, it triggered an interrupt on the Atmega8, which signalled it to send immediately. After transmission, the Atmega8 went back into *receive* mode. In this way, the ultrasound signal was coordinated exactly with the sending of a radio message. Notice that it was also always enveloped in a radio message to prevent collisions, as demonstrated in [7].

When the Atmega128 was receiving a radio message, it would trigger the same interrupt on the Atmega8. Since the Atmega8 is by default in receive mode, this triggered the Atmega8 to enable the analog comparator and set the Timer0 value to zero. If an ultrasound signal was detected, the Timer0 value was read again and sent over the UART to the Atmega128. Thus, a the

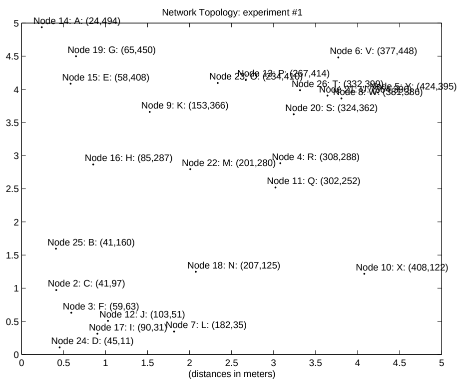

Figure 1:

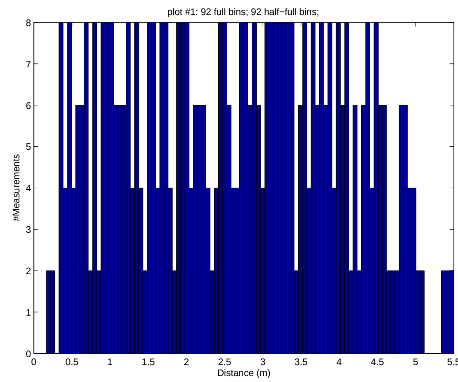

Figure 2:

receive time of the radio message and the ultrasound message was coordinated via a hardware interrupt. After the signal is detected, of if no ultrasound signal was detected, the Atmega8 again disabled the analog comparator.

2.2 Experimental Setup

Most ad-hoc localization algorithms use simulated ranging data instead of real, empirical ranging measurements, and for a good reason: to have robust results, simulations must average results over many random topologies. However, each random topology will have different distances between the different nodes. It is much easier to interpolate what a ranging technology *would probably* generate at each of those distances than it is to actually measure them for each random topology.

In order to generate ranging estimates that could be reused in any simulation, a special topology was generated and used to collect distance estimates once. The topology, shown in Figure 1, was generated using rejection sampling by generating random topologies until a histogram of measured distances included at least two measurements in each ten centimeter bin. The histogram for this topology is shown in Figure 2. The random topology therefore measured all distances between .7m and 5m with a spacing of on average 3cm, which is less than the nominal errors that were observed. Because every possible distance is measured, these distance estimates can then be reused in a simulation of any random topology.

Of course, the measurement of each distance is affected by the idiosyncrasies of the pair of nodes that was used to measure it. To ameliorate this effect, distance estimates were taken five times, each time with a different random mapping of nodes onto the topology locations. Ten measurements were collected over each distance at each mapping. Since each distance appears in the topology at least twice, each one was estimated with at least ten different pairs of nodes and was sampled about one hundred times.

This entire routine of experiments was then repeated four times in four different environments: on carpet, in the grass, about 30 centimeters above the grass, and on pavement, for a total of at most 400 samples at each distance. All resulting distance estimates generated during this experiment are shown in Figure 3, after uniform calibration [11].

