

The Limits of Localization Using Signal Strength: A Comparative Study

Eiman Elnahrawy, Xiaoyan Li, Richard P. Martin
{eiman,xili,rmartin}@cs.rutgers.edu

Department of Computer Science, Rutgers University
110 Frelinghuysen Rd, Piscataway, NJ 08854

Abstract— We characterize the fundamental limits of localization using signal strength in indoor environments. Signal strength approaches are attractive because they are widely applicable to wireless sensor networks and do not require additional localization hardware. We show that although a broad spectrum of algorithms can trade accuracy for precision, none has a significant advantage in localization performance. We found that using commodity 802.11 technology over a range of algorithms, approaches and environments, one can expect a median localization error of 10ft and 97th percentile of 30ft. We present strong evidence that these limitations are fundamental and that they are unlikely to be transcended without fundamentally more complex environmental models or additional localization infrastructure.

I. INTRODUCTION

Localizing sensors is necessary for many higher level sensor network functions such as tracking, monitoring and geometric-based routing. Recent years have seen intense research investigating using off-the-shelf radios as a localization infrastructure for sensor nodes. The motivation has been a dual use one: using the same radio hardware for both communication and localization would enable a tremendous savings over deployment of a specific localization infrastructure, such as ones using directional antennas, very high frequency clocks, ultrasound, and infrared. In addition, such a system could avoid the high densities required by sensor aggregation approaches.

In this paper we explore the fundamental limits of localization using signal strength in indoor environments. Such environments are challenging because the radio propagation is much more chaotic than outdoor settings, where signals travel with little obstruction (e.g., GPS).

Exploring the limits of signal strength approaches is important because it tells us the localization performance we can expect without additional hardware in the sensor nodes and base-stations. We will show that a broad spectrum of signal-strength based algorithms have similar localization performance. We also present strong evidence that these limitations are fundamental and that they are unlikely to be transcended without qualitatively more complex models of the environment or additional hardware above that required for communication.

Although we use 802.11 Wireless Local Area Network (WLAN) technology because of its commodity status, our results are applicable to any radio technology where there are considerable environmental effects on the signal

propagation.

In order to better explore the limits of localization performance, we developed 3 algorithms. These algorithms are *area-based* rather than *point-based*. That is, the returned localization answer is a possible area (or volume) that might contain the sensor radio rather than a point.

We focus on area-based algorithms because they have a critical advantage in their ability to describe localization uncertainty. The key property area-based approaches have is that they can trade accuracy for precision, where *accuracy* is the likelihood the object is within the area and *precision* is the size of the returned area. Point-based approaches have difficulty describing such probabilistic trade-offs in a systematic manner. Using accuracy and precision, we can quantitatively describe the limits of different localization approaches by observing the impact of increased precision (i.e. less area) on accuracy.

To generalize our results, we compared our area-based approaches with several variants of well known existing algorithms. We ran our comparisons using measured data from two distinct buildings. We found that although area-based approaches are better at describing uncertainty, their absolute performance is similar to existing point-based approaches. In addition, our data combined with others shows that no existing WLAN based indoor localization approach has a substantial advantage in localization performance. A general rule of thumb we found is that using 802.11 technology, with much sampling and a good algorithm one can expect a median error of roughly 10ft and a 97th percentile of roughly 30ft.

A corollary of this result is that computationally simple algorithms that do not require many training samples are preferable because the performance of more complex algorithms is unlikely to be justified.

However, a promising result of our study is that we found with relatively sparse sampling, every 20 ft, or 400 ft²/sample, one can still get median errors of 15ft and 95th percentiles at 40ft. This is a promising result, because hand-sampling or deploying automatic sniffers is much more tractable at such densities.

Although examining the accuracy vs. precision tradeoff gives insight into performance limits, such an approach does not help us reason if the observed limitations are fundamental to the algorithm or inherent in the data. Using a Bayesian network, we express the uncertainties arising

Algorithm	Abbreviation	Description
Area-Based		
Simple Point Matching	SPM	Matches the RSS to a tile set using thresholds.
Area Based Probability	ABP- α	Matches the RSS to a tile set probabilistically with confidence bound $\alpha\%$.
Bayesian Network	BN	Returns the most likely tiles using a Bayesian network.
Point-Based		
Bayesian Point	B1	Returns the most likely point using a Bayesian network.
Averaged Bayesian	B2	Returns the mid-point of the top 2 most likely points.
RADAR	R1	Finds the closest training point based on distance in signal space.
Averaged RADAR	R2	Returns the midpoint of the closest 2 training points in signal space.
Gridded RADAR	GR	Applies RADAR using an interpolated grid.
Highest Probability	P1	Applies likelihood estimation to the received signal.
Averaged Highest Probability	P2	Returns the midpoint of the top 2 likelihoods.
Gridded Highest Probability	GP	Applies likelihoods to an interpolated grid.

TABLE I

All algorithms and variants.

from these effects in terms of probability density functions (PDFs) that describe the likely position as a function of the observed data and a widely used propagation model. Our results show that there is significant uncertainty arising from the data given the model.

The rest of this paper is organized as follows. In Section II we provide a description of related work. Section III sketches both the area-based and point-based approaches used in the work. Next, Section IV introduces area-based performance metrics. In Section V we present the performance of the algorithms. Finally, in Section VI we conclude.

II. RELATED WORK

Recent years have seen tremendous efforts at building small and medium scale localization systems for sensor networks. The underlying principles vary from trilateration, triangulation, scene matching (e.g., fingerprinting), and combinations of these approaches. When aggregates of sensors are available, a wider set of mathematical foundations are possible, including multidimensional scaling [1], optimization[2], and ad-hoc approaches [3], [4]. The technologies used have also exhibited a wide range: ultrasound [5], [6], [7], infrared [8], 802.11, and custom radios [9].

