Cost and Accuracy: Factors Concerning Various Indoor Location Estimation Methods

Alan Liu, Tyler Robison
Department of Computer Science & Engineering
University of Washington, Seattle, WA 98195
{aliu,trobison}@cs.washington.edu

ABSTRACT
There are tradeoffs between cost and accuracy among the myriad of approaches to indoor localization. In this paper, we explore a number of approaches aimed at reducing overall cost of deployment and use in the form of equipment, infrastructure, effort, and algorithm tuning requirements.

PROBLEM
Indoor localization is the process of determining a mobile computer's location inside a building. We investigate the accuracy of various indoor localization schemes that use 802.11 (WiFi) access point (AP) broadcast beacons. These schemes differ from each other in numerous dimensions, such as software requirements, assumptions of the environment, and cost. In turn, these dimensions affect the feasibility and usefulness of the schemes. Designers of location-aware systems must choose a scheme that provides suitable localization for their location-aware applications while minimizing the resources necessary for it. Therefore, this paper will also discuss the significance of the dimensions and the impact of different choices in this design space.

MOTIVATION
There are many reasons why knowing the location of a device indoors could be useful. For instance, a system supporting location-aware reminders would allow a user to set reminders relevant to their location, such as "pick up your printouts" when near the printer closet [CB]. Another system that could leverage this positioning functionality is a location-aware buddy list, where users can choose to reveal their location to friends and associates. An obvious use would be to aid someone unfamiliar with the building layout in navigation. This is something that would be invaluable to the cognitively impaired, who often have trouble navigating indoors, even in buildings they frequent. Other applications that could utilize this knowledge might enhance the interactivity of a "smart" environment.

Of course the use of 802.11 beacons is not the only way to locate a mobile computer. An alternate approach might be to use RFID readers placed near locations where localization is desired. The readers could detect the proximity of an RFID-tagged object and inform some system providing location information. However, this would have issues over privacy and instrumentation cost. 802.11 is an attractive option because all computation can take place locally on a client, so as long as a client chooses to remain silent, its location cannot be known by the infrastructure or any other clients. In addition, in many buildings 802.11 infrastructure already exists, while the technologies used by other schemes are less common.

There are two major aspects of indoor localization with WiFi that are not found in outdoor WiFi localization. The first is the need for higher accuracy. The second is the added dimension of altitude (or logically, floor).

While 30-75 meter accuracy can still be useful outdoors [OpKn], such accuracy would be nearly useless indoors as many buildings are not much larger. To achieve room-level accuracy, the localization scheme must have errors of only 3-5 meters. Without room-level accuracy, indoor location-aware applications, such as those previously described, would be severely limited. One would not be able to place location-aware reminders to specific rooms. The location-aware buddy list would not be able to tell what room a person was in, which creates a problem with how it should represent a user's location in a terse, text format. If a navigation system cannot determine to a high degree of accuracy whether its user is at a room, it would place a burden for figuring that out on the user, which could pose to be too difficult for users who are cognitively impaired (who are also those who would benefit the most from such a system). So accuracy is critical for many location-aware application functions.

Floors

WiFi localization schemes for the outdoors do not concern themselves with determining altitude, because it is unlikely that two locations will be at the same two-dimensional coordinate but separated by different altitudes. However, the notion of a location containing the third dimension is natural to multi-story buildings. This added dimension poses a challenge to localization schemes because 802.11 signals may not change a great deal across a floor, but the penalty for estimating the wrong floor may be even worse than estimating the wrong room on the right floor. If a user must correct an error in a floor estimate, for instance to find someone she needs to talk to, she may need to visit several floors, which is significantly more time- and effort- consuming than checking the vicinity of an estimate on one floor.

In fact, there are applications that could use a good floor estimate even when the 2-D estimate is imprecise. Location-aware computing researchers have noted that privacy is a major concern with applications that reveal users' locations. One approach that deals with this is to obscure the location by reducing its resolution [Gru]. A natural way to reduce the resolution of an indoor location is to simply drop 2-D coordinates and report only the floor. This could be used by an application such as the location-aware buddy list, where a user may wish to restrict the resolution of her location to certain buddies while still providing a useful indicator of her whereabouts.

