Experiments in Monte-Carlo Localization using WiFi Signal Strength

Sajid M. Siddiqi, Gaurav S. Sukhatme and Andrew Howard

Robotic Embedded Systems Laboratory
Center for Robotics and Embedded Systems
Department of Computer Science
University of Southern California
Los Angeles, California, USA
{siddiqi,gaurav,ahoward}@usc.edu

Abstract

We study the problem of globally localizing a mobile robot in a structured environment given an 802.11b card on the robot, one or more 802.11b access points (APs) in the environment and a signal-strength map. The method for acquiring this map is described. A particle filter-based algorithm is implemented with an odometry-based action model and an observation model based on the signal-strength map. Tests carried out using a Pioneer 2DX robot with a wireless card show promising results. Localization is accurate a significant percentage of the time. We test the algorithm with one, two and three APs for observations and show empirically that improved results are obtained with a higher number of APs.

1 Introduction

We study the problem of localizing a mobile robot with an 802.11b card as the only sensor used in the observation model. We assume that a signal strength map of the environment is available and describe a method for acquiring and representing such a map. We assume that one or more static wireless access points (APs) are placed in the environment. As the robot moves through the environment, it measures the signal strength from the APs and compares the results to its signal-strength map. A probability density function is used to represent the location of the robot. This belief function associates a probability mass with each value of the possible robot pose (x, y, θ), a standard technique [5] in robot localization. Each time the robot moves or senses the signal strength of an AP, a Bayes filter [5] is used to recursively update the belief function.

The non-linear, fluctuating nature of radio signals, and their sensitivity to interference, dead spots and reflection, are well known [1, 2, 3]. In a structured environment such as our test area, multi-path fading is also an issue [4]. Thus radio propagation is complex enough to render analytic models impractical. The observation model must be built empirically, by sampling from the environment.

Probabilistic techniques such as the Bayes filter [5] are an attractive option to cope with the uncertainty inherent in sensing and action, and allow us to perform global localization with no initial estimate of robot pose.

In the present work, we choose to represent the belief function associated with the robot's pose by samples (or particles). A particle filter was chosen because of its ease of implementation [6] and fast real-time properties. The mathematical basis for our implementation of the particle filter is standard [6]. We discuss it briefly in the next section.

After the initial data-gathering for building the observation model, we collected four data sets for testing the filter. Because of the way the observation model works, it is easy for us to simulate cases with fewer APs visible than actually exist, once we have a set of data for all APs. After running the filter on the data sets for a number of different AP visibility settings, we examined the average error statistics as a function of time, and as a function of the number of visible APs. We obtained reasonable results for localization accuracy given the vagaries of wireless propagation. The results were revealing about the behavior of the filter and the effect of changes in AP visibility settings.

Our results show that accurate localization ($\sim$ 2 m) is achieved in most test cases. Average localization error decreases with time, and the degree of clustering of the particles increases with time, as expected. We examined localization error as a function of the number of APs and found that the error decreases as the number of APs increases.

2 Related Work

The problem of mobile robot localization has been studied in great depth. Research focusing on single-robot localization with prior map knowledge includes [7, 5]. The problem of wireless signal-based localization indoors has been recently studied [8, 9] and is becoming particularly relevant given the recent proliferation of WiFi technology (particularly 802.11b). It is interesting to note that such technology could also be used to localize people using

WiFi connectivity in a building. In fact, the robot localization problem is easier than the people-localization problem since it is possible to easily build a motion model for robots, and to equip them with motion sensors, which can use such a model.

Previous attempts in WiFi-based localization have used data readings to model signal strength observations [8] as well as more general models that use the geometric properties of the indoor environment [9] together with knowledge about the wireless card and AP hardware characteristics. Our approach is similar to the approach in [8] in the use of a Bayes filter, and the use of prior knowledge of signal-strength maps for all APs to construct the observation model. However, we incorporate an action model into the localization estimate based on encoder data from the robot (in [8] a human operator carried a laptop equipped with a 802.11b card). We also explore the relationship between localization error and the number of APs used in the observation model.