Within this wide variance, the works using 802.11 and signal strength are closely related to ours [10], [11], [12], [13], [14], [15], [16], [17]. The two fundamental building blocks of all these are: (1) a classifier to relate an observed set of signal strengths to ones at known locations, and (2) a function of distance to signal strength. A full treatment of the myriad of techniques for estimating signal strength at locations, classifiers, and distance functions is beyond the scope of this work. We instead show they perform similarly. While our results show that absolute performance depends on the environment, our absolute results are still consistent with the above works.

The work most closely related to ours is [11]. Their data shows that a host of matching and classification algorithms, ranging from maximum-likelihood analysis to neural networks, has similar performance. They did not speculate on the resulting similarity, however.

III. LOCALIZATION ALGORITHMS

In this section we give a broad overview of our algorithm menagerie, summarized in Table I. Because our purpose is to explore similarities in algorithmic performance our descriptions focus on each algorithm’s broad strategy. The reader is encouraged to pursue the references for details.

Before describing the algorithms, we first define terms and then describe how we interpolate a topological grid used by many of the algorithms. We then describe 3 area-based algorithms: Simple Point Matching (SPM), Area-Based Probability (ABP), and Bayesian (BN). We then describe the point-based algorithms: Bayesian Point (B1), Averaged Bayesian (B2), RADAR, (R1), Averaged RADAR (R2), Gridded RADAR (GR), Highest Probability (P1), Averaged Highest Probability (P2) and Gridded Highest Probability (GP).

We use the following definitions and terms. The n access points are AP_1, \dots, AP_n . The training data, T_o , is used as offline input. It consists of a set of empirically measured signal *fingerprints*, \bar{S} along with the m locations, (x, y) where they were collected. I.e., $T_o = \{[(x_i, y_i), \bar{S}_i]\}, i = 1 \dots m$, thus there are m fingerprints. The fingerprint at a location, i , consists of the set of expected average received signal strength, \bar{s}_{ij} , for each AP_j . A default value for \bar{s}_{ij} is assigned in a fingerprint if no signal is received from AP_j at a location.

The object to be localized collects a set of *received signal strengths* (RSS) when it is at a location. An RSS is similar to a fingerprint in that it contains a set of APs, and a mean for each AP_j . An RSS also maintains a standard deviation of the sample set at each location i and AP_j , σ_{ij} .

A. Interpolating Fingerprints

Many of the algorithms require a method of building a regular grid of tiles that describe the expected fingerprint for the area described by each tile. The tiles are a simple way to map the expected signal strength to locations, as opposed to field vectors or more complex shapes. Because direct measurement of the fingerprint for each tile is expensive, we use an interpolation approach to build an Interpolated Map Grid (IMG). Building an IMG is similar

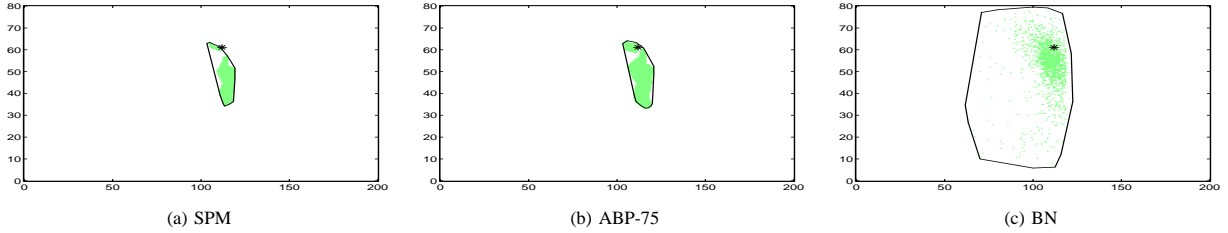


Fig. 1. Area returned by the different algorithms for the CoRE building. The area is circumscribed by the smallest convex polygon. The true location is shown as a point.

to “surface fitting”; the goal is to derive an expected fingerprint for each tile from the training set that would be similar to an observed one.

We build an IMG for a floor using the training fingerprints for each AP_j independently on a grid of 30in square tiles. Although there are several approaches in the literature for interpolating surfaces, e.g, splines, we used triangle-based linear interpolation. We divide the floor into triangular regions using a Delaunay triangulation where the location of the T_o samples serve as anchor points. In a few cases, we had to add anchor points at the corners of the floor. We then linearly interpolate the expected signal strength using the “height” of the triangle at the center of the tile.

We found our approach desirable because: (1) it is simple and fast, (2) the derivative of the RSS as a function of location does not vary widely, so simple interpolation performed adequately, (3) it is insensitive to the size of the underlying tiles (we tried tiny 10×5in tiles, observing almost no effect) and (4) the sample spacing need only to follow a uniform distribution, rather than have “precise” spacing.

B. Area-Based Algorithms

Figure 1 shows a sample of the area-based algorithms’ results for the CoRE building. The actual point is shown by a “*” and the convex hulls of the returned areas are outlined. The SPM and ABP algorithms perform similarly, but the BN algorithm has a much different profile.

1) *Simple Point Matching*: The strategy behind SPM is to find a set of tiles that fall within a threshold of the RSS for each AP independently, then return the tiles that form the intersection of each AP’s set.