RELATED WORK

There has been quite a large body of previous work on a range of positioning systems, each which use its own localization method(s).

GPS

The most notable positioning system is the Global Positioning System (GPS). GPS enables users to determine their locations using a GPS receiver, without the need to subscribe to a service. This model, where the system merely broadcasts signals, is very similar to the way 802.11 APs broadcast their identities, without the need for the user to initiate any communication channels. Privacy in this system is maintained because the user can prevent the infrastructure from tracking them, while still being able to determine their own location.

Unfortunately, GPS requires clear line of sight to several satellites in the sky, which is something that does not work indoors without expensive GPS repeaters.

A GPS receiver triangulates its distance to the GPS satellites using signal times of flight. High accuracy is maintained only if the receiver can synchronize its time with the GPS satellites. 802.11 does not support this tight timing, so we cannot simply use GPS' localization scheme.

Active Bat

Active Bat makes use of ultrasound devices mounted on (or above) the ceiling in order to intercept the signals of an ultrasound emitting device carried by the user, or object, to be localized [Bat]. These ceiling mounted devices all connect to a centralized board which uses their readings to interpret the Bat's position. As with other techniques, this localization requires substantial deployment, here in terms of ultrasound receiving devices and the calculation board. In addition, because the calculation takes place outside of the object being localized, there is no guarantee on privacy. However, the system provides a very high degree of accuracy and precision.

SpotON

SpotON provides excellent ad-hoc localization, but with the drawback that specific hardware is required [Spot]. In addition, these SpotON tags need to be placed everywhere that localization is to occur, and while the system is flexible in that these tags can be relocated to other areas, doing so would require significant effort on the part of the user. SpotON may be a good solution for many situations, but here we would like to develop a solution using existing wireless networks.

WiFi

The major approaches to WiFi localization have been triangulation [PL], modeling [Rad], and fingerprinting [Rad,LL,HEM]. Triangulation relies on the assumption that signals radiate spherically, so the actual location would be at the centroid of the APs heard. This assumption is fine for outdoors use simply because the typical range of most APs is less than 100m, thus upper bounding the error. Both modeling signal propagation and fingerprinting have achieved much better accuracy than triangulation, but they do so by using samples collected in the areas where localization is desired because signal propagation varies widely across different environments. Therefore their generality is questionable.

Although existing WiFi localization schemes have achieved room-level (3-10m) accuracy, only Haeberlen et. al. have done so across multiple floors in a building [Rad,HEM]. None of the existing work treats floor determination as a more important dimension to localize than 2-D position, therefore there has not been any specific treatment of how well floors are localized.

The best existing WiFi localization schemes have an average error of 3-10 meters [Rad, HEM]. Elnahrawy et. al. suggest that this is the physical limit to localization using signal strength, but also that simpler algorithms may be able to achieve similar accuracy and precision [Lim]. Despite this finding, the majority of schemes rely on sophisticated methods for matching newly observed signal strength signatures to a known set of signature-location pairs (fingerprinting). However, using fingerprints is not without downsides. Fingerprinting requires a large amount of up-front data collection. In lieu of mainstream robotic mapping systems, human effort on the scale of one minute per square meter must be expended. Although this task can be parallelized to reduce the overall time necessary in collecting fingerprints, there is no reducing the total cost of effort. Not only is there an initial cost for setting up a fingerprinting system, but there is also the cost to periodically recalibrate and/or update fingerprints when something in the environment changes in a way that affects signal propagation. This can occur more often than desired simply because the 2.6GHz frequency band that 802.11 uses is sensitive to many environmental factors.

WALRUS

WALRUS is the result of an undergraduate project for the UW course CSE 477 [WA]. WALRUS uses simultaneous ultra-sound and 802.11 beacon broadcasting to determine the room. The idea behind WALRUS is to work around the fact that 802.11 signals can travel through walls, while ultra-sound is usually dampened by them and can therefore be a better indicator of proximity. Since light travels faster than sound, we use the 802.11 beacon not only to carry information about the room, but also as the "lightning" to the ultra-sound's "thunder."