Our method is similar to one of the empirical methods studied in [9] for tracking wireless device users in a building. However in [9], no Bayes filter was used, and localization was performed by triangulation.

3 Approach

We examine the problem of single robot localization using wireless signal strength as the only sensor available on the robot. Prior signal strength data are gathered on which the observation model is based. Standard action model and observation model update equations are used, and are described briefly in the next two sections. The robot starts with no idea about its location and moves about the environment guided by a human controller. The localization computation is done offline in the experiments reported here, but is fast enough for the technique to be adapted for online use.

3.1 Assumptions

We make a number of assumptions in our approach. The environment is assumed to be static with sparse activity if any. Wireless AP signals in any given area of the environment are assumed to be time-invariant in the absence of interference and signal absorption by humans. We assume there are no large variations in signal strength over small distances. The robot is assumed to move in a smooth, continuous manner and is constrained to be somewhere in the map. Observations are assumed to be independent. We make the Markov assumption, which allows the Bayes filter to operate. In the light of these first-order assumptions, most of which are violated by the system, the results from experiments are especially encouraging.

3.2 The particle filter

The particle filter is a *Monte Carlo localization (MCL)* algorithm [10, 11]. It is an easily implemented, efficient and fast representation of the Bayes filter, suitable for any form of the localization problem, regardless of the initial assumptions and pose estimates. Part of the reason is the particle filter's flexibility compared to the Kalman filter [12] (the other commonly used representation of the Bayes filter). The Kalman filter can represent normal distributions whereas the particle filter can approximate a large range of possibly multi-modal probability distributions. The essential idea of a particle filter is to approximately represent the belief function by a set of weighted samples distributed according to the belief function. The reader is referred to [6] for details.

4 Action Model

We use an odometry-based action model. In each (small) timestep for which encoder data are available, the robot is assumed to take an action (make a small motion). Let a denote the action, and s and s' denote the state of the robot before and after the action respectively. The output of the action model for each timestep is the probability $P(s'|s,a)$ that the robot is in state s', as a consequence of executing action a in state s. We compute this probability by assuming that the action a is notionally split into separate rotation and translation components, each of which is perturbed by Gaussian noise. We sample from the action model at every iteration of the filter. We note that the odometry of the robot we used is very good, with little translational bias and a very slight positive rotational bias. We exploited this fact to obtain ground truth from odometry (over small motions on the order of a few meters) as described later.

We found a need to add more noise to the action model than would normally be necessary given the robot's accurate odometry. This was done to compensate for the lack of directionality in the observation model, which often caused the filter to lose track of the robot's location as it moved. A side effect of the added noise is an orientation accuracy level that is lower than expected, however, the additional noise significantly improves the position accuracy.

5 Observation Model

The observation model plays an especially critical role since it enforces the map constraints as well as outlier rejection. The robot is constrained to be somewhere in the map, so particles that fall off the map can be weeded out. This fact alone enables fairly good localization, even without the use of signal strength. In implementation, probabilities returned by the observation model were augmented with a constant uniform component to avoid the situation where all probabilities slip below the range of the programming language data types being used. This was done to avoid a normalization failure in the filter update. The Bayesian equation for observations for the i^{th} particle which is in state s_i, given the signal-strength map m, and using n APs in the observation model, is given by:

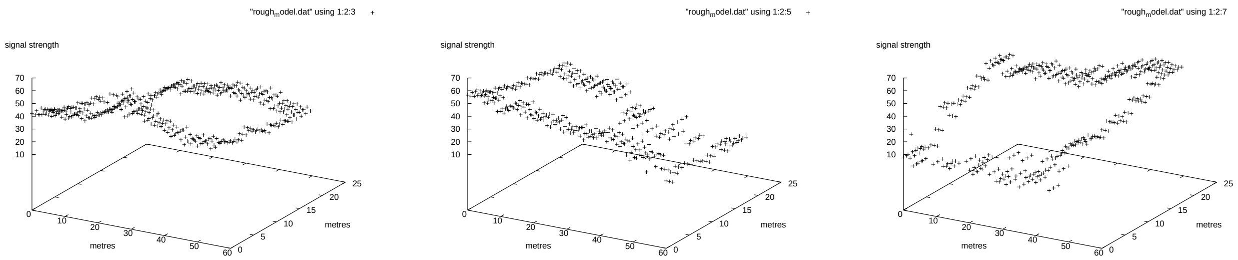

Figure 1: *Mean signal strength measurements of three access points, from which the observation model is derived*

