Minimizing Calibration Effort for an Indoor 802.11 Device Location Measurement System

John Krumm
John Platt

November 13, 2003

Technical Report
MSR-TR-2003-82

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Abstract

Using an 802.11 wireless client as a location sensor is an increasingly popular way of enabling location-based services. Triangulation on signal strengths from multiple access points can be used to pinpoint location down to a few meters. However, this level of accuracy comes at the price of a manual, tedious, spatially high-density calibration of signal strength as a function of location. This paper presents a new 802.11 location algorithm based on a relatively coarse calibration. This helps answer the question of how accurate location can be computed based on a realistic level of calibration effort. The algorithm uses an interpolation function that gives location as a function of signal strength. As such, it is suited to maintaining some degree of performance in spite of reduced calibration data. We use this feature to test the effect of reducing the number of calibration readings per location and the number of locations visited during calibration. Our experiments show that calibration effort can be significantly reduced with only a minor reduction in spatial accuracy. This effectively diminishes one of the most daunting practical barriers to wider adoption of this type of location measurement technique.

1 Introduction

Knowing the locations of users and devices inside a building is an important prerequisite for location-based services and aspects of ubiquitous computing. Applications including printing on the nearest printer, walking directions, and inferring context for messaging. One promising approach to measuring location is triangulation from 802.11 signal strength on wireless devices. Given radio signal strength measurements on the client from a few different access points, researchers have shown how to compute location down to a few meters. This type of location measurement is especially attractive because it uses the building's and user's existing devices and because it works indoors where GPS and cell phone location often break down.

However, the accuracy of such systems usually depends on a meticulous calibration procedure that consists of physically moving a wireless client to many different known locations, and sometimes orientations, inside a building. It may be unrealistic to expect anyone to spend the resources on such work. When presented with this prospect as part of a new product, software product planners sometimes balk, complaining that system administrators are reluctant to even keep the locations of printers updated, much less create and maintain a high-resolution table of 802.11 signal strengths.

One alternative to manual calibration is to analytically predict signal strengths based on a physical simulation of the building and radio frequency propagation. There is work on predicting signal strengths for wireless networking (*e.g.* [1, 2]), but mostly aimed as a guide to the placement of access points and not location measurement. Bahl and Padmanabhan's RADAR[3] system was one of the first and most comprehensive studies of 802.11 location, and they considered the question of physical simulation versus manual calibration of signal strengths. They discovered, for their chosen simulation method, that physically simulating signal strengths increased their median location error by about 46% (from 2.94 meters to 4.3 meters) over manual calibration. Moreover, a good physical simulation usually requires a more detailed model of the building than is normally available.

While we cannot yet say that manual calibration will always give more accuracy than physical simulation, the evidence thus far suggests so. We can also assume that people's propensity for performing tedious calibration will not grow. Assuming we have to live with manual calibration for the time being, this paper attempts to answer two questions:

1. How much spatial accuracy can we get from an 802.11 location system whose calibration requirements are not terribly tedious?
2. How much can we reduce the calibration effort before accuracy is significantly compromised?

We answer the first question by developing a new 802.11 location algorithm based on the relatively easy calibration procedure of recording signal strengths from one location in every office-sized space in a building. Offices are a natural spatial fiducial in buildings, so our technique does not require measuring points on the floor as long as a floor map is available.

We answer the second question by testing our system with different amounts of calibration data, shortening the time spent at each calibration location and skipping some of the locations altogether.

Our new location algorithm is designed to work in spite of missing calibration data. It takes a set of signal strengths from known locations in a building and builds an interpolation function giving (x, y) as a function of signal strength. For interpolation we use radial basis functions, which are simple to express and compute. We evaluated this new algorithm on one floor of a building with 118 rooms. The rms location error was 3.75 meters using, on average, one calibration point for every 19.5 square meters of floor area.

To study the problem of calibration effort, we reduced the amount of calibration data as if we had spent less time at each location and as if we had skipped certain locations. The fact that our algorithm interpolates on signal strength to give location makes it possible to skip whole rooms during calibration yet still test in those rooms. This is more difficult with most other 802.11 location algorithms which instead must classify signal strengths into only previously seen locations. As expected, the accuracy goes down with reduced calibration data, but it goes down surprisingly little. The results quantify the tradeoff between accuracy and effort, and suggest a prescription for manually calibrating systems of this type.