2.3 Calibration and Filtering

The noise in the raw ultrasound ranging data is unacceptable, even after calibration. This can be fixed by using all ranging estimates between a

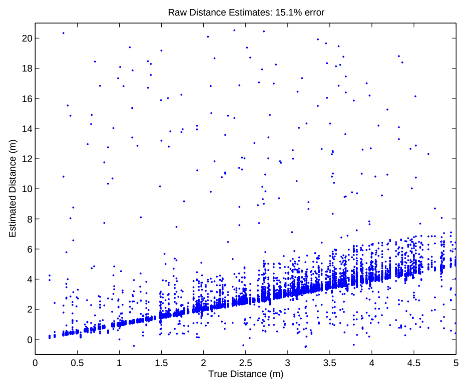

Figure 3:

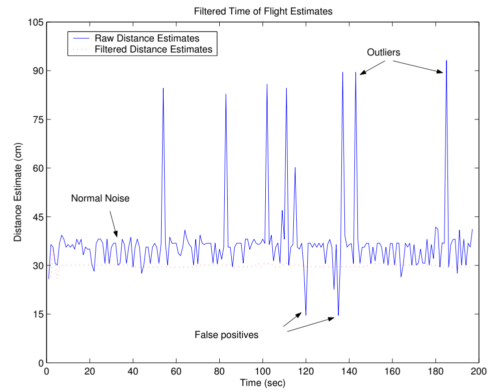

Figure 4:

pair to eliminate outliers and better estimate the true distance. Because of the nature of the noise in an ultrasound ranging estimate, a non-linear filter must be used.

First of all, we notice that there are many outliers. This can be due to the transducer being in an undefined state during the reception, or random noise in the signal, such as automobile traffic or a door slamming. The easiest way to eliminate these outliers is to remove all readings that fall outside of some range of the median. In this case, the range was chosen to be about 20 centimeters, but it could be varied with the value of the median. This non-linear filter is called *medianTube*. Second, we notice that all readings within the medianTube are greater than or equal to the true distance due to the nature of time of flight. We therefore choose the minimum value within the median tube.

A sample of the filter at work on a real data sequence (from audible range data) is shown in Figure 4. The resulting error estimates are much better than those without filtering, and are shown in Figure 5. As one can see, this data has a low error rate of 3.5% average error. As shown in Figure 6, this error is also gaussian distributed. The error rate changes, though not significantly, between different environments. Figure 7 shows the error rate for each environment as it changes

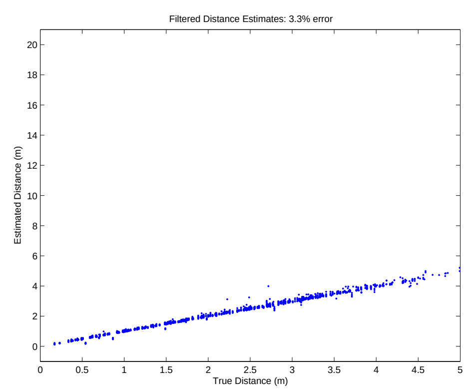

Figure 5:

with distance.

3 Localization Simulations

While many ad-hoc localization algorithms have been proposed, this paper evaluates the *DV-distance* distance-vector based algorithm [6] because of its simplicity and influence on other algorithms [10, 8]. The algorithm works by having each node estimate its distance to each base station to be the shortest path distance to that base station. This can be done simply using a distributed distance vector algorithm. Once the node has at least three of these distance estimates, they can be used with multi-lateration, a standard least-squares method of estimating position given distance estimates to three known positions.

To evaluate this algorithm, 100 random topologies in a 15m x 15m area were generated at each of ten densities ranging from 0.07 nodes/meter to 0.751 nodes/meter. Simulated ranging estimates with 5 meter maximum range and 3.5% gaussian noise were used, the same properties as the empirical results found above. These simulated ranging estimates were generated by choosing all distances d less than or equal to 5m and purturbing the value with gaussian noise $N(0, d*0.035)$. All pairs of nodes greater than 5m apart were considered disconnected.

To incorporate the emprirical ranging estimates into the simulation, each distance estimate d less than or equal to 5m was paired with a distance in the topology shown in 1. All distances were paired with other distances less than 3 centimeters difference. Then, the distance estimate for one random pair that was used to measure that distance was chosen and inserted into the simultion. This procedure was repeated for the data collected in each of the four environments: grass, free space, pavement, and carpet. For comparison, results with simulated ranging estimates with 3.5% gaussian noise and maxmimum ranges of 4m, 3.5m, 3m, 2,5m, 2m, and 1m were also plotted. The results are all shown in Figure 8 (top).