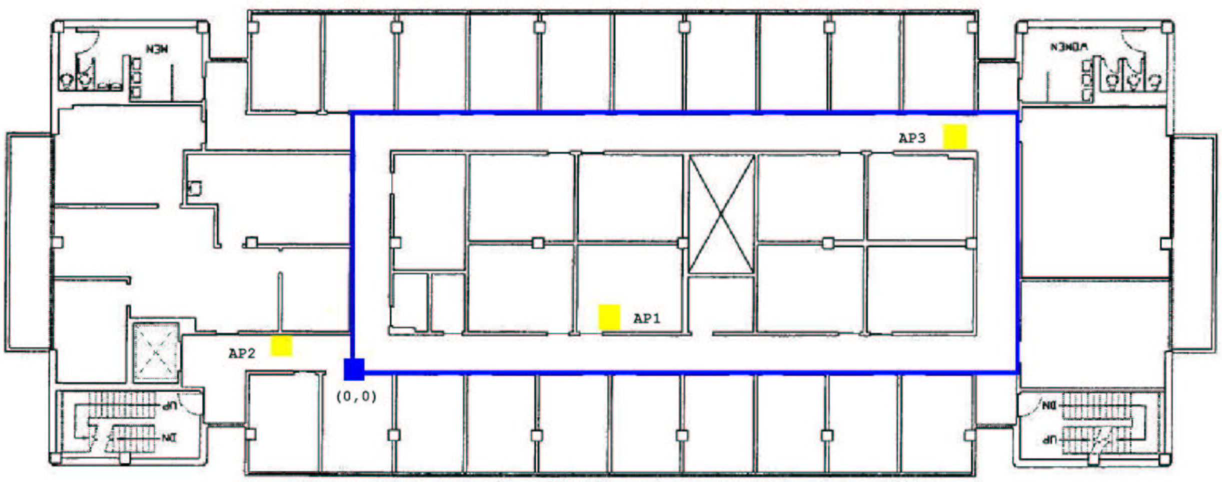

Figure 2: *Salvatori Hall 2^{nd} floor, where the experiments were done. The locations of the access points are shown*

$$P(o|s_i, m) = \prod_{j=1}^{n} (z \cdot P(o_j|s_i, m) + (1-z) \cdot UnifNoise) \quad (1)$$

The constant z describes the relative weights of the two components. The observation model is based on a large amount of signal strength data which were gathered and processed. The environment surface was divided into grid squares 0.5 metres in width. The signal strength for a particular AP in a particular grid square was measured. From these measurements, the mean and variance of the signal strength for each grid square was computed for each AP. Thus for 3 APs, we stored 6 numbers per grid cell. This set of numbers constitute the signal strength map used by the robot. Representative values of the signal strength observations are plotted in Figure 1. The origin is in the lower left corner of the Figure and the x and y axes correspond to the x and y directions in the large rectangle in Figure 2.

Note that robot orientation is of no consequence in the observation model. Despite this, we initialize our particles to random orientations and attempt to solve the (x,y,θ) localization problem rather than just the (x,y) problem. This is justified because of the high degree of map structure : mis-oriented particles have a high likelihood of wandering out of bounds and eventually being weeded out of the sample of particles that represents the probability distribution function.

6 Experimental Design

As explained in the previous section, to acquire the signal-strength map for the observation model, we first imposed a grid on the environment. The robot was then made to roam one full circle around the entire hallway in Figure 2. Signal strengths for every AP were recorded

continuously and tagged with location information. Then, the means and variances of signal strengths for each AP were calculated for each grid square the robot had passed through. Assuming that radio propagation properties do not change over short distances, the signal-strength means and variances for the grid squares the robot had not entered were assumed to be the same as for nearby grid squares which the robot had passed through. In this way a signal-strength map for the entire environment was obtained.