2 Location Measurement with 802.11 Signal Strength

One of the most attractive features of an 802.11 location system is that is does not require any extra infrastructure beyond the wireless network that already exists in many buildings. This is in contrast to other person-tracking systems like active badges and cameras which require the installation and maintenance of extra equipment. For a survey of location systems, see [4].

In the realm of 802.11 location, the first published work was Bahl and Padmanabhan's RADAR system[3]. RADAR worked based on a table of indoor locations and corresponding signal strengths. Using a manully calibrated table (as we do), their nearest neighbor algorithm gave a median spatial error of 2.94 meters. Another table based on simulated radio wave propagation allowed them to avoid most of the calibration work at the cost of increasing the median error to 4.3 meters. The RADAR paper also looked at the problem of reducing calibration effort. They found that reducing the number of calibration points from 70 to 40 had only a small negative impact on accuracy. This is similar to one of our results. In following work [5], RADAR was enhanced to use a Viterbi-like algorithm on short paths through the building. This reduced the median error to 2.37 meters.

As part of Carnegie Mellon's Andrew system, Small *et al.*[6] did a limited study of 802.11 location using eight discrete locations in a hallway. They built a table of signal strength *vs.* location and found that, upon returning to the eight locations, their system inferred the right location 87.5% of the time.

The Nibble[7] location service used signal-to-noise ratios instead of the more commonly used raw signal strengths. (The RADAR[3] researchers, however, found that signal strength was more indicative of location than signal-to-noise ratio.) The location algorithm was a Bayesian network, manually trained at discrete locations in two buildings. The Bayes formulation allowed the inclusion of *a priori* probabilities of a person's location as well as transition probabilities between locations. In one test on 12 locations in a hallway, Nibble assigned the highest probability to the correct location 97% of the time, not counting the 15% of the time it was inconclusive.

UCSD's ActiveCampus[8] project uses 802.11 to compute the location of wireless PocketPCs both indoors and outdoors. Instead of manual calibration, they use a formula that approximates the distance to an AP as a function of signal strength. Using a hillclimbing algorithm, their system computes location down to about 10 meters (35 feet) using signal strengths from multiple APs.

Ladd *et al.*[9] reported an 802.11 location system using Bayesian reasoning and a hidden Markov model. They took into account not only signal strengths, but also the probability of seeing an access point from a given location. Like other work, it was based on a manual calibration. Their system explicitly modeled orientation and achieved a median spatial error of about one meter using calibration samples taken approximately every 1.5 meters in hallways. In terms of accuracy, this is the best result we know of. However, they acknowledge the problem of calibration effort and suggest that calibrated locations could be automatically inferred by outfitting the calibrator with an accelerometer and magnetic compass.

Some of these systems are explicitly working toward more accuracy, but at the expense of increased calibration effort. The goals of this paper are to show how well a system can perform with a reasonable amount of calibration effort and to show how performance degrades if the effort is reduced.

3 Signal Strength Calibration

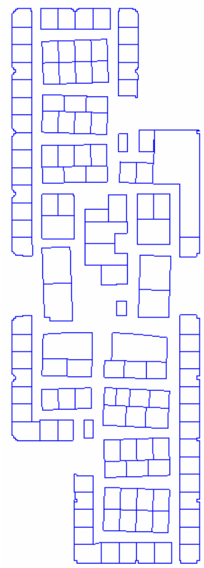

Figure 1: We did our experiments on this floor of a normal office building.

Our location algorithm works based on interpolation of signal strength calibration data taken from known locations. We gathered training data from rooms of one floor of a normal office building, shown in Figure 1. The area of the floor is about 2680 square meters, and it contains 132 rooms, 118 of which were accessible to us. Our building maps were extracted from our organization's database of Microsoft Visio® floor plans both as polygon representations and bitmaps. The coordinates of all the maps were expressed in actual floor coordinates in meters, giving us a consistent, intuitive representation of location for all our work.

