

Using Area-based Presentations and Metrics for Localization Systems in Wireless LANs

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

In this work we show the utility of WLAN localization using areas and volumes as the fundamental localization unit. We demonstrate that area-based algorithms have a critical advantage over point-based approaches because they are better able to describe localization uncertainty, which is a common theme across WLAN based localization systems. Next, we present two novel area-based algorithms. To evaluate area-based approaches, we introduce several new localization performance metrics. We then evaluate the two algorithms using our metrics with data collected from our local WLAN. Finally, we compare our area-based algorithms against traditional point-based approaches. We found that using areas improves the ability of the localization system to give users meaningful alternatives in the face of position uncertainty.

1. Introduction

Recent years have seen intense research investigating using Wireless Local Area Networks (WLANs) as a localization infrastructure. The primary motivation has been a dual use one: reusing the existing WLAN investment would provide a tremendous cost and deployment savings over using a specific localization infrastructure, such as ultrasound or infrared based systems.

Previous WLAN localization work uses a traditional *point-based* approach to localization. In these approaches, the localization goal is to return a single point for the object; in our case, a WLAN-enabled mobile device. In this work, we introduce a novel *area-based* presentation approach for WLAN localization systems. The goal in an area-based localization system is to return the *possible locations* of the mobile object as an area or volume (areas and volumes are interchangeable from our perspective) rather than a single location.

The advantages of area-based approaches arise from the fundamental uncertainty resulting from probabilistic approaches used by WLAN localization systems. Area-based systems are better able to utilize and describe

this uncertainty in a meaningful manner as compared to point-based approaches.

One advantage is that area-based approaches can direct the user in his search for an object in a more systematic manner as compared to point-based approaches. For example, when looking for lost keys using an area-based approach, the user can begin his search for the keys in the most likely area, then continually expand his search to the next most likely area. The user thus searches in a systematic manner in relation to the likelihood of the object's presence in the area.

A second advantage of area-based localization is that it presents the user an understanding of the system's confidence in a more natural and intuitive manner than a point-based approach. The larger the area, the less confidence the system has in its placement of the object. The user can adjust his expectations accordingly, which is difficult using a point-based approach.

The critical property that area-based systems exhibit in dealing with the uncertainty is their ability to trade accuracy for precision. Intuitively, localization accuracy is the error between the estimated position and the object's true position. For area-based systems, we define accuracy as the distance the object is from the returned area. Precision describes the size of the area. A point is infinitely precise, but may not be very accurate. On the other hand, the area containing the entire scope of the localization system (e.g. the whole building) would have a high accuracy but poor precision.

Our area-based approach demands new metrics to describe how well the WLAN-based system localizes objects. We thus provide mathematical rigor to the intuitive concepts of accuracy and precision. We also provide several variations on these two metrics that allow us to compare area and point-based approaches. We then show how these concepts can be used to evaluate various localization approaches.

We use two results to show the utility of our area-based approach. The first is a localization system with a much higher error along one axis as compared to another. In our case, this was due to a combination of the building

environment and base-station placement. An area-based approach makes the localization system’s bias in one dimension quite obvious. Such a bias is difficult to recognize in a point-based system. An area-based approach would allow users to identify and correct the problem much faster than a point-based approach would.

The second example is a higher level room-based placement algorithm layered on top of a basic localization system. Here, the area-based approach provides a simple way to expand the set of possible rooms containing the object beyond what is possible using a point-based approach. The user might be willing to accept a localization answer as an ordered set of rooms rather than always returning a single room. Such a response makes sense even if the accuracy of the first room in the set is lower than a point-based approach because the greater number of alternatives increases the likelihood of finding the object.

The rest of this paper is organized as follows. We compare and contrast our work to related work in Section 2. In Section 3 we introduce two area-based algorithms. Section 4 introduces the evaluation metrics. In Section 5, we evaluate our algorithms. Finally, in Section 6 we conclude.