An environment running WALRUS can have a much higher level of confidence for room-level positioning, but requires that every room have an access point synchronized with an ultrasound broadcasting speaker. Thus the cost of instrumenting WALRUS in a building would be significantly higher than any of the previous methods, which rely only on existing WiFi infrastructure. An open question will be whether we can create a system that can use a partial WALRUS infrastructure, which can provide high confidence positioning in certain rooms, and fall back on one of the other methods of positioning when no ultrasound can be heard

OUR USAGE OF PLACE LAB

Place Lab is a collection of java classes for performing localization using such signals as WiFi and GPS, and is quite open and versatile, and is designed to be expanded. It provides much of the basic functionality needed to read WiFi and GPS signals, and also a number of classes useful in analyzing them; basic particle filters and centroid trackers are provided in the class hierarchy. Just as importantly, Place Lab provides a structure that lends itself to this sort of work, and can easily be extended to incorporate other techniques. A significant limitation of Place Lab, however, is that it is largely intended for outdoor use, and as such has somewhat limited use. This is why we use it as a foundation from which to build an effective indoor localization system.

METHODOLOGY

For purposes of training the bin-system, and in order to evaluate our methods, we collected large amounts of WiFi signal and position data. Because we collect these WiFi signals along with the true position at the time, we have ground truth; we know the exact locations of the wireless device when it received each set of signals. In order to test our localization methods we can read these signal logs as though they were live signals, feed them to our localization procedures, and compare their estimated positions with the known correct position, also stored in the log. The exact procedure will be described in more detail later.

The Place Lab software provides functionality useful for gathering such data: a map of the area is displayed to the user as the WiFi card collects signals, and the user is able to click on this map in order to mark their location; the pixel coordinates of the map are translated to the latitude/longitude coordinate system used for localization. A change of floor is recorded by selecting the new floor via a drop-down menu.

We collected data through this software by walking around each floor of the Allen Center, navigating various walkways, hallways, offices and conference rooms as we went. We also made sure to include numerous floor transitions in our data collection. Each stairwell and elevator was used at least once; most were used repeatedly. Overall we gathered over 10,000 beacon readings over the course of 2 hours, including 14 floor transitions. We logged all detectable beacon signals, not just those originating from CSE APs. The localization schemes we employ use different subsets of the traces depending on the beacon metadata they use – if a scheme requires certain information that we do not have on a particular AP, it is not considered.

In order to evaluate our techniques we, as mentioned earlier, treated this test data as though it was a set of live signals, ran it through the trackers to estimate our position, and finally compared the estimated positions with those recorded as ground truth in the logs.

In order to more accurately evaluate our techniques, we randomly divided the log data into halves; one for training, and one for evaluation. Although this meant less data to train on, it does give us a more realistic evaluation of our methods.

Training was required for several of our methods, specifically the floor histogram as well as both binning algorithms. The training half of the log data was used to prepare these methods offline; no online learning takes place in any of our techniques.

LOCALIZATION SCHEMES

Centroid

The first localization scheme we applied was the *centroid tracking method*. This scheme uses triangulation, so it is simple algorithmically and cheap computationally. In addition, it does not require previous beacon readings, so the frequency at which it is run is unimportant. This may be advantageous for computationally-limited clients who for reasons of spare cycles or power may not want location estimates often.

Place Lab contains a centroid tracker that computes the centroid over all access points whose location is in its database. This works outdoors for the reason previously mentioned – the error is bounded by AP range so in most cases this scheme's accuracy is reasonable.

However, there are some inherent shortcomings with using a centroid approach for indoor localization.