Clearly, the empirical data is not performing as well as the simulated data with the same characteristics of 5m maximum range and 3.5% gaussian error. In fact, most of the empirical data seems to be performing around that quality of 3.25m. The grass data is the exception, which performed extremely badly, more closely approximating simulated data of about 2m. This is surprising not only because of the bad performance, but also because of the trend in bad performance with environment; Figure 7 shows that the worse error rates were in fact gotten in the carpet environment, not the grass environment.

4 Effective Range

To understand why empirical ranging estimates do more poorly than simulated estimates, Figure

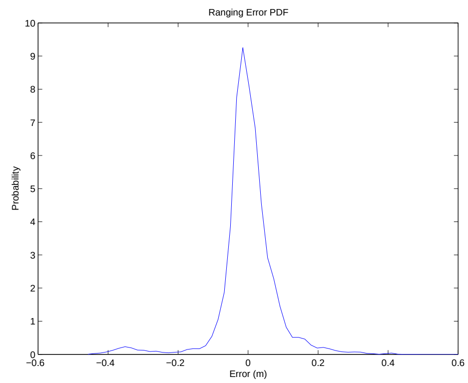

Figure 6:

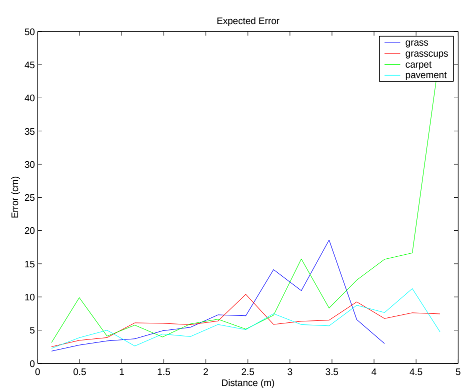

Figure 7:

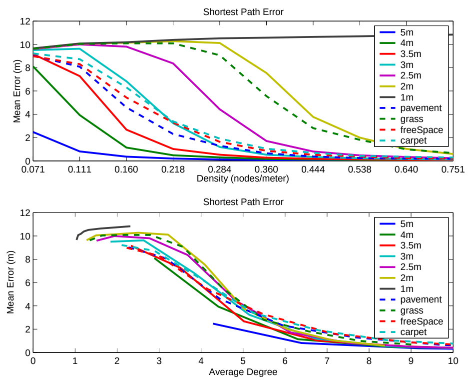

Figure 8:

8 (bottom) re-plots the data from Figure 8 (top) against average node degree. The average node degree is the average number of nodes to which each node has a distance estimate. Surprisingly, this new plot shows that all types of ranging estimates behave similarly given node degree, indicating that node degree must *not* be the same between the empirical ranging estimates and the simulated ranging estimates.

Figure 9 shows the probability of a ranging success between two nodes given the distance between them. Indeed, this graph explains why the average degree given empirical estimates is lower than with simulated estimates; many distance estimates fail in a real deployment and these failures dramatically decrease degree. Unfortunately, none of the ad-hoc localization simulations in the literature [9, 8, 6] vary failure rate as one of the ranging characteristics.

The failure rate of ranging is indeed somewhat surprising even in itself, especially given the fact that most ranging technologies are characterized in terms of *range*. As we can see in Figure 9, *range* is actually ill-defined; similar to radio-range, ultrasonic range is probabilistic and no clear point-estimate of range exists.

One remaining question is, given that the carpet and the grass data seem to have roughly the same probability mass, what explains why the grass data did so poorly while the carpet data did nearly as well as the other data? The answer, lies in how this probability mass is converted into node degree. Assuming the neighbor positions are roughly uniformly distributed with density D, the average node degree can be calculated with

$$\int_{r=0}^{5} \Pi * r^2 * p(\text{ranging}|r)$$

with we will call the *effective range*. Figure 10 is a diagram illustrating the effective range of the pavement and grass data, showing that the covered areas are significantly smaller than the original estimate of 5m.