2. Related Work

Recent years have seen tremendous efforts at building small and medium scale localization systems using WLAN technologies, especially using 802.11 and signal strength [1, 2, 4, 6, 7, 9, 10, 12]. The two fundamental building blocks of these works are: (1) a classifier to relate an observed set of signal strengths to ones at known locations, and (2) a propagation function relating 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. However, all of the above systems used the classifiers and propagation functions to arrive at points rather than areas.

Our area-based probability algorithm extends the Bayesian probabilistic approaches used in [7, 9, 12]. However, those algorithms only return the single highest probability location. Returning a single location can be misleading, because most of the time, many locations are equally likely. In addition, our area-based algorithms offer the flexibility of specifying a user-defined confidence level on the returned area.

There is a rich history of localization approaches in the robotics community [8, 11]. These approaches are similar to ours in that they build probabilistic maps of the location of the localized object, in this case, the robot. However, these maps are only intermediate structures used ultimately to define a single point of localization for the

robot, rather than as final end-results presented to users.

3. Area-based Algorithms

This section presents our two area-based localization algorithms. We begin with an illustration of the differences between the output of our algorithms and single-location based localization. We then introduce terminology and definitions. Next, we describe an interpolation technique we used for both areas-based algorithms. Finally, we detail the Simple Point Matching (SPM) and Area-Based Probability (ABP) algorithms.

3.1. Example Areas

Figure 1 shows 2 sample returned areas for a floor in our computer science department. (Figure 5 shows a more detailed map). The presented areas are shown by a dark color. The true location of the mobile object is shown as a “*”. The smallest circumscribing circles and rectangles are also shown.

In the case of Figure 1(a), the area shows the localization can contain the object to an area the size of a single room. In the case of Figure 1(b), the localization is more diffuse, in this case spanning a printer room and a kitchen. The figure also points out a limitation of the environment for our set-up. In general, the localization error is much higher along the y -axis than the x -axis, suggesting changing the base-station placement to be less along a horizontal line.

The circumscribing circles show that augmenting a point with a distance to describe the uncertainty, in point-based localization algorithms, would likely return a much larger area than a strictly area-based algorithm. For example, in Figure 1(a), the circle extends into the local offices, while the rectangle is contained mostly within the hallway (which is the ‘room’ in this case). In Figure 1(b), the circle extends well out of the kitchen and printer rooms into the surrounding hallways. Returning rectangles, while reducing the inaccuracy of circles, no longer fits the definition of a point-based approaches, however.

3.2. Terms and Definitions

There are n Access Points (APs) on the floor, AP_1, \dots, AP_n . The offline measured signal strengths and locations an algorithm uses is called the *training set*, T_o ; traditionally this is called a radio map [1]. A training set consists of a set of *fingerprints*, \bar{S} , that are vectors of signal strength readings, one per AP, along with the (x, y) location where each fingerprint was collected. I.e., $T_o = \{[(x_i, y_i), \bar{S}_i]\}, i = 1 \dots m$, where m is the total number of fingerprints. The fingerprint $\bar{S}_i = (\bar{s}_{i1}, \dots, \bar{s}_{in})$, where \bar{s}_{ij} , is the expected average signal strength from AP_j . Additional fingerprints can be generated via interpolation on T_o .

To generate a fingerprint via sampling, we read a series s_{ijk} of signal strengths at location i ((x_i, y_i)), with a