For the actual experiment, four different data sets were gathered. Each data set consisted of signal-strength readings and ground truth obtained while linearly translating the robot anti-clockwise in one of the four hallways in Figure 2. The robot moved approximately 27 metres in each of the longer hallways and approximately 10 metres in the shorter hallways. The data were then processed offline with varying filter settings and the results were recorded. The localization errors in the results were measured and analyzed. The results are discussed below.

For ground truth, we used odometry data gathered during the data collection runs with the robot's initial coordinates factored into the odometry data. We found odometry to be sufficiently accurate for providing ground truth, given the short, linear paths traversed by the robot during the data collection runs. Ordinarily this would not be useful since odometry is not reliable over long traverses. However, by hand-initializing the robot's location, and executing a short traverse (less than 10 m) while gathering signal strength data, stopping, re-initializing the robot's location manually, we obtained a reasonable set of measurements with location ground truth.

6.1 Hardware and experiment configuration

No simulations were used due to the lack of available radio signal strength modeling devices within the simulators available at hand. All data-gathering experiments were carried out on a Pioneer 2DX mobile robot with an on-board Pentium processor running Linux. The robot has many sensors such as laser range-finders, sonar and camera, but only wireless signal strength was used. The test area was a *27.84m* $\times$ *10.68m* rectangular hallway which occupies most of Salvatori Hall's 2^{nd} floor. Three Cisco Aeronet Access Points were set up at widely separated points on the floor. Figure 2 shows the test area with gray blocks for AP locations and a large rectangular box with a black block at the origin delimiting our test area.

6.2 Experimental protocol

Because of the way the observation model works it was easy for us to simulate cases with fewer APs visible than actually existed, once we had a set of data for all APs. After running the filter on the data sets for a number of different AP visibility settings, we examined the average error statistics as a function of time and as a function of the number of visible APs. We ran a total of 60 filter instantiations, with 5 runs per $\{$number of visible APs,dataset$\}$ combination, with the number of APs ranging from 1 to 3, over 4 data sets. Average localization error data were collected.

7 Results

It would clearly be interesting to see how the average particle displacement from truth, and the standard deviation of particle distance from average particle location (which would be inversely related to degree of convergence) vary as the filter iterates on a particular scenario. Another relationship we have plotted with respect to time is the percentage of "correctly oriented" particles, with a $\pm 30°$ window where a particle is correctly oriented. It would also be of interest to know how the errors and deviations changed when one or two of the three APs were deliberately ignored. An illustration of the particles converging during a typical experimental trial is shown in Figure 4. We obtained good results in localization, and the results were very revealing about the behavior of the filter and the effects of changes in AP visibility settings.

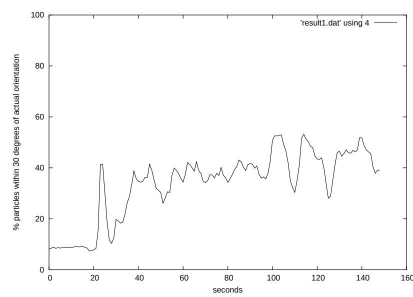

Figure 3: *Percentage of particles within* $\pm 30°$ *of truth, over time*

7.1 Position Accuracy

Figure 5 shows the average localization errors over five runs as plotted against time for data set 1, with 3 APs visible. As we can see, the error drops significantly after a few iterations of the filter but maintains a certain significantly non-zero magnitude. The standard deviation of particle distance from mean indicates the degree to which convergence has occurred : low values indicate high degree of convergence. The mean y-error is considerably lower (approximately 2 m) than the mean-x error (approximately 4 m) simply because the data set is based on a translational movement in the hallway along the x-axis and the y-error is limited by hall width.

The initial dip in average localization error reflects the fact that the robot was initially stationary. It was possible for the particle filter to localize the robot more accurately in this state since the uncertainty introduced by the action model during robot motion was absent during this time.