For calibration, we walked into each accessible room with a wirelessly connected laptop PC running the logging program shown in Figure 2. The logging program uses a new prerelease version of the Wireless Research Application Programming Interface (WRAPI) [1] to get signal strengths from all the "visible" 802.11 access points (APs). In office-sized rooms, we stood as close as possible to the room's center, taking calibration data at this single point. In larger rooms like conference rooms, we took data from a few more locations for a total of 137 calibration locations. These locations are shown in the lower right segment of Figure 6. Given the area of the floor and the number of calibration points, there was on average one calibration location for every 19.5 square meters. We indicated the approximate (x,y) position of the calibration point by clicking on the map in Figure 2.

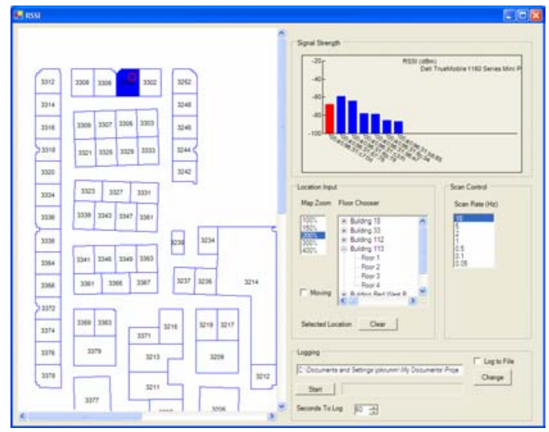

Figure 2: This is a screen shot of our signal strength logging program. The bar chart shows the signal strengths of the visible access points. The user indicates his or her position on the map by clicking. Signal strengths and locations are saved to a file for later training of the location program.

We spent 60 seconds at every calibration location, spinning around to factor out orientation effects. Our program scanned the available access points at about 3.4 Hz, giving about 200 scans from each location. Each scan gave the set of signal strengths and Media Access Control (MAC) addresses of the 802.11 access points that the laptop could see. On average, the wireless card could see 3.6 APs at any given time.

In all we took 27,796 sets of signal strength readings for calibration. On a different excursion through the same space a few days later, we took another 25,457 sets of readings to serve as test data. The two excursions were separated by a few days to avoid unnatural similarities between test and training data. Our laptop saw 22 different access points during calibration. As

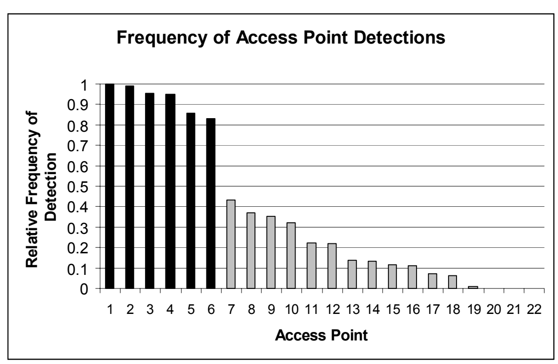

Figure 3: This plot shows how often each access point was seen as a fraction of how many times the most visible access point was seen. The gray bars are for access points on floors of the building other than where we did our experiments.

---

[1] WRAPI is a software library that provides an interface to 802.11 hardware in PCs running Windows XP. It works with any conforming wireless hardware and is available from the University of California, San Diego at http://ramp.ucsd.edu/pawn/wrapi/.

shown in Figure 3, some APs were not seen very often. About 31% of the signal strength readings came from APs on floors other than the one where we were doing our experiments.

In spite of our efforts to make calibration easy, the calibration procedure took about four tedious hours for one floor of a medium-sized office building, including the time it took to explain to curious office occupants what we were doing. This effort would grow quickly for an enterprise like a university or corporate campus with tens of buildings with multiple floors. Likewise, the effort would be large for warehouses or retail chains like grocery stores, both with a professed interest in this sort of location technology. Also periodic recalibration would be necessary to account for the addition or replacement of access points and structural changes that would affect radio propagation. This served as the inspiration for studying the effects of reducing the amount of necessary calibration data.

In order to facilitate the upcoming math, we will designate each set of calibration signal strength readings with a vector $\mathbf{s}_i$, where i indexes over all the calibration vectors in all the locations we visited. Each calibration vector has a corresponding (x_i, y_i) giving the location on the floor from which it was taken. Each signal strength vector $\mathbf{s}_i$ has 22 elements, one element for each AP ever seen in the experiment. The elements in $\mathbf{s}_i$ corresponding to unseen APs were given a value of one less than the minimum signal strength seen for the whole experiment. The signal strengths are returned from WRAPI as integers in units of dBm, where $\mathrm{dBm} = 10\log_{10}(\mathrm{milliwatts})$. We used these units throughout.