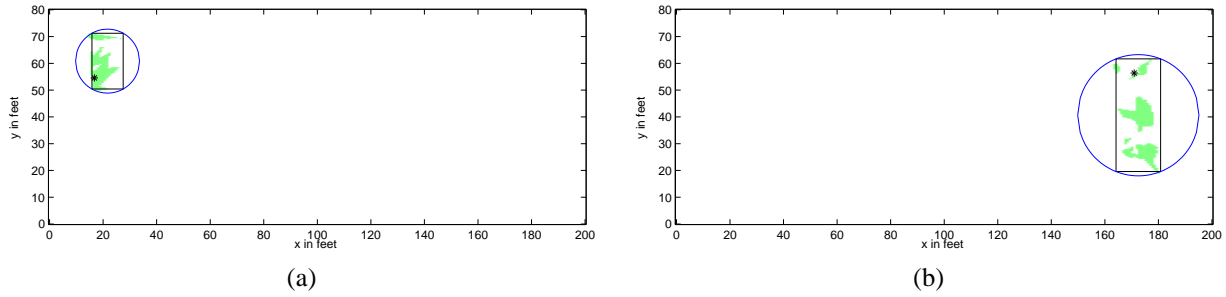


Figure 1. Sample areas returned by our presentation vs. single-location based presentation. The true location is marked by a “**”.

constant time between samples, generating a *sample set*, which is a series, $k = 1 \dots o_{ij}$, o_{ij} is the number of samples from AP_j at location i . We then estimate each $\overline{s_{ij}}$ by averaging the series from AP_j . We follow the same procedure to estimate the fingerprint vector of the object to be localized, \overline{S}_l . Other parameters that we use is the standard deviation, σ_{ij} and the variance, v_{ij} , of the set.

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. An RSS also maintains a standard deviation of the sample set.

3.3. Interpolated Map Grid

An Interpolated Map Grid (IMG) is an interpolation that extends the size of T_o to cover the entire floor. The floor is divided into a regular grid of equal sized tiles. The center of the tile is representative of its location. Building an IMG is a core step of our 2 sample area-based algorithms.

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. Building an IMG is thus similar 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.

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 interpolate $\overline{s_{ij}} = f(x_i, y_i)$ at each tile i for each AP_j using the “height” of the triangle at the center of the tile. Figure 2 shows a sample IMG.

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 and large

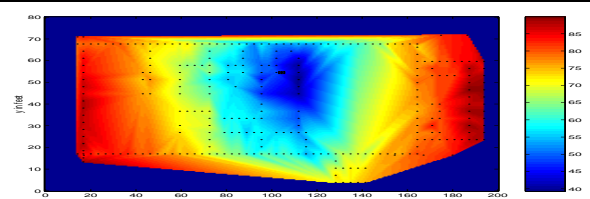


Figure 2. An example IMG for one AP. The surrounding dark area is not part of the grid. The 286 dots show the location of the training samples. Shades show the expected signal strength.

30×30in tiles, observing almost no effect)

The accuracy of the IMG can be affected by both the number of fingerprints used and their location. To gauge these effects, we varied the number of points used in T_o from 30 - 250 fingerprints. We found the impact of sample size was weaker than what we expected, with little difference in localization performance going from 30-115 points, and no difference using more than 115 fingerprints. This is an encouraging result because 30 points is tractable to collect by hand or with deployed sniffers.

We also used different methodologies for picking the fingerprint locations, such as uniformly distributed random, and uniform grid with different grid sizes. As long as the samples are uniformly-distributed over the floor, but not necessarily uniformly-spaced, the specific sampling methodology had no measurable effects.

3.4. 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, and then return the tiles that form the intersection of each AP’s set.

Figure 3 describes SPM in pseudocode. The *Grid*, $\Sigma = (\epsilon_1, \dots, \epsilon_n)$, *FT*, and *Area*, correspond to the IMG, a vector of the expected signals’ standard deviation received from each *AP*, a set of all the tiles on the floor, and the returned area, respectively.

More formally, SPM first finds n sets of tiles, one for each $AP_j, j = 1 \dots n$, that “match” all fingerprints

$\bar{S}_l = (\bar{s}_{l1}, \dots, \bar{s}_{ln})$. 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 $q, 2q, 3q, \dots$, until a non-zero set of tiles results. Even in the worst case, a non-empty intersection will result, although q may expand to the dynamic range of signal readings.