More formally, SPM first finds n sets of tiles, one for each $AP_j, j = 1 \dots n$, that “match” all fingerprints $\bar{S}_l = (\bar{s}_{l1}, \dots, \bar{s}_{ln})$, for the object to be localized. The matching tiles for each AP_j are found by adding an expected “noise” level, q to \bar{s}_{lj} , and then returning all the floor tiles that fall within the expected threshold, $\bar{s}_{lj} \pm q$ (We substituted a value of -92 dBm for missing signals). SPM then returns the area formed by intersecting all matched tiles from the different AP tile sets.

For the algorithm to be eager, i.e., to find the fewest high probability tiles, it starts from a very low q . However, it then runs the risk of returning no tiles when the intersection among the APs is empty. Thus, on an empty

intersection, the algorithm additively increases q i.e., it first tries $q, 2q \dots$ until a non-zero set of tiles results. In the worst case, a non-empty intersection will result, even if q expands to the dynamic range of signal readings.

An important issue is how to pick q for each AP_j . We experimented with the value of q to observe how sensitive the algorithm is to this parameter and found that the algorithm is quite insensitive to q when it is close to the maximum $\{\sigma_{ij}\}$. We thus used the maximum $\{\sigma_{ij}\}$ over all fingerprints for q . Although choosing q in this manner makes our algorithm more ad-hoc, it also makes SPM simpler, scalable, and faster.

Although SPM is ad-hoc, it is quite similar to a more formal approach using Maximum Likelihood Estimation (MLE) [18]. The SPM noise level corresponds quite closely to the confidence $z_{\frac{\alpha}{2}} \times \sigma_{lj}$ of that algorithm, σ_{lj} is the standard deviation at the object’s location. SPM is in effect an approximation of the MLE method, where SPM eagerly searches for the lowest confidence that yields a non-empty area.

2) *Area Based Probability*: The strategy used by ABP- α is to return a set of tiles bounded by a probability that the object is within the returned set. We call the probability, α , the *confidence*, and it is an adjustable parameter. ABP’s approach to finding a tile set is to compute the likelihood of an RSS matching a fingerprint for each tile, and then normalize these likelihoods given the priors: (1) object must be on floor, and (2) all tiles are equally likely. ABP then returns the top probability tiles whose sum matches the desired confidence. ABP thus stands on a more formal mathematical foundation than SPM.

In order to find the likelihood of the RSS matching each tile in isolation, ABP assumes the distribution of both the received signal strengths for each AP in a fingerprint and RSS follows a Gaussian distribution with mean \bar{s}_{ij} . Although this assumption is often not true, it significantly simplifies the computations with little performance loss. We model the standard deviation for each AP_j separately, using maximum deviation observed in any fingerprint for the AP.

Using Bayes’ rule, ABP computes the probability of being at each tile L_i on the floor given the fingerprint of the localized object, $\bar{S}_l = (\bar{s}_{lj})$:

$$P(L_i | \bar{S}_l) = \frac{P(\bar{S}_l | L_i) \times P(L_i)}{P(\bar{S}_l)} \quad (1)$$

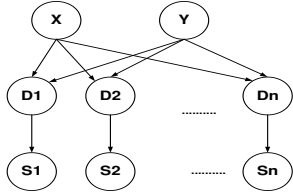


Fig. 2. The Bayesian network used in our experiments.

However, $P(\bar{S}_l)$ is a constant c . With no prior information about the exact object’s location, ABP assumes that the object to be localized is equally likely to be at any location on the floor, i.e., $P(L_i) = P(L_j), \forall i, j$. Thus, Equation 1 can be rewritten as:

$$P(L_i|\bar{S}_l) = c \times P(\bar{S}_l|L_i) \quad (2)$$

Without having to know the value c , ABP can just return the tile L_{max} , where $L_{max} = \arg \max(P(\bar{S}_l|L_i))$, by computing $P(\bar{S}_l|L_i)$ for every tile i on the floor. Up to this step ABP is very similar to the traditional Bayesian approaches [14], [17], with the exception of the Gaussian and variance assumptions.

ABP extends the referenced approaches by its final step where it computes the actual probability density of the object for each tile on the floor, given that the object must be at exactly one tile, i.e., $\sum_{i=1}^L P(L_i|\bar{S}_l) = 1$. Given the resulting density, ABP returns the top probability tiles up to its confidence, α . We found that useful values of α can have a wide dynamic range, between 0.5 and less than 1. While a confidence of 1 returns all the tiles on the floor, picking a useful α is not difficult because in practice, some tiles have a much higher probability than the others, while at the same time the difference between these high-probability tiles is small. Therefore only a sufficiently high α is needed to return these tiles, while at the same time making the size of a tile set insensitive to small changes in α .

3) *Bayesian Networks (BN)*: Bayes nets are graphical models that encode dependencies and relationships among a set of random variables. The vertices of the graph correspond to the variables and the edges represent dependencies [19]. The Bayes net we use encodes the relationship between the RSS and the location based on a signal-versus-distance propagation model. The initial parameters of the model are unknown, and the training set is used to adjust the specific parameters of the model according to the relationships encoded in the network.

Figure 2 shows the simple network we used. Each random variable $s_j, j = 1 \dots n$ denotes the expected signal strength from the corresponding access point AP_j . The values of these random variables depend on the Euclidean distance D_j between the AP’s location, (x_j, y_j) , and the location where the signal s_j is measured (x, y) . The baseline expected value of s_j follows a signal propagation model $s_j = b_{0j} + b_{1j} \log D_j$, where b_{0j}, b_{1j} are the parameters specific to each AP_j . The distance $D_j = \sqrt{(x - x_j)^2 + (y - y_j)^2}$ in turn “depends” on the location (x, y) of the measured signal. The network models noise and outliers by modeling the expected

value, s_j , as a t -distribution around the above propagation model, with variance τ_j (which has a long tail). I.e., $s_j \sim t(b_{0j} + b_{1j} \log D_j, \tau_j, 2)$. We specifically used a t -distribution rather than a Gaussian in order to better model the outliers of real data.