One further observation is that the carpet data is not significantly worse than the pavement or free space data despite its smaller probability mass. This can be explained by the non-uniform probability distribution of neighbor distances; almost no neighbors are right next to each other. Therefore, the effective range could be calculated even more accurately by estimating $p(n|r)$, the probability of having a neighbor at distance r and updating the equation above to be

$$\int_{r=0}^{5} \Pi * r^2 * p(\text{ranging}|r) * p(\text{neighbor}|r)$$

which might be called the weighted effective range.

4.1 Effective Range and Filtering

Effective range is a useful way of characterizing ranging technologies such that localization simulations can be compared with each other and projected onto real empirical results. However, the knowledge that loss rates have such an extreme effect on localization results also suggests that we improve results greatly not by reducing average error but by increasing the probability mass in Figure 9. One way to do that is to modify the filters presented in Section 2.3.

Figure 11 shows the effect of various filter parameters on the effective range (y-axis) and error (x-axis). The lower-left and top-right graphs are most easily understood: as the minimum number of readings accepted is increased in the lower-left graph, the mean error gets smaller because we have more samples and are more confident of our result. However, the effective radius goes down because we are eliminating valid ranging estimates that simply do not have enough samples. As the number of samples initially collected is increased in the top-right graph, the mean error goes down while the effective radius increases. Thus, obtaining more initial samples boosts ranging overall, although not by very much. The top-left graph is most difficult to interpret. Given a particular value for minReadings and filterLength, as the width of the median tube increases, both mean error and effective range increase.

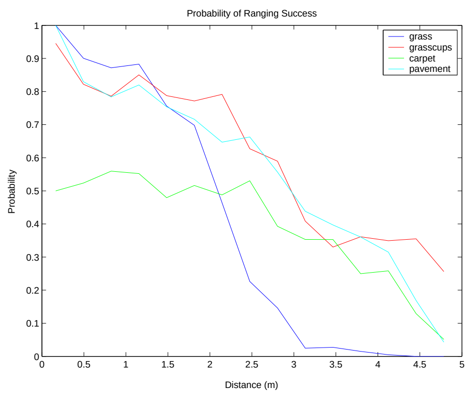

Figure 9:

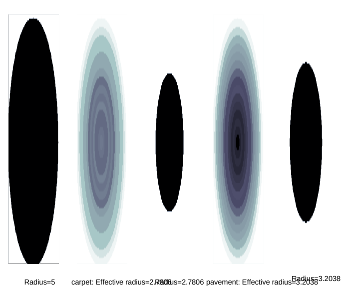

Figure 10:

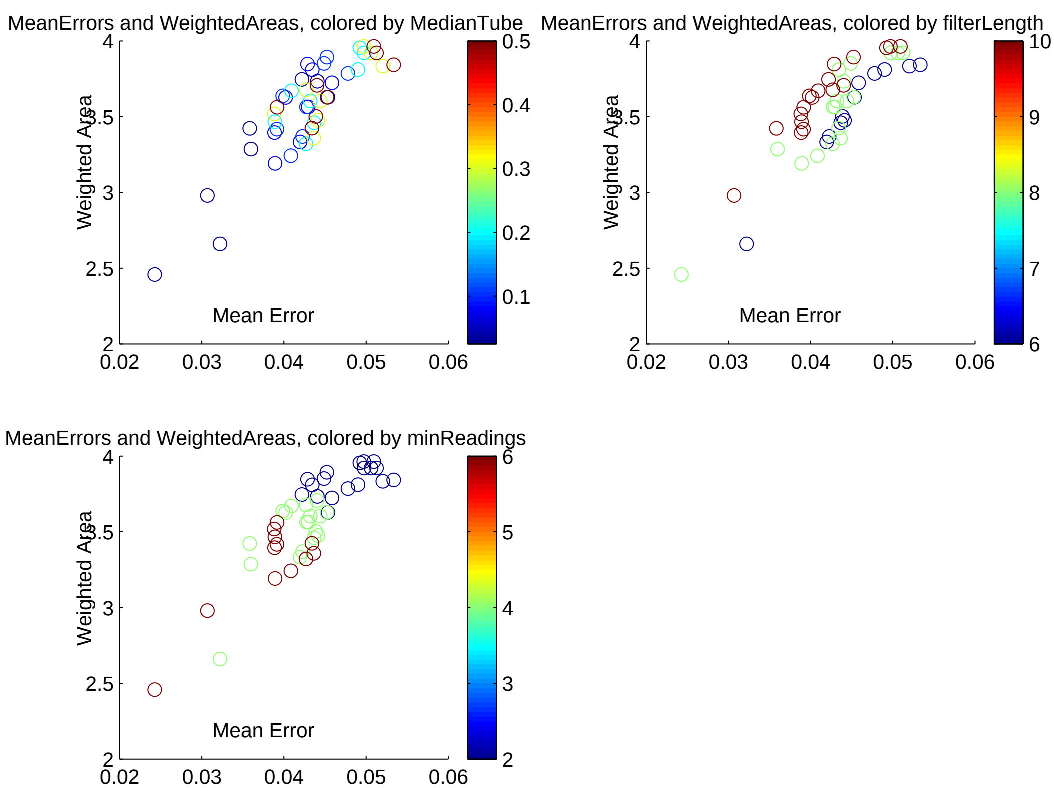
MeanErrors and WeightedAreas, colored by MedianTube
MeanErrors and WeightedAreas, colored by filterLength
MeanErrors and WeightedAreas, colored by minReadings
Weighted Area
Mean Error
0.02
0.03
0.04
0.05
0.06
2
2.5
3
3.5
4
0.1
0.2
0.3
0.4
0.5
6
7
8
9
10
2
3
4
5
6

Figure 11:

Finally, we notice that the mean error doesn't change significantly no matter what our filter characteristics are, although our effective radius can vary by up to 2.5 meters. Thus, we choose parameters that give us the highest possible effective radius with the lowest mean error, which are: filterLength=10, minReadings=2, and medianTube=0.2. Surprisingly, the median tube is extremely narrow, probably because the minReadings variable is so low.

5 Conclusion

We identified a property of ranging technologies called *effective range* and showed that the effect of effective range is much greater than the effect of mean error. Effective range should therefore by included in the tests for algorithmic robustness in localization simulations. We also show that there is in fact a trade-off between mean error and effective range, and that modifying our filter can help us achieve a good balance of both.

Acknowledgements

This work is funded in part by the National Defense Science and Engineering Graduate Fellowship, the DARPA NEST contract F33615-01-C-1895, and Intel Research.

References

[1] Air ultrasonic ceramic transducers. http://www.prowave.com.tw/p1.htm.

[2] Wireless sensor networks. http://www.xbow.com/Products/Wireless_Sensor_Networks.htm.

[3] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, March 2000.

[4] Jason Hill and David E. Culler. Mica: a wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24, nov/dec 2002.

[5] B. Hofmann-Wellenho, H. Lichtenegger, and J.Collins. *Global Positioning System: Theory and Practice*. Springer Verlag, fourth edition, 1997.

[6] Dragos Niculescu and Badri Nath. Ad Hoc Positioning System (APS). In *GLOBECOM (1)*, pages 2926–2931, 2001.

[7] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Mobile Computing and Networking*, pages 32–43, 2000.

[8] Chris Savarese. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. Master's thesis, University of California at Berkeley, 2002.

[9] A. Savvides, H. Park, and M. B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *First ACM International Workshop on Sensor Networks and Applications*, Septmber 2002.

[10] Kamin Whitehouse. The design of calamari: an ad-hoc localization system for sensor networks. Master's thesis, University of California at Berkeley, 2002.

[11] Kamin Whitehouse and David Culler. Calibration as Parameter Estimation in Sensor Networks. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, USA, September 2002.