An important issue is how to pick the q for each AP_j . An intuitive way is to pick the expected standard deviation of the signals received from AP_j , which varies from an access point to another. However, it may also vary for the same access point, depending on the strength of the signal. We intentionally used an ad-hoc approximate value for this parameter, which was computed by picking an approximate upper bound on the standard deviation, ϵ_j , for all $\bar{s}_{lj} \in T_o$, for each AP_j , separately.

We varied the value of the noise level to gauge SPM’s sensitivity to this parameter. We found that our results are quite insensitive to q when the upper bound, ϵ_j , is close or higher than the maximum of $\{\sigma_{ij}\}$. However, when ϵ_j is much higher it tends to yield larger areas. Such a behavior is expected; since lower probability tiles tend to be matched at high noise levels.

SPM bears similarities to the Maximum Likelihood Estimation (MLE) [5], when we assume that the APs are independent. Specifically, when we try to localize a new object, the set of signals received from each AP_j of that object during the time window, $\{s_{ljk}\}, k = 1 \dots o_{lj}$, can be estimated by a Gaussian distribution centered around \bar{s}_{lj} , with variance equals to v_{lj} . Therefore, a $(1-\alpha)100\%$ confidence interval for the estimator equals $\bar{s}_{lj} \pm z_{\frac{\alpha}{2}} \times \sigma_{lj}$, where $z_{\frac{\alpha}{2}}$ is a constant that depends on α , e.g., it equals 1.96 for a 95% confidence interval of the estimator [3]. Clearly, for single-mode distributions, such as Gaussian, increasing the confidence level, $1-\alpha$, increases the width of the estimator’s interval at the cost of adding less probable values to it. I.e., less confidence is indeed better in our context; since higher probability values are the only ones included in the interval.

Although the Gaussian approximation assumption may not be true in general [7], it works well in practice as will be shown in Section 5; since we over-estimate the noise level to overcome this limitation. It also simplifies the computations dramatically.

Compared to SPM, the noise level corresponds to $z_{\frac{\alpha}{2}} \times$

```

input Grid,  $\Sigma$ , FT output Area
 $\forall j = 1 \dots n, noise[j] = 0$ 
loop until Area  $\neq \phi$ 
  Area = FT
  for  $j = 1$  to  $n$  begin
    noise[j] = noise[j] +  $\Sigma[j]$ 
    candidateTiles[j] = findTiles( $\bar{s}_{lj} - noise[j], \bar{s}_{lj} + noise[j]$ )
    Area = Area  $\cap$  candidateTiles[j]
  end for
end loop
return Area

```

Figure 3. The SPM algorithm.

```

input L,  $\bar{S}_l$ , conf output Area
 $\forall i = 1 \dots L, H(i, 1) = L_i, H(i, 2) = P(\bar{S}_l | L_i)$ 
 $c = \sum_{i=1}^L H(i, 2)$ 
 $\forall i = 1 \dots L, H(i, 2) = \frac{H(i, 2)}{c}$ 
sort descending on second column(H)
Area =  $\phi, k = 1, prob = 0$ 
loop until prob  $\geq$  conf
  Area = Area  $\cup H(k, 1), prob = prob + H(k, 2)$ 
  k = k + 1
end loop
return Area

```

Figure 4. The ABP algorithm.

σ_{lj} . SPM eagerly attempts to find the appropriate (lowest) confidence level $(1-\alpha)$ for each AP_j that yields an overall non-empty area. It pretends that, for each AP_j , “any” collected fingerprint follows a Gaussian distributions with a standard deviation ϵ_j equals to the highest σ_{ij} , among all the fingerprints in T_o . Therefore, it starts searching by adding a noise level of ϵ_j , then $2\epsilon_j$, and so on, till a non-empty overall area is found.

3.5. Area Based Probability

The strategy used by the Area-Based Probability (ABP) algorithm 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 runs with confidence level α is ABP- α . ABP’s approach to finding a tile set is to compute the likelihood of an RSS matching a fingerprint for each tile, and then normalizing these likelihoods given the prior conditions: (1) the object must be on floor, and (2) all tiles are a-priori equally likely. ABP then returns the top probability tiles whose sum matches the desired confidence. The confidence controls the accuracy-precision trade-off. ABP thus stands on a more formal mathematical foundation than SPM. Figure 4 summarizes the ABP algorithm.