Using the training fingerprints T_o and the fingerprint vector of the mobile object, the network then learns the specific values for all the unknown parameters b_{0j}, b_{1j}, τ_j and the joint distribution of the (x, y) location of the object.

In general, there is no closed form solution for the returned joint distribution of the (x, y) location. Therefore, we use a Markov Chain Monte Carlo (MCMC) simulation approach to draw samples from the joint density [20], using an off-the-shelf statistics package, BUGS (www.mrc-bsu.cam.ac.uk/bugs/). We then pick the samples that give a 95% confidence on the density. Finally, we approximate the returned area by the tiles where those samples fall.

Figure 1(c) shows a substantive drawback of the Bayes net approach is that it yields a large number of disconnected tiles; we call this the “scatter effect”. Although the tiles are concentrated around the most likely location, the disconnection is substantial and can interfere with higher-level functions, such as mapping the object into a room.

We developed two additional point-based versions of our Bayes net approach. Specifically, after obtaining the joint density of the (x, y) location of the localized object, we either return the center point of the highest probability tile (Bayesian Point, **B1**), or the midpoint of the top two tiles (Averaged Bayesian, **B2**).

C. Point-Based Algorithms

The first point-based algorithm we used is the well known RADAR [10], which we refer to as **R1**. Its approach is to return the location of the closest fingerprint to the RSS in the training set, using Euclidean distance in “signal space” as the measurement function (i.e., it views the fingerprints as points in an N-dimension space, where each AP forms a dimension).

A second version of the algorithm returns the average position (centroid) of the top k closest vectors; our averaged RADAR algorithm, **R2**, averages the closest 2 fingerprints. A disadvantage of RADAR is that it can require a large number of training points. Our gridded RADAR algorithm, **GR**, uses the IMG as a set of additional fingerprints over the basic R1.

Our second point-based approach, **P1** uses a typical probabilistic approach applying Bayes’ rule [15]. We also evaluate a modified version, **P2**, that returns the mid-point of the top 2 training fingerprints. Finally, much like GR, we evaluate a variant of P1, **GP**, that uses fingerprints based on an IMG.

IV. LOCALIZATION METRICS

In this section we describe the performance metrics we use to evaluate the algorithms. The traditional localization metric is the distance error between the returned position and the true position. There are many ways to describe

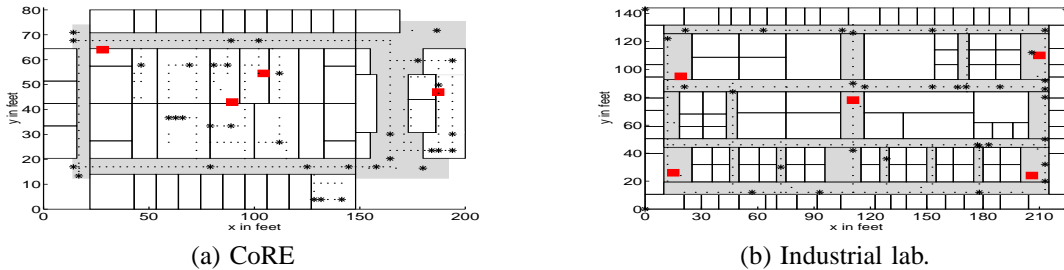


Fig. 3. Plan and setup of floors in the CoRE building and Industrial laboratory. Grey spaces are corridors, white spaces are offices or laboratories. Squares are the APs locations. Small dots show the testing locations, large dots an example random training set.

the distribution of the distance error: the average, the 95th percentile, or even the full CDF. The problem with this traditional metric is that it does not apply to area-based approaches. We thus introduce metrics appropriate for area-based algorithms.

Because many indoor sensor-network applications can operate at the level of rooms, we also extend the traditional metric, accuracy and precision to operate at the room-level. That is, we translate the returned points and areas into rooms and observe the performance of the algorithms in units of rooms rather than as raw distances or areas.

A. Area-Based Metrics

a) Tile Accuracy: Tile accuracy refers to the percentage of times the algorithm is able to return the true tile, i.e., the tile containing the object. This metric can be somewhat misleading because often, the true tile is close to the returned set, which motivates the next metric.

b) Distance Accuracy: This metric is the distance between the true tile and tiles in the returned area, as measured from the tiles' center.

In order to gauge the distribution of tiles in relation to the true location, we first sort all the tiles according to this metric. We can then return the distances of the 0th (minimum), 25th, 50th (median), 75th, and 100th (maximum) percentiles of the tiles. This metric is somewhat comparable to the traditional metric, although one should look at both the minimum and maximum distances.

c) Precision: The overall precision refers to the size of the returned area, i.e., the sq.ft. To normalize the metric across systems, it can be expressed as a percentage of the entire possible space (i.e., the size of the floor).

B. Room-level Metrics

We divide up a floor into a set of rooms, where each room is a rectangular area or a union of adjacent rectangular areas. The entire floor is covered by rooms. For point-based algorithms the mapping of the returned points into rooms is simple: the returned room is the room containing the point.