As mentioned above, the calibration excursion gave 27,796 signal strength vectors. In the next section we describe how we used these vectors to compute an interpolation function that gives location as a function of the signal strength vector. After that, we reduce the number of calibration vectors in a principled way to see how reducing the amount of calibration data affects the accuracy of location measurement.

4 Location as a Function of Signal Strength

Most other 802.11-based location work has formulated the task of location measurement as a classification problem, where the goal is to classify the signal strength vector into a discrete set of locations. This includes the probabilistic formulations where the classification result is given as a set of probabilities over all the possible locations.

The classification formulation is unsuitable for our goal of completely skipping certain rooms during the calibration phase. If a trained classifier has never seen a certain room, it will not ever classify data as coming from that room. Our algorithm can still interpolate a signal strength vector into a room it has never seen. If we want to then classify, we can check to see which room, if any, contains the computed location.

The interpolation formulation fits a function whose input is a signals strength vector $\mathbf{s}$ and whose output is a location (x, y). We chose to use radial basis functions, which means

$$
\begin{aligned}
x(\mathbf{s}) &= c_x + \sum_{j=0}^{M-1} \alpha_j K\left(\left\|\mathbf{s} - \mathbf{s}_j^*\right\|\right) \\
y(\mathbf{s}) &= c_y + \sum_{j=0}^{M-1} \beta_j K\left(\left\|\mathbf{s} - \mathbf{s}_j^*\right\|\right)
\end{aligned}
\tag{1}
$$

where $K(r)$ is the chosen kernel function, $\mathbf{s}_j^*$ are the M kernel function centers (described below), and α_j and β_j are the computed weights based on calibration data. $\left\|\mathbf{s} - \mathbf{s}_j^*\right\|$ is the Euclidian distance between the test signal strength vector $\mathbf{s}$ and the kernel center $\mathbf{s}_j^*$ in signal strength space. The offset (c_x, c_y) is simply the centroid of the training data, *i.e.*

$$
(c_x, c_y) = \frac{1}{N}\left(\sum_{i=0}^{N-1} x_i, \sum_{i=0}^{N-1} y_i\right) \tag{2}
$$

where N is the number of calibration vectors (27,796). The basics of radial basis functions are explained in [10].

We chose a simple, isotropic Gaussian kernel function:

$$
K(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \tag{3}
$$

This choice of kernel function also requires a choice of σ which we describe below. We also describe below our choice of the M kernel centers $\mathbf{s}_j^*$ using a k-means algorithm.

Our calibration data provides the basis for a least squares fit to compute the kernel weights. To compute the α_j (for the x coordinate), we minimize the squared error between the calibration data and $x(\mathbf{s}_i)$, which is

$$err = \sum_{i=0}^{N-1} \left(x_i - c_x - \sum_{j=0}^{M-1} \alpha_j K_{ij} \right)^2 \quad (4)$$

where $K_{ij} = K\left(\left\| \mathbf{s}_i - \mathbf{s}_j^* \right\|\right)$. Minimizing with respect to α_j gives a linear equation that can be solved for the vector $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{M-2}, \alpha_{M-1})^T$:

$$K^T K \boldsymbol{\alpha} = K^T \mathbf{x} \quad (5)$$

Here K is an $N \times M$ matrix of K_{ij} and $\mathbf{x} = (x_0 - c_x, x_1 - c_x, \ldots, x_{N-2} - c_x, x_{N-1} - c_x)^T$. Analogously, we get the β_j from $K^T K \boldsymbol{\beta} = K^T \mathbf{y}$.

We note that $K^T K$ has size $M \times M$, where M is our chosen number of kernel centers. One possible choice is to let each calibration point $\mathbf{s}_i$ serve as a kernel center, giving $M = N$. For us however, solving Equation (5) with $M = 27{,}796$ would exceed our patience and our PC's memory. Instead, we chose to cluster the signal strength calibration vectors from each location and use the cluster centers as kernel centers.