7.2 Orientation Accuracy

In Figure 3 the interesting thing to note is the relatively low percentage, even at best (around 50%), of correctly

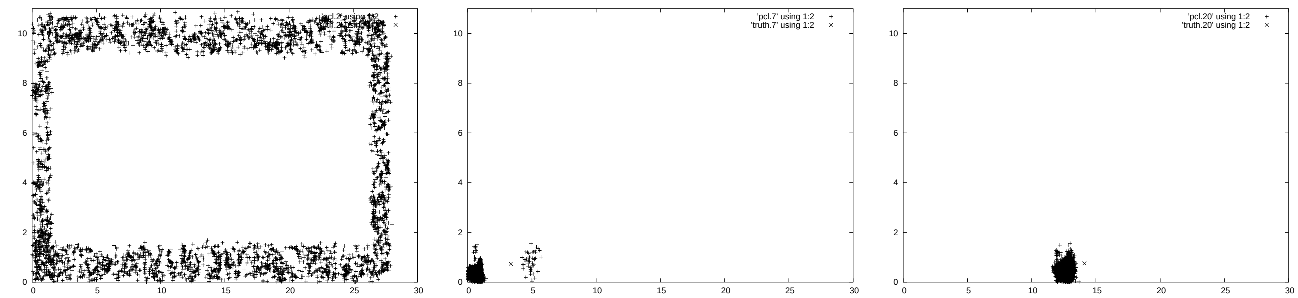

Figure 4: *A set of 5000 particles converging to within a meter of the actual robot location. '+' indicates a particle and 'x' indicates the robot position.*

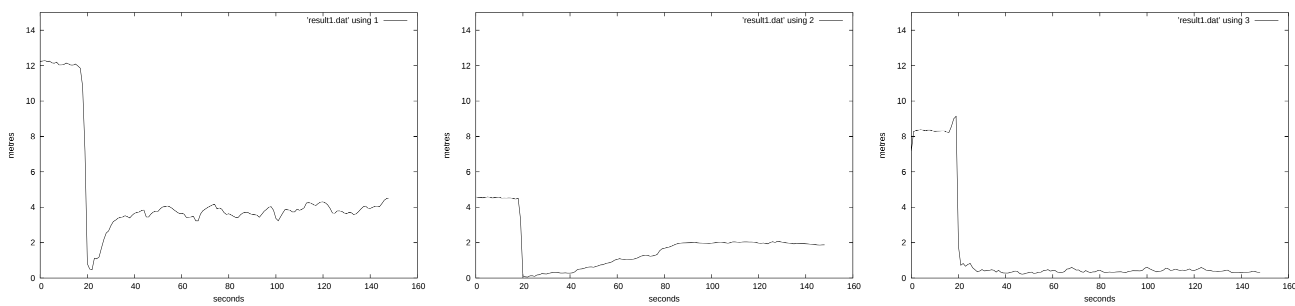

Figure 5: *Average x-error, y-error, closeness of convergence in distance (m) vs. time (s)*

oriented particles that is achieved. Keeping in mind that accurate localization is achieved, and that a $\pm 30°$ window of correctness is allowed for, this percentage is lower than we expected. This is possibly due to excess noise in the action model. However, as noted earlier, this noise helps with the lack of directionality in the observation model.

7.3 Varying the Number of APs

In Figure 6 we graph mean errors in x, y and closeness of convergence for data set 1 over three different AP visibility settings (1, 2 and 3 visible). The relationship here is as expected; localization is significantly better with a larger number of APs being considered. The x-error reduces from approximately 8 to 4.5 m, y-error starts at above 2 metres and declines, and closeness of convergence declines slowly to about 2 m.

However, our expectation is that under conventional Bayes filters such as the one used here, and in test areas such as ours where each AP typically has positive signal-strength readings throughout the environment, this relationship will taper off (or even reverse itself) as the number of APs increases beyond a certain point due to increased symmetry in the signal-strength map. This is a possible avenue for future investigation.