In ABP, signals received from different access points are assumed to be independent. For each $AP_j, j = 1 \dots n$, the sequence of received signal strengths $s_{ijk}, k = 1 \dots o_{ij}$, at each (x_i, y_i) in the original training data, T_o , is modeled as a Gaussian distribution. Again, although this assumption is not generally

true, it significantly simplifies the computations with little performance degradation. A cross-comparison with the other approaches, that model the distribution exactly, is discussed in Section 5. We compute the “mean” parameter of the distribution, $\overline{s_{ij}}$ for each fingerprint in the original training data as before, we then use the IMG to find the mean at each tile on the floor. For each AP_j , we assume that the standard deviation of the distribution at all the tiles on the floor is constant, and equals to ϵ_j . The latter is computed following the procedure described in Section 3.4.

The algorithm then works as follows. Using Bayes’ rule, we compute the probability of being at each tile’s location, L_i , on the floor given the fingerprint vector $\tilde{S}_l = (\overline{s_{lj}})$ of the object to be located, as follows. Note that we use the whole vector for computing the probability, i.e., signals from all the access points, substituting a value of -92 dBm for missing signals, when the location is not covered by the access point.

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

However, $P(\tilde{S}_l)$ is a constant c . Moreover, given we do not have prior knowledge about the exact object’s location, we assume that the object to be localized is equally likely to be at any location on the floor, i.e., $P(L_i) = P(L_j), \forall i, j$. Therefore, Equation 1 is rewritten as:

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

Without having to know the value c , we can just return the location (tile) $L_{max}, L_{max} = \text{argmax}(P(\tilde{S}_l|L_i))$. Specifically, we compute $P(\tilde{S}_l|L_i)$ for every tile i on the floor, using our Gaussian distribution assumption. Up to this step we follow the traditional Bayesian approach with the exception of using the Gaussian assumption.

ABP extends the above approach 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., $\sum_{i=1}^L P(L_i|\tilde{S}_l) = 1$. Given the resulting density, ABP returns the top probability tiles up to its confidence, α . In other words, we return the top probability locations, such that their overall probability is $\geq \text{confidence}$. Those tiles form our returned area.

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 useful tile set insensitive to small changes in α . A more detailed discussion of the effect of the confidence on the returned area is given in Section 5.

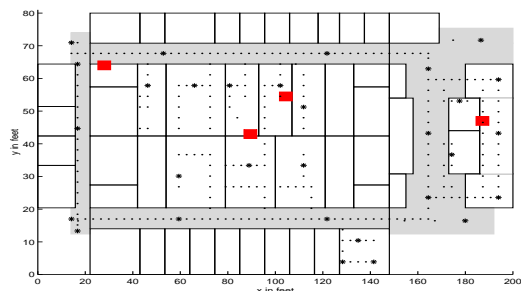


Figure 5. The floor plan and experimental setup. Grey spaces are corridors, white spaces are offices or laboratories. Squares show the locations of the APs, small dots the testing set and large dots the training set.

4. 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 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.

Distance Accuracy. This metric is the distance between the true tile and tiles in the returned set, 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 error metric (i.e. distance between the returned point and true location), although one should look at both the minimum and maximum distances.

Precision. The overall precision refers to the size of the returned area, i.e. the ft². To normalize the metric across environments, it can be expressed as a percentage of the entire possible space (i.e. the size of the floor).

Because many indoor WLAN localization applications can operate at the level of rooms, we also extend the accuracy and precision metrics 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, as follows.