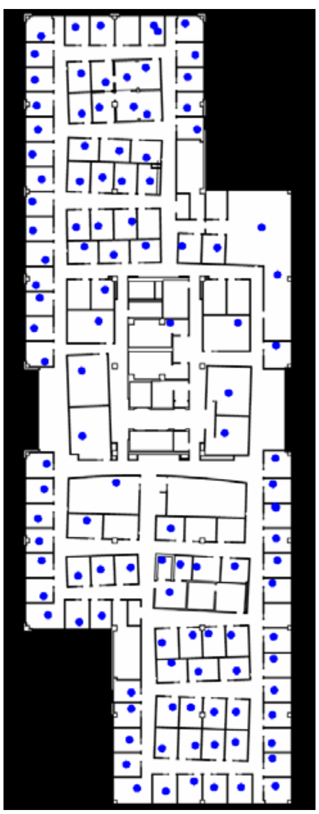

Figure 4: These are the locations we used to test our location algorithm. Altogether we tested on 25,457 signal strength vectors taken at 115 different locations.

Using a standard k-means algorithm, we computed $k = 5$ signal strength clusters from each calibration location, giving us $M = 137k = 685$ kernel centers to represent all 137 calibration locations on the test floor.

The only remaining choice was σ , which controls the size of the kernel functions in signal strength space. The same σ is used for all M kernel functions. For this we performed a simple linear search over possible values of σ . For each candidate σ , we first computed the kernel weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ using 70% of our calibration data. We then evaluated the candidates using the remaining 30% and picked the σ that gave the least rms distance error in (x, y) . In spite of the 70/30 split for computing σ , we still used 100% of the calibration data to cluster for the kernel centers.

As mentioned previously, we had another 25,457 test vectors taken a few days after the training data. To simulate a realistic scenario, we took these points by setting the laptop down on a flat surface for 60 seconds, not necessarily at the center of each room, logging signal strengths at the same 3.4 Hz rate. As with the calibration set, we approximated the ground truth position of the laptop by clicking on the map. Testing on this data, the radial basis function method gave an rms error of 3.75 meters.

The rms error was computed from the $L = 25{,}457$ test points, whose signal strengths are $\mathbf{s}_l$ and whose coordinates are (x_l, y_l) . For the rms error, we computed

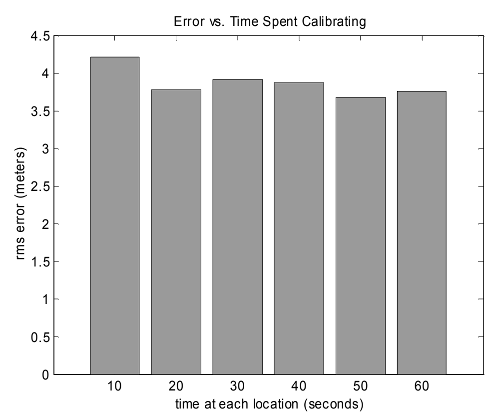

Figure 5: Reducing the amount of time spent gathering training data at each location did not significantly affect accuracy.

$$rms_s = \sqrt{\frac{1}{L} \sum_{l=0}^{L-1} \left\| \hat{\mathbf{x}}_l - \mathbf{x}_l \right\|^2} \quad (6)$$

where

$$\hat{\mathbf{x}}_l = (\hat{x}_l, \hat{y}_l)^T$$

$$\hat{x}_l = c_x + \sum_{j=0}^{M-1} \alpha_j K\left(\left\|\mathbf{s}_l - \mathbf{s}_j^*\right\|\right)$$

$$\hat{y}_l = c_y + \sum_{j=0}^{M-1} \beta_j K\left(\left\|\mathbf{s}_l - \mathbf{s}_j^*\right\|\right)$$

As a way of reducing noise and increasing accuracy, we applied a simple running average filter to the computed location vectors. The filter was 10 samples long, which induced a delay of about 2.9 seconds at our scanning rate of 3.4 Hz.

At first glance, an rms error of 3.75 meters seems significantly worse than RADAR's[3] median error of 2.94 meters or Ladd *et al.*'s[9] median error of about 1 meter. But both these systems required much more calibration effort. The first RADAR experiment covered the hallway outside about 54 rooms with 70 calibration points, and Ladd *et al.* covered the hallway with calibration points about 1.5 meters (5 feet) apart. In contrast, we used one calibration point per office-sized room on rooms with an average spacing of 2.85 meters. While RADAR and Ladd *et al.* show what is achievable with a careful calibration, we show what is achievable with a practical one. In the next section we show how the calibration effort can be significantly reduced without a significant reduction in accuracy.