1. Relatively weak signals from APs that are far away (on a building scale) from the client influence this scheme, so we must limit what beacons we calculate the centroid over if we wish to constrain error.
2. The centroid works best when there is a large, evenly distributed set of known APs. The number of possible estimates that a centroid will compute is equal to the cardinality of the power set of known APs. For the scheme to produce more estimates requires more APs.
3. It is impossible to produce a location estimate that is beyond the convex hull of the known AP locations set. APs must be placed such that they surround all desired locations. However, existing AP placement is done without this consideration. The Paul G. Allen Center is an

example of such a topology. This is especially problematic as many offices are along the building perimeter and it is reasonable that a location-aware application would desire office-level accuracy.

4. This problem also affects floor estimation. It becomes much less likely that the scheme produces estimates for the upper or lower floors of a building, because the distance between APs on the same floor is sometimes larger than the distance between APs on separate floors. This leads to cases where APs from nearby floors may be heard more often than APs on the same floor. Since there are no APs below the lowest floors or above the highest floors, when a client on such a floor hears even one beacon from the middle floors, its estimate will be affected.

To address the first shortcoming, we compute the centroid over only the AP locations from the set of stronger beacon readings. It is currently set to take the strongest half of all the beacons heard. The advantage of this is that it is a very minor and simple change.

Only purchasing more APs or moving existing ones can address the second shortcoming. How feasible either is depends on the cost to place a new AP or to move APs while also updating any associated network management policies, which may be the job of network staff and not application developers.

We will investigate the third and fourth shortcomings in the next section. It turns out that the fourth approach is more tractable. We noticed that there is roughly the same number of APs in the same locations on every floor in the Allen Center. Intuitively, an AP on the same floor as a client should have a stronger beacon signal since it travels a shorter distance and will not be affected by floor attenuation. If we leave out weaker beacons from our calculation, a plurality of the beacons left should be on the same floor as the client. This leads to our *floor mode estimation*, which produces a floor estimate that is the mode over strong beacons. It relies on APs to be distributed in the same locations on every floor, which is close to the case in the Allen Center, but not necessarily true in other buildings.

Our results for the centroid with floor mode estimation are shown in Figures 1 and 2. The results used a random sampling of half of our continuous trace. The scheme estimates the correct floor 65 percent of the time, and the majority of 2-D error is within 15 meters.

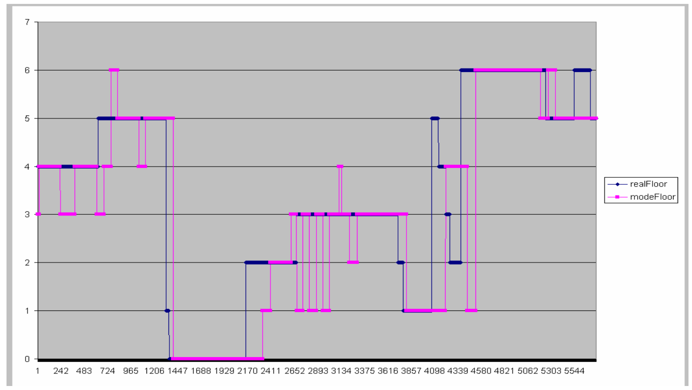

Figure 1. Floor estimate accuracy of centroid tracker

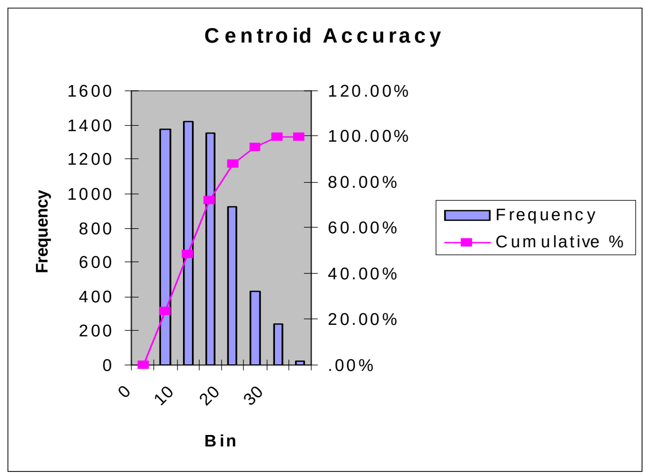

Figure 2. 2-D accuracy of Centroid tracker

Weighted Centroid

We observed that the previous scheme often placed its estimates near the middle of the building. Although the estimates are not severely wrong, they are also not very useful because the center of the building on many floors is empty space. This is an artifact of the third shortcoming of the centroid tracker. We also noticed that certain APs are heard more often because they are centrally located and more often within client range, “pulling” estimates in toward the center. This suggests that a *weighted centroid* scheme where beacons that are less likely to be heard affect the estimate more heavily may alleviate some of the problems with the previous unweighted centroid scheme.