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

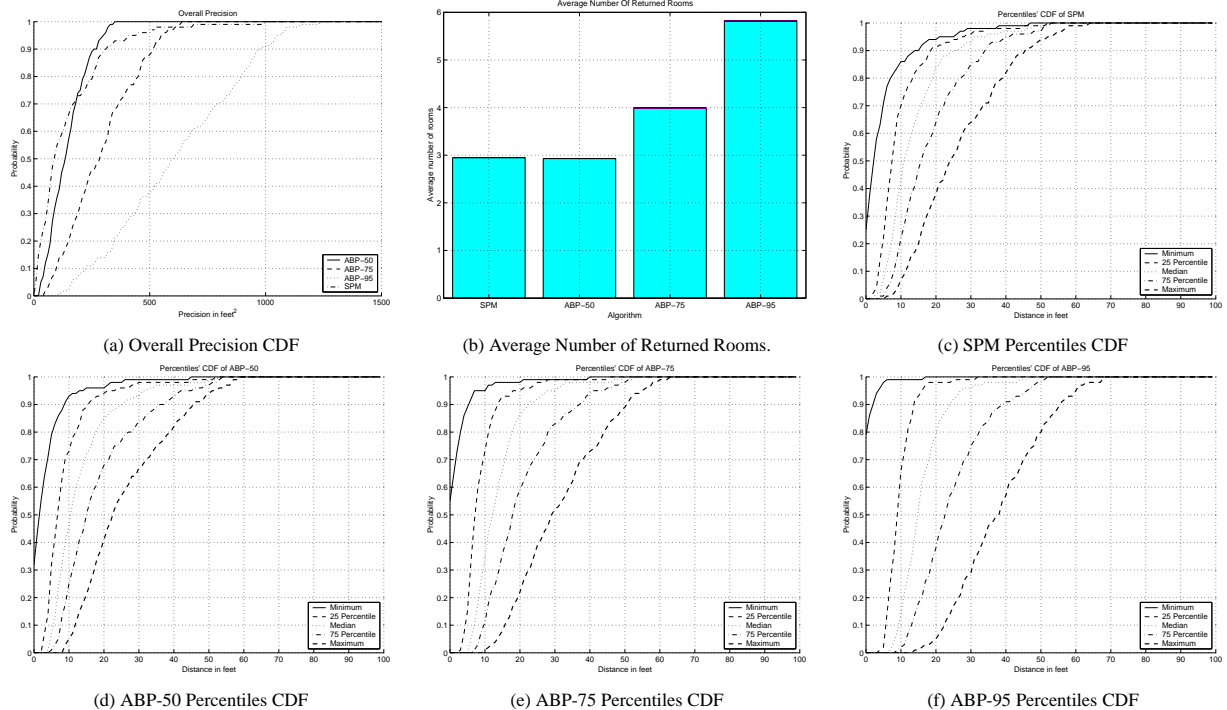


Figure 6. Area-based localization performance for a training set of size 30. The metrics shown are: (a) precision CDF, (b) room precision, (c)-(d) distance accuracy percentiles.

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.

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 returned rooms.

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 we have a few, but large, rooms on the floor, it can be expressed as a percentage of the total number of rooms.

5. Performance Evaluation

In this section we evaluate our algorithms. After describing our experimental methodology and set up, we first characterize the performance of our area-based localization algorithms using area-based metrics. We then compare their performance against traditional single-location based approaches.

5.1. Experimental Setup

Our methodology is to collect a large number of fingerprints and divide them into a *training set* and a *testing set*. To evaluate an algorithm, we localize points in the testing set using the training set as input to the algorithm. The assumption is that the training set was collected offline, either by hand-sampling or deployed sniffers, while the testing set represents localization opportunities. We then quantify the distributions for the metrics over all the localization attempts.