5 Reducing Calibration Effort

One barrier to deploying an 802.11-based location system is the calibration effort. We spent about four hours calibrating at 137 locations on one floor of our building. We would like to know if this amount of calibration is really necessary. In particular, we are interested in evaluating the effect of reducing the time spent at each location and reducing the number of locations visited. By training on subsets of our original training data, we simulated the effects of reducing the time and number of locations. This section shows how location accuracy is affected with reduced calibration effort. It also gives a method to decide which locations to skip.

5.1 Reducing Time at Each Location

For our original calibration excursion, we spent at least 60 seconds at each location, gathering signal strength vectors at a rate of about 3.4 Hz. To test the effect of reduced time, we took the first s seconds of calibration data, processed it with the same training algorithm as described in Section 4, and then tested with the entire test set. Surprisingly, accuracy does not suffer significantly even when the time spent in each location is only ten seconds, as shown in Figure 5. At 10 seconds, the rms error had only grown by about 12% (0.45 meters) from the rms error at 60 seconds. At a data rate of 3.4 Hz, 10 seconds of data is only 34 signal strength vectors. This somewhat surprising result shows that it is not necessary to spend much time at each location during calibration.

5.2 Reducing Number of Locations

Our radial basis function method lends itself to interpolating over missing data better than a pure classification method would. Thus we can test the effect of skipping certain locations in the calibration phase, but still test anywhere in the test phase. For this part of our evaluation, we omitted some locations from our calibration data, fit the interpolation functions as normal on this reduced set, and tested on the original test set.

Figure 7 shows how we progressively reduced the number of calibration locations from the original full set of 137 locations down to 10% of the original. To choose k locations from the original calibration set, we ran a k-means clustering algorithm on the original locations to make k clusters. We picked the k original locations nearest the k cluster centroids as those for calibration.

Figure 7 shows the effect on rms error of reducing the number of locations visited for calibration. As expected, the error grows as the number of locations decreases. But even at 50%, the rms error has only grown by 20% (0.74 meters), and at 20% has grown by 42% (1.59 meters). At 10% of the original locations, the rms error is 9.19 meters, an increase of 145% (5.44 meters) over the best result at 100%. This shows that there is a significantly diminishing return for moving to a denser set of calibration points.

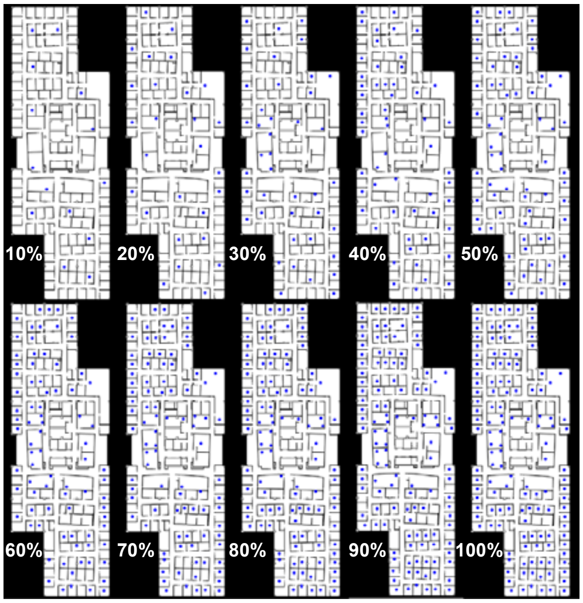

Figure 6: We experimented with the effect of reducing the number of calibration locations, going from 10% of the original 137 to the full set. Subsets were chosen with a k-means algorithm on the (x,y) locations.

This experiment also suggests a way to choose calibration points in a space. Starting with a dense set, say at the centroid of every room, use k-means to cluster the set into a representative subsample.

5.3 Reducing Time and Locations Together

Our experiments show that we can significantly reduce either the time spent at each location or the number of locations with only a minor degradation in accuracy. Figure 8 shows the effect of reducing both the time and the number of locations simultaneously. This plot shows that both the parameters can be reduced significantly without a correspondingly significant reduction in accuracy. As an example, spending 30 seconds in 40% of the locations increases the rms error by only about 21% (from 3.75 meters to 4.55 meters), yet reduces the calibration effort by much more than half.