We assigned both 2-D and floor weights to beacons based on their relative distance from the center of the building and limited line-of-sight. The intuition for this is that the line-of-sight of APs near the perimeter of the building is smaller and their beacons are heard less often. Hearing such an AP's beacon should indicate being closer to the AP than others.

The results for floor accuracy are shown in Figures 3. Surprisingly, the weighted centroid scheme does no better than centroid for 2-D error. It also does slightly worse at getting the floor correct, although it does a better job of avoiding localizing to the atrium when the client is not actually there. Unfortunately, it localizes to the sixth floor too easily, suggesting that weighing is not a panacea for floor determination.

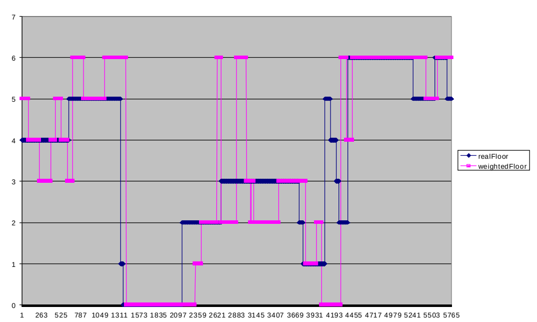

Figure 3. Floor estimate accuracy of weighted centroid

Particle filters

Another localization scheme is *particle filter tracking*. A particle filter is a probabilistic technique used successfully in localization in the face of sensor uncertainty [Rob,Inf,LL]. It consists of:

1. A collection of particles, each representing one position estimate
2. A sensor model that maps how likely an estimate is given sensor readings.
3. A motion model that determines how particles move over time.

At every sensor update, the filter averages over the particles to reach the most likely position. Between updates, it applies the motion model to simulate a client's movement.

Letchner and Limketkai showed that a particle filter can achieve 3 to 10 meter accuracy for localizing in hallways of one floor of the Allen Center [LL]. We wanted to know how the particle filter would do across all accessible areas of the building because the particle filter already exists in Place Lab. To adapt the particle filter for such a purpose, we modified the Place Lab particle filter with simple parameters.

We updated the sensor model to include a floor attenuation factor that accounts for the attenuation of beacon signals between access points on different floors. This is a simple way of dealing with the presence of floors. The floor attenuation factor penalizes estimates based on the signal strength of a beacon from an AP on a different floor from the current estimate. To put it another way, it tries to reduce the likelihood calculation to only 2-D, since Place Lab has an existing sensor model for likelihoods over 2-D distances. When the AP is on a different floor than the estimate, this likelihood is discounted by a multiplicative factor. In lieu of knowledge of how the floors in the Allen Center attenuate 802.11 signals, we chose a multiplier of 0.8 for every additional floor.

We modified the motion model so that all particles may either stay still or move at walking pace and can undergo a transition to a different floor with some probability. These values were not empirically derived but rather assumed. We wanted to see how well it would do without tuning, as tuning requires more work to set up and is inherently specific to the environmental characteristics of a particular building and the movement tendencies of particular clients.

As seen in Figure 4 and 5, the resulting accuracy with these modifications is very poor.