For area-based algorithms, the relationship on how to map areas into rooms is more complex. Our approach is to map the area into an ordered set of rooms, where the ordering tells the user which room order to try to find the localized object. We chose a simple approach that orders

the rooms by the absolute sizes of the intersection of the returned areas with room areas. While this favors larger rooms, how to normalize for room sizes remains unclear. On one hand, a large fraction of a room returned could imply the room is likely to contain the object, while on the other hand a very small room might be fully covered only due to noise.

d) Room Accuracy: The room accuracy corresponds to the percentage of times the true room, i.e., the room where the object is located, is returned in the ordered set of rooms. An important variation of this metric is the n -room accuracy, which is the percentage of times the true room is among the top n -rooms.

e) Room Precision: This metric corresponds to the average number of rooms on the floor returned by the algorithm. because this metric can be misleading in cases where there are a few large rooms on the floor, it can be expressed as a percentage of the total number of rooms.

V. EXPERIMENTAL STUDY

In this section we describe our experimental study and results. We characterize the performance of our area-based localization algorithms and then compare their performance with single-location based approaches. We close with a description of the uncertainty.

A. Experimental Setup

In order to show our results are not an artifact of a specific floor, we used measured RSS data from 2 sites. The first site is our Computer Science department CoRE building ("CoRE"), while the second site is an office building at an industrial laboratory ("industrial"). Figures 3(a) and (b) show the layout of these 2 floors, respectively. The figure shows the position of all the fingerprints as dots and the locations of the AP's as larger squares. Hallways are shaded in grey, offices and laboratories are white. We did not collect data in all rooms because many of the side rooms are private offices.

We collected fingerprints at 286 locations on the 3rd floor of the CoRE building over a period of 2 days. The floor contains just over 50 rooms in a 200x80ft (16000 ft²) area. To map the data into rooms, we divided the corridor spaces into 4 "rooms". A Dell laptop running Linux equipped with an Orinoco silver card was used to collect the samples. The sampling procedure was to run the `iwlist scan` command once a second for 60 seconds.

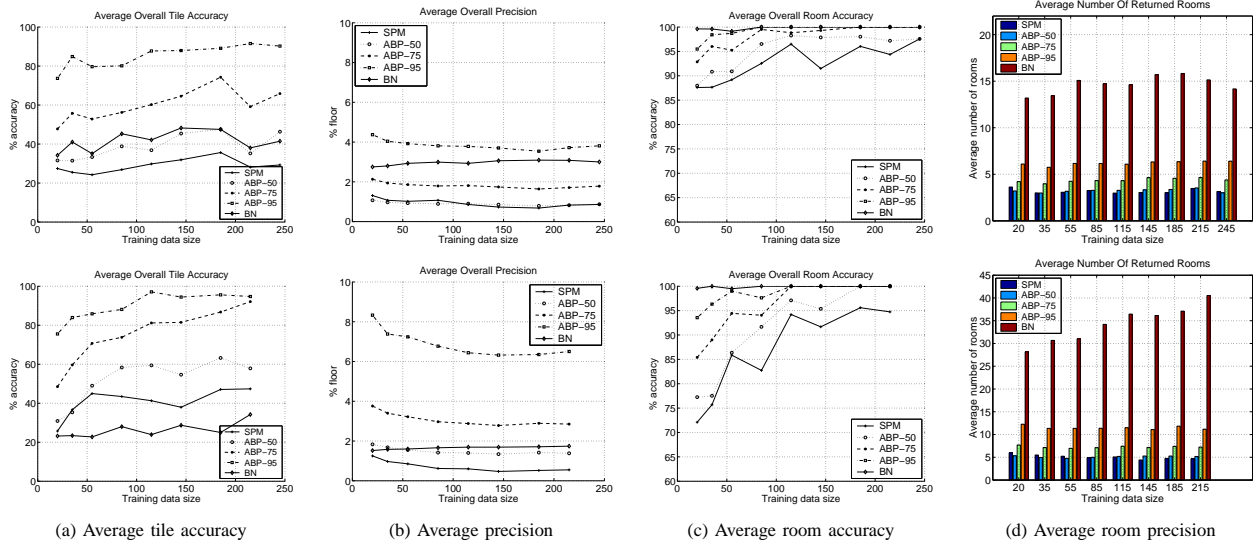


Fig. 4. Comparing accuracy and precision in area-based algorithms over different training data sizes. The first row is for the CoRE floor, and the second row is for the industrial floor.

A total of 253 fingerprint vectors were collected from the industrial site. All of them lie along the corridors. The fingerprints were collected over several days using a Linux IPAQ. The floor includes about 115 rooms in a 225x144ft (32400 ft²) area and has many corridors in-between these rooms. We divided the corridors into 20 rooms

In the CoRE data, we measured a value of -3 dBm for the SPM and ABP noise levels for 3 APs, and a value of -5 dBm for the fourth AP. For the industrial data we used a noise level of -4 dBm for all 5 APs.

To evaluate the algorithms, we divide the data into training and testing sets. To emulate an actual system, we assume the training set is collected in some manner, e.g., by hand sampling or previously deployed sniffers, and then each algorithm must locate the RSS vectors in the testing set. Our metrics then summarize these localization attempts.

B. Impact of the Training Set

Both the number and location of the fingerprints in the training set impact localization performance. To investigate the effect of location we experimented with different ways of picking training sets depending on the fingerprints' coordinates. We tried random uniform distributions over x and y , the closest point to a regular grid (using different grid sizes), and uniformly distributed around each AP. The stars in Figure 3 show an example training set of about 30 samples picked following a random uniform distribution. We found that as long as the samples are uniformly-distributed, but not necessarily uniformly-spaced, the specific methodology had no measurable effect on our results.

The number of samples has an impact, although it is not as strong as one might expect. Figure 4 shows the effect of the training set size on various metrics: (a) tile accuracy, (b) precision, (c) room accuracy, and (d) room precision. The figure presents these metrics as averaged values.