6 Conclusion

Calibration for 802.11-based location can be very tedious. We calibrated one floor of an office building with the effort close to what we could expect for a large-scale deployment of an 802.11 location system. Using radial basis functions to interpolate location as a function of signal strength, we achieved an rms error of 3.75 meters. By formulating the problem as one of interpolation, we showed how it is possible to make calibration easier by skipping a significant fraction of the calibration locations. We also showed that it is unnecessary to spend much time at each location, as more time beyond a short minimum does not improve accuracy very much. These results show what level of accuracy we can achieve with a practical amount of calibration effort and how to maintain this level of accuracy with significantly reduced effort, making the deployment of such a system much more practical.

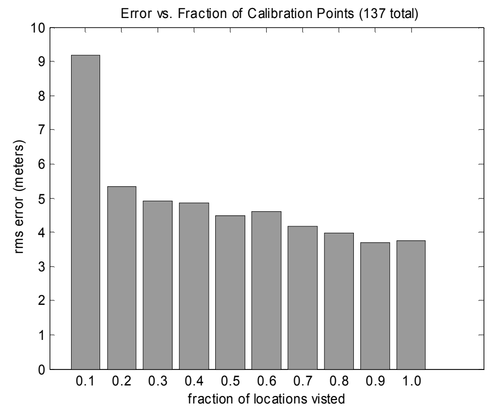

Figure 7: Reducing the number of calibration locations for training gradually increased location error. The effect was gradual until only 10% of the total points were used out of a total of 137.

References

1. Seidel, S.Y. and T.S. Rapport, *914 MHz Path Loss Prediction Model for Indoor Wireless Communications in Multifloored Buildings.* IEEE Transactions on Antennas and Propagation, 1992. **40**(2): p. 207-217.
2. Wolfle, G. and F.M. Landstorfer. *Dominant*

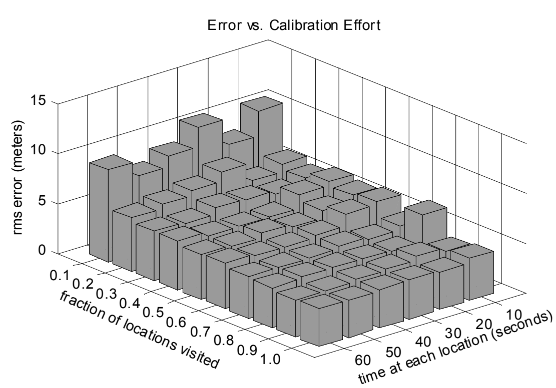

Figure 8: Reducing both the time spent in each location and the number of locations visited increases the error, but not until quite large reductions along both dimensions.

Paths for the Field Strength Prediction. in *48th IEEE Vehicular Technology Conference (VTC)*. 1998. Ottawa, Ontario, Canada.
3. Bahl, P. and V.N. Padmanabhan. *RADAR: An In-Building RF-Based Location and Tracking System*. in *IEEE INFOCOM 2000*. 2000. Tel-Aviv, Israel.
4. Hightower, J. and G. Borriello, *Location Systems for Ubiquitous Computing.* Computer, 2001. **34**(8): p. 57-66.
5. Bahl, P., V.N. Padmanabhan, and A. Balachandran, *Enhancements to the RADAR User Location and Tracking System*. 2000, Microsoft Research: Redmond, WA.
6. Small, J., A. Smailagic, and D.P. Siewiorek, *Determining User Location For Context Aware Computing Through the Use of a Wireless LAN Infrastructure*. 2000: Carnegie Mellon University.
7. Castro, P., et al. *A Probabilistic Room Location Service for Wireless Networked Environments*. in *Ubicomp 2001*. 2001. Atlanta, GA, USA: Springer.
8. Griswold, W.G., et al., *ActiveCampus - Sustaining Educational Communities through Mobile Technology*. 2002, University of California, San Diego: La Jolla. p. 19.
9. Ladd, A.M., et al. *Robotics-Based Location Sensing using Wireless Ethernet*. in *Eighth International Conference on Mobile Computing and Networking*. 2002. Atlanta, GA, USA.
10. Bishop, C.M., *Neural Networks for Pattern Recognition*. 1995, Oxford: Clarendon Press. 482.