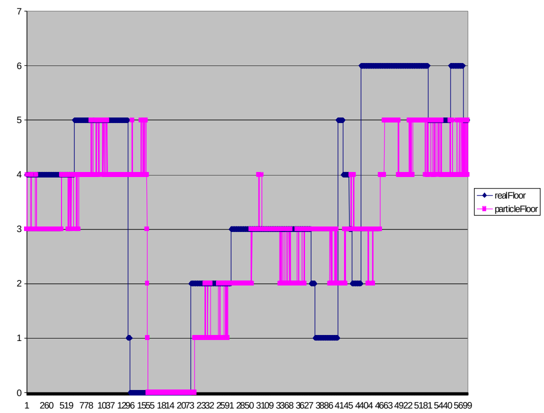

Figure 4. Floor estimation accuracy of Particle Filter

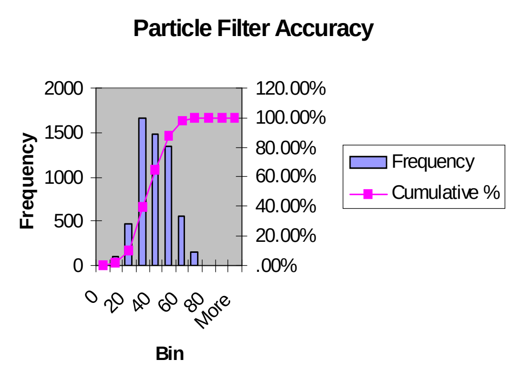

Figure 5. Particle filter 2-D accuracy

It is highly likely that our intuitive parameter values for the previous schemes were not good ones. Unfortunately, this means that training may be unavoidable, despite the higher effort required to collect and maintain training data. With this in mind, we explored several different methods for training. We avoid duplicating pre-existing schemes such as RADAR and Letchner and Limketkai's localization. Rather, we focus on simpler and less computationally expensive schemes. We do so because their performance is less well understood, despite the possibility that they are "good enough" as Elnahrawy suggests [Lim]. If this were the case, then localization becomes feasible for mobile clients with thin computational powers, such as PDAs or even cell phones. This would broaden the platforms that location-aware applications could support.

Histogram for floor determination

We first modify the Place Lab sensor model to emphasize floor determination. The *histogram* approach bins the readings from our trace according to floor and the beacons heard while on that floor. The likelihood that a floor estimate is correct given that the client hears a certain beacon is simply the fraction of times the beacon was heard in the training set when on that floor. We omit binning based on signal strength ranges to reduce one dimension in the bin space.

The advantage of this scheme is that beacon locations need not be known. The disadvantage is that the beacon database must increase in size, both in terms of number of beacons and also amount of metadata per beacon. If there are not too many buildings, this increase in beacon database size is negligible. However, if some unified database among all buildings exists, the size demands would be significant.

Surprisingly again, the results are not very good. The scheme is only right 40% of the time and within 1 floor 82% of the time. Two weaknesses in our model could account for this poor performance. The first is that the particle filter still depends heavily on the motion model, which causes it to be prone to *mislocalization* [LL]. The filter should begin spreading out particles as they become less likely, but we did not observe this as we noticed the standard deviation across particles actually decreased some points when the floor error was high. This suggests that our histogram-based approach may have fundamental problems in determining floor. Ignoring 2-D position may have negatively affected the scheme. We next examine whether taking 2-D position into account also matters.

Binning based on distance

We explore binning based on 2-D distance from AP, ignoring floor difference. Unfortunately we can no longer use beacons from APs with unknown locations. We wish to see if the new requirement for AP locations is worth a benefit. First we try with a bin being 20 meters in size, and the results are in Figure 6 and 7. No surprise, the results show that binning without regards to floor does poorly with floor accuracy, getting the correct floor only 28% of the time. Also unsurprisingly, the average 2-D error is better than the histogram approach that ignored floors, at 13-16 meters. These are not competitive numbers against the centroid tracking methods, but they do not require any specific placement of APs to work.

To investigate whether bin size matters, we then set the bin size to 10 meters. This gives more precise counts in each bin. However, the results actually showed no difference in 2-D error (see Figure 8). This might indicate that the particle filter is not sensitive to small changes in likelihoods, that a change from 20 to 10 meters might not significantly change signal strength, or that the motion model is dictating the location of the estimate more than such a change in the sensor model.