The algorithms have substantially different tile-level accuracies and precisions. The results also clearly show the fundamental tradeoff for tile accuracy and precision; any algorithm that improves tile accuracy worsens precision. Interestingly, for a given algorithm the precision is quite insensitive to the size of the training set.

Although the accuracy on the industrial data seems more sensitive to training set size, this is somewhat misleading because the industrial floor is twice the size of CoRE. Tile accuracy of the algorithms stabilizes at around 85 and 115 fingerprints for the CoRE and industrial floors, respectively. Room accuracy stabilizes at about 115 points for both sets. However, the accuracy is quite robust to lower samples sizes. Only for very low sample densities, as can be seen in the 35 fingerprint industrial set, does accuracy fall off sharply.

Examining the room-level performance, we see that all the algorithms are quite good at finding the correct room with a sufficient training set. The room-level precision of BN is quite poor due to the scatter effect.

C. Accuracy

Although average tile accuracy gives us general trends, we must peer into the data for a better picture. Figure 5 shows some sample distance accuracy CDFs for different training sizes. We found that in general, the SPM and ABP algorithms have comparable distance accuracy in the intermediate percentiles (25%, median and 75%); the differences are most pronounced at the edges of the distribution. Figures 5(a) and (c) show that for SPM and ABP, the 90th percentile for these intermediate accuracy percentiles occur around 15ft, 20ft, and 30ft, respectively. The Bayesian accuracy percentiles are a little larger due to the scatter effect.

A mathematical way to express this behavior is to examine what happens as the confidence level of ABP increases; the minimum distance between the returned

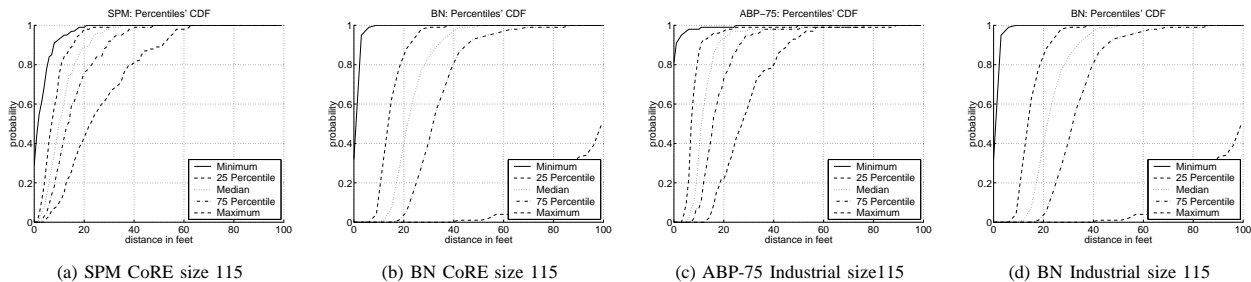


Fig. 5. Distance Accuracy CDFs for SPM and BN on CoRE (a,b) and ABP-75 and BN on the Industrial data (c,d).

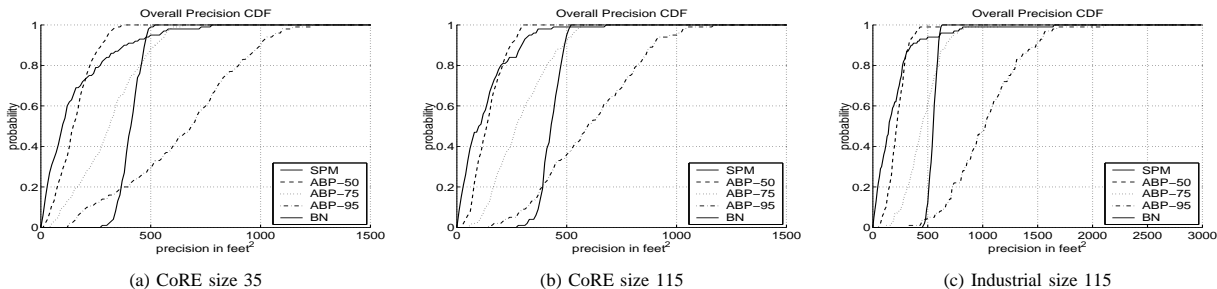


Fig. 6. Sample precision CDFs across area-based algorithms with different training sizes and floors.

area and the true location decreases while the maximum distance increases. As the confidence goes up, the added tiles are either closer to, or farther from, the true location with almost equal probability.

The percentiles of the distance accuracy tell us that tile-accuracy can be misleading; the returned areas by the different algorithms are actually much more similar in their spatial relationship to the true location than tile-level accuracy would suggest. As illustrated in Figure 5, what happens is that the primary difference in the algorithms is in how much uncertainty they return rather than a fundamental difference in accuracy.

D. Precision

Turning to precision, Figure 6 shows the CDF of the precision for the different algorithms. Indeed, the algorithms have a wide variety of precisions. As expected, SPM tends to return the lowest precision, performing somewhere between ABP-50 and ABP-75. BN has a surprisingly consistent, although somewhat low, precision, as evidenced by its steep CDF. An interesting phenomena is that in spite of the scatter effect, the precision of BN is comparable to the other approaches.

E. Comparing Algorithms

Having shown that a wide range of area-based algorithms have similar fundamental performance, we now expand our investigation to point-based algorithms.

Figure 7 shows the CDF of the traditional distance error metric for the point-based algorithms, along with the CDF of the median percentile for ABP-75 and BN. The other area-based approaches are not shown because they perform similarly to ABP-75 in terms of the median error.

The key result of Figure 7 is the striking similarity of the algorithms. The CDFs have a similar slope, medians around 10-15ft, and long tails after the 97th percentile. Indeed, many CDFs differ by less than a few feet, and there are regions where they cross. The exceptions are Bayesian approaches, BN, B1 and B2, which have uniformly higher errors than the rest; this effect is most clearly seen in Figure 7(b) for the industrial set.