We collected localization fingerprint vectors from 286 locations, over a period of two days, on the third floor of the Computer Science department building. This floor is roughly 200 feet by 80 feet. The floor includes more than 50 rooms. There are 4 Access Points (APs) deployed on the floor as part of the school’s wireless LAN, which uses the IEEE 802.11b standard. The mobile equipment, used for collecting the data, is a Dell laptop equipped with an

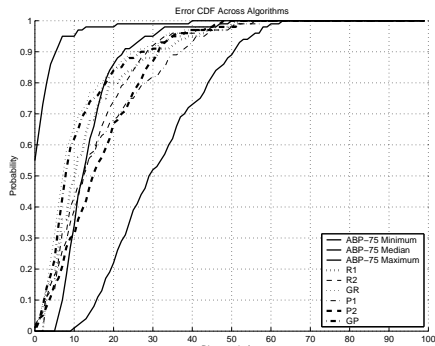


Figure 7. Comparing the error of area-based algorithms with point-location based ones.

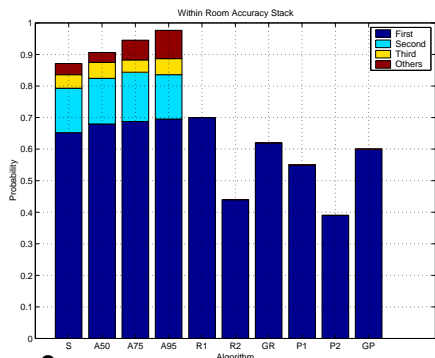


Figure 8. Comparing the n-room accuracy of area-based algorithms with point-based ones.

Orinoco Silver card by Lucent Technologies. Figure 5 is a map of the rooms, fingerprint locations, and APs. To collect the signal strength information at each location, `iwlist scan` is triggered for 60 times with a 1 second interval. This effectively reports the signal strength information every second over a window of 1 minute.

For the noise level, we found that the fingerprints from a specific AP on our floor tend to have a higher variance. Using our methodology to approximate the noise level, described in Section 3, we arrived at a value of -3 dBm for ϵ_j for the first 3 APs, and a value of -5 dBm for the fourth.

5.2. Performance of SPM and ABP

In this section we evaluate the performance of our two proposed area-based localization algorithms.

Figure 6(a) shows the CDF for the precision. Recall that the floor size is roughly 160,000 sq.ft., so it is almost guaranteed that the precision is below 3% of the floor size for SPM, ABP-50, and ABP-75, and below 10% for ABP-95. Our results for the above 4 algorithms showed that average precisions to be 1%, 1%, 2% and 4% of the floor area, respectively. As expected, ABP returns a higher precision (larger area) as the user-specified

confidence level goes up; since confidence improvements come at the price of higher precision. Specifically, more less-probable tiles are incorporated into the returned area. SPM tends to return lower precision results, performing somewhere between ABP-50 and ABP-75. In these particular experiments, its performance is similar to ABP-50. Figure 6(b) shows the precision in terms of the number of returned rooms.

Figure 6(c-f) depicts the CDFs of the distance accuracy. As the confidence increases, the minimum distance between the area returned by ABP and the true location decreases, while the maximum distance increases. When the confidence goes up, it is equally likely that the extra added tiles are either closer to, or far from, the true location. This is confirmed by observing that the intermediate percentiles (25%, median and 75%) remain almost unchanged.

The first 4 bars in Figure 8 show the n-room accuracy for SPM and ABP. The “others” bar corresponds to cases where the true location is not inside the top-3 returned rooms, but is still in some returned room. In other words, it is in one of the returned rooms, but not in the top-3. The majority of the testing locations are guaranteed to fall into the top 3 rooms returned by SPM and ABP. Increasing the confidence level has a little impact on the first, or even the top 3 room accuracy. It tends to improve the overall accuracy though; when less probability tiles are incorporated into the returned area, it is expected that some of them will eventually fall into the targeted true room, and consequently the number of misses is reduced. The overall room accuracy of the algorithms corresponds to the height of the stack $\times 100$. Based on our experiments with different training sizes, it tends to improve as the size of the training data increases, without significant degradation in the overall precision.

The percentage the correct tile is returned can be concluded from the min distance CDF in Figure 6. Specifically it corresponds to the intersection of the minimum curve with the y -axis. The latter is the probability that the distance between the true location and the returned area is 0. The low value for that metric indicates that most of the time the returned area is very close to the true location but does not necessarily overlap it.