8 Conclusion and Future Work

We conclude based on our experiments that (x,y,θ) localization using wireless signal strength on a mobile robot with an 802.11b card is quite feasible, however, certain refinements are necessary to the observation model and filter settings in order to be more effective. Having a higher number of APs decreases localization error noticeably, at least up until the maximum number of APs we used.

This does not rule out the existence of an environment-specific saturation quantity of APs beyond which localization error rises. We will investigate the possibility of this phenomenon in the future. For other future work, there are symmetry issues to be explored, both symmetry in AP layout as well as symmetry in signal-strength spread, which may not always coincide. Some of the assumptions such as static environment and constant signal-strength characteristics over small distances can be relaxed and the differences in results can be analyzed. In this case we would need either a denser data set on which to base the signal-strength map, or a meaningful method to extrapolate signal-strength value for the entire map based on the sparse data.

Acknowledgments

We would like to gratefully acknowledge Sanjeev Koppal's extensive assistance in data-gathering experiments,

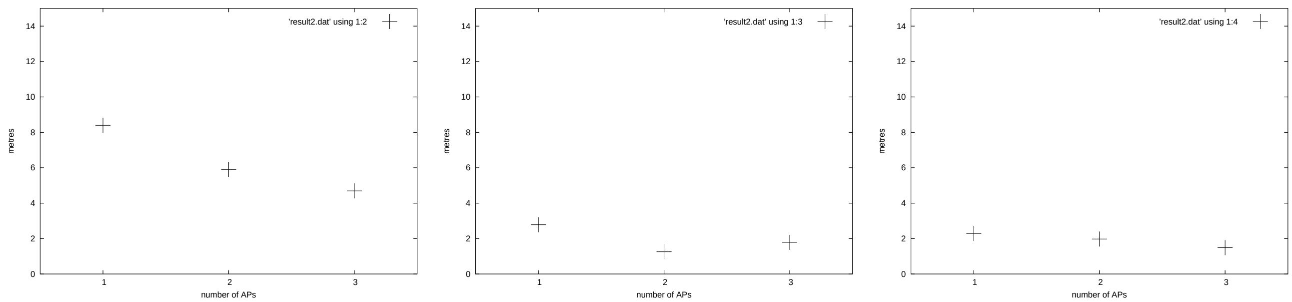

Figure 6: *average x-error, y-error, closeness of convergence in distance (m) vs. number of visible APs*

and Boyoon Jung and other members of the Robotics lab for invaluable suggestions and assistance. This work is supported in part by NSF grants ANI-0082498, EIA-0121141, and IIS-0133947.

References

[1] A.Chakraborty, "A distributed architecture for mobile, location-dependent applications," M.S. thesis, Massachusetts Institute of Technology, May 2000.

[2] T.Cutler, "Wireless ethernet and how to use it," in *The Online Industrial Ethernet Book*, Issue 5. 1999.

[3] A.Neskovic, N.Neskovic, and G.Paunovic, "Modern approaches in modeling of mobile radio systems propagation environment," *IEEE Communications Surveys*, 2000.

[4] H.Hashemi, "The indoor radio propagation channel," in *Proceedings of the IEEE*, July 1993, vol. 81, pp. 943–968.

[5] D.Fox, W.Burgard, and S.Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.

[6] D.Fox, S.Thrun, F.Dellaert, and W.Burgard, "Particle filters for mobile robot localization," in *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[7] J.J.Leonard and Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 2, pp. 1080–1087, 1991.

[8] A.M.Ladd, K.E.Bekris, G.Marceau, A.Rudys, D.S.Wallach, and L.E.Kavraki, "Robotics-based location sensing for wireless ethernet," in *Eighth ACM International Conference of Mobile Computing and Networking (MOBICOM 2002)*, Atlanta,GA, September 2002.

[9] P.Bahl and V.N.Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Proceedings of IEEE Infocom 2000*, Tel-Aviv,Israel, March 2000, vol. 2, pp. 775–784.

[10] F.Dellaert, D.Fox, W.Burgard, and S.Thrun, "Monte carlo localization for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[11] D.Fox, W.Burgard, F.Dellaert, and S.Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, 1999.

[12] R.E.Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME - Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.