The CDFs of the point-based algorithms also show only marginal improvements in localization performance as a function of sample size with a sufficient sample density. As a general rule of thumb for both data sets, a sample density of 1/230 ft² (every 15ft) was sufficient coverage for all the algorithms. However, as Figure 7(a) shows, reasonable performance is obtainable with much less sampling at 1/450 ft² (every 21 ft).

Turning to room-level accuracy, Figure 8 shows the room accuracy. The point-based algorithms can only return a single room. For the area-based algorithms, the stacked bar-graph shows the cumulative percentiles of the top-3 rooms, followed by all other returned rooms as a single stack.

Figure 8 shows similar accuracies across many of the algorithms, with the exception of the Bayesian approaches and at low sampling densities. The lower room precision of the BN algorithm is due to the “scatter” effect, illustrated in Figure 1(c). The low room performance for some of the algorithms in Figure 8(a) is accounted for by the low number of samples, which make building an accurate IMG difficult.

F. Fundamental Uncertainty

We now show strong evidence of why the algorithms deliver similar performance. Our approach begins with

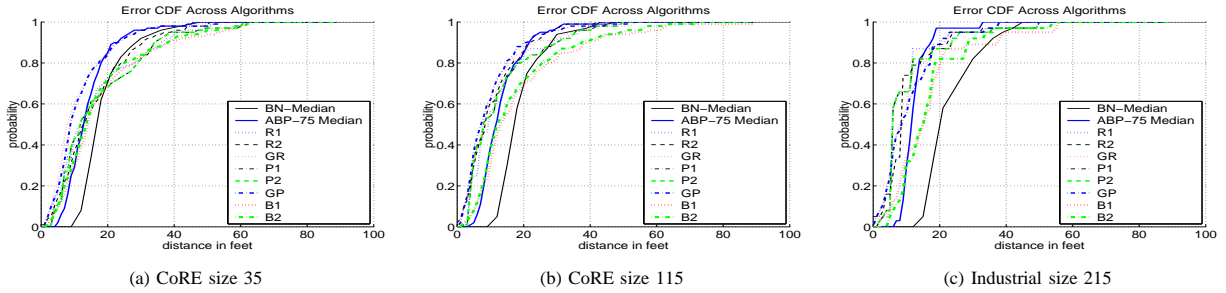


Fig. 7. Error CDF across all algorithms with different training set sizes.

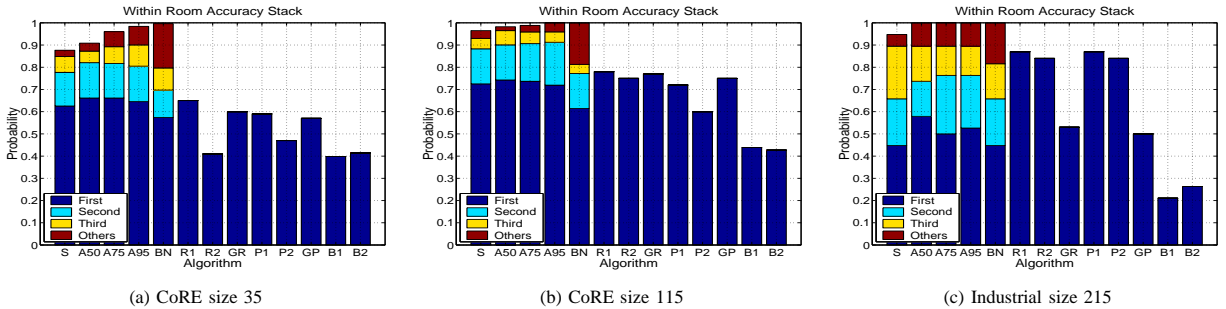


Fig. 8. Room accuracy. The stacked bars show the percentage of times the 1st, 2nd, 3rd, and remaining rooms are the correct ones.

the Bayesian network because this approach alone gives a view of the spatial uncertainty PDF given both measurements and a mathematical model of causal relationships.

Figure 9 shows 4 sample uncertainty PDFs along both the x, y axes generated from the Bayesian network. The wide distributions, especially in the industrial data set, show there is a high degree of uncertainty in the positions. The strong attenuation along the x -axis in the CoRE data, due to the higher number of walls in that dimension, greatly reduces the uncertainty along the x -axis.

Given that [11] found a host of learning approaches had similar performance to maximum likelihood estimation (similar to P1) with sufficient sampling, and our results also show similar performance between P1 and a broad spectrum of approaches, we can speculate using transitive reasoning that the algorithms in [11] would also have similar performance to those evaluated here.

The PDFs from the BN algorithm, along with the very similar performance shown by Figure 7 for the rest of the algorithms, give very strong evidence that the fundamental uncertainty of all of the algorithms is not much better than those shown in Figure 9. Intuitively, the point-based algorithms find the peaks in the PDFs and return that as the location. The area-based algorithms can explore more of the PDF, but cannot narrow it.

VI. CONCLUSIONS

In this work we characterized the limits of a wide variety of approaches to localization in indoor environments using signal strength and 802.11 technology. We found that a median error of 10ft and a 97th percentile of 30ft is an expected bound for the performance of a good algorithm and much sampling. However, our results also showed

that a median error of 15ft with a 40ft 97th percentile is obtainable with much less sampling effort.

Our comparisons and examinations of uncertainty PDFs suggest that algorithms based on matching and signal-to-distance functions are unable to capture the myriad of effects on signal propagation in an indoor environment. While many of the algorithms can explore the space of this uncertainty in useful ways, e.g., by returning likely areas and rooms, they cannot reduce it. Still, the localization accuracy is significant and useful, as we showed when mapping the objects into rooms.