5.3. Comparison with Single-Location Based Presentation

We now compare our area-based approach with the two traditional point-based approaches: RADAR and Probabilistic. To compare approaches, we combine the traditional accuracy performance metric with our distance accuracy metric as well as our room accuracy metric. Space constraints prevent us from giving more than a brief description of the algorithms here; consult the references for a full description.

The first point-based algorithm we used is the well

known RADAR [1], which we refer to as **R1**. Its approach is to return the location of the closest training fingerprint to the RSS, 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 fingerprints; our averaged RADAR algorithm, **R2**, averages the closest 2. 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 [9] to select the most likely training fingerprint. 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.

Results Figure 7 shows the traditional error CDF, as well as the distance accuracy CDF for the min, median, and max tile percentiles. The error CDFs for the traditional point based algorithms fall in-between the minimum and maximum CDFs for the area-based algorithms. In fact, they all perform more or less the same. Their error usually closely matches the median CDF of our area-based approach. We only show results for ABP-75, because all our approaches tend to perform similarly in terms of the median percentile accuracy as shown in Figure 6. Interestingly, the figure shows that increases in the returned area come from expanding the area in both closer to and farther away to the true location from the median; although the shapes tend to be irregular, the min percentile shows they often overlap the true location.

Figure 8 shows the n -room accuracy as a bar graph. For the point-based approaches, they only return the room enclosing the returned point. The area based presentation do not outperform single-location presentation in terms of the 1-room accuracy. Yet, they offer alternatives, and thus give much higher overall accuracy. In contrast, when the true room is missed in point-based algorithms, the user has no additional information as to which room to try next.

6. Conclusion

In this work we demonstrated the utility of area-based presentations for WLAN localization systems. We argued that the area-based results present users a more intuitive way to reason about localization uncertainty than point-based approaches. We showed how area-based algorithms can give insights into the localization perfor-

mance that are much more difficult to do using point-based systems.

We presented 2 novel area-based algorithms along with several performance metrics and compared them to traditional point-based approaches. We found that qualitatively, the accuracy of all algorithms was quite similar. However, the area-based approaches were useful for expanding the possible location where an the object might be found. We demonstrated this function of our area-based algorithms by showing how they can return an ordered set of rooms, and that the room set is quite useful in increasing the set of likely locations for the object.

The similarity of all the algorithms’ accuracy points to interesting work examining if any algorithm has a substantial advantage. While our small study can not answer such a question, our results show that it may be the case that all algorithms perform similarly. In that case, simple algorithms which do not require much sampling (e.g. SPM) are preferable to more complex or those that require a high sample density.

References

- [1] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *INFOCOM*, March 2000.
- [2] R. Battiti, M. Brunato, and A. Villani. Statistical Learning Theory for Location Fingerprinting in Wireless LANs. Technical Report DIT-02-086, University of Trento, Informatica e Telecomunicazioni, Oct. 2002.
- [3] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury Press, Belmont, California, 1990.
- [4] Ekahau, Inc. The Ekahau Positioning Engine 2.1. Whitepaper available from www.ekahau.com, July 2003.
- [5] D. Hand, H. Mannila, and P. Smyth. *Principles Of Data Mining*. The MIT Press, 2001.
- [6] 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.
- [7] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-based location sensing using wireless Ethernet. In *MOBICOM*, Sept. 2002.
- [8] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, 1988.
- [9] T. Roos, P. Myllymaki, and H. Tirri. A Statistical Modeling Approach to Location Estimation. *IEEE Transactions on Mobile Computing*, 1(1), Jan-March 2002.
- [10] A. Smailagic and D. Kogan. Location sensing and privacy in a context aware computing environment. *IEEE Wireless Communications*, 9(5), Oct. 2002.
- [11] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1-3):29–53, 1998.
- [12] M. Youssef, A. Agrawal, and A. U. Shankar. WLAN location determination via clustering and probability distributions. In *PerCom*, Mar. 2003.