Given our large training sets, it is unlikely that additional sampling will increase accuracy. Adding additional hardware and altering the model are the only alternatives. For example, ray-tracing models that account for walls and other obstacles have been employed [21]. Pursuing the modeling strategy, however, we are left with a trade-off in model complexity vs. accuracy, and such questions are not easily answered. For example, it is unclear if building models at the level of detail where one must model all items impacting signal propagation (walls, large bookshelves, etc.) would be worth the improvements in localization accuracy.

REFERENCES

- [1] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Fourth ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)*, Annapolis, MD, June 2003.
- [2] L. Doherty1, K. S. J. Pister, and L. E. Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, AK, Apr. 2001.
- [3] T. He, C. Huang, B. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes in large scale

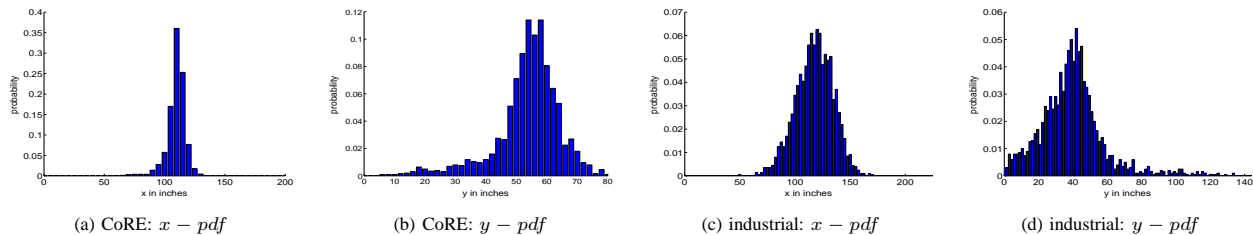


Fig. 9. The uncertainty along the x, y dimensions in the CoRE and the industrial setups.

- sensor networks,” in *Proceedings of the Ninth Annual ACM International Conference on Mobile Computing and Networking (MobiCom’03)*, San Diego, CA, Sept. 2003. [Online]. Available: citeseer.nj.nec.com/he03rangefree.html
- [4] D. Niculescu and B. Nath, “Ad hoc positioning system (APS),” in *GLOBECOM (1)*, 2001, pp. 2926–2931. [Online]. Available: citeseer.nj.nec.com/niculescu01ad.html
- [5] M. Hazas and A. Ward, “A high performance privacy-oriented location system,” in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Dallas, TX, Mar. 2003. [Online]. Available: <http://citeseer.nj.nec.com/hazas03highperformance.html>
- [6] N. Priyantha, A. Chakraborty, and H. Balakrishnan, “The Cricket Location-Support system,” in *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, Aug. 2000. [Online]. Available: <http://citeseer.nj.nec.com/li00scalable.html>
- [7] A. Savvides, C.-C. Han, and M. Srivastava, “Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors,” in *Proceedings of the Seventh Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, July 2001.
- [8] R. Want, A. Hopper, V. Falcao, and J. Gibbons, “The active badge location system,” *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, Jan. 1992.
- [9] K. Lorincz and M. Welsh, “Motetrack: A robust, decentralized location tracking system for disaster response,” in preparation.
- [10] P. Bahl and V. N. Padmanabhan, “RADAR: An In-Building RF-Based User Location and Tracking System,” in *INFOCOM*, March 2000. [Online]. Available: citeseer.nj.nec.com/bahl00radar.html
- [11] R. Battiti, M. Brunato, and A. Villani, “Statistical Learning Theory for Location Fingerprinting in Wireless LANs,” University of Trento, Informatica e Telecomunicazioni, Technical Report DIT-02-086, Oct. 2002.
- [12] Ekahau, Inc., “The Ekahau Positioning Engine 2.1,” July 2003, whitepaper available from <http://www.ekahau.com>.
- [13] P. Krishnan, A. S. Krishnakumar, W.-H. Ju, C. Mallows, and S. Ganu, “A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks,” in *INFOCOM*, Oct. 2004.
- [14] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavradi, and D. S. Wallach, “Robotics-based location sensing using wireless Ethernet,” in *Proceedings of The Eighth ACM International Conference on Mobile Computing and Networking (MOBICOM)*, Atlanta, GA, Sept. 2002.
- [15] T. Roos, P. Myllymaki, and H. Tirri, “A Statistical Modeling Approach to Location Estimation,” *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, Jan-March 2002.
- [16] A. Smailagic and D. Kogan, “Location sensing and privacy in a context aware computing environment,” *IEEE Wireless Communications*, vol. 9, no. 5, Oct. 2002. [Online]. Available: <http://citeseer.nj.nec.com/smailagic01location.html>
- [17] M. Youssef, A. Agrawal, and A. U. Shankar, “WLAN location determination via clustering and probability distributions,” in *Proceedings of IEEE PerCom’03*, Fort Worth, TX, Mar. 2003.
- [18] D. Hand, H. Mannila, and P. Smyth, *Principles Of Data Mining*. The MIT Press, 2001.
- [19] D. Heckerman, “A tutorial on learning with bayesian networks,” Microsoft Research, Tech. Rep. MSR-TR-95-06, March 1995.
- [20] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed. Chapman and Hall, 2004.
- [21] K. R. Schauback, N. Davis, and T. Rappaport, “Ray tracing method for predicting path loss and delay in microcellular environments,” in *IEEE Vehicular Technology Conference, VTC92*, Denver, CO, May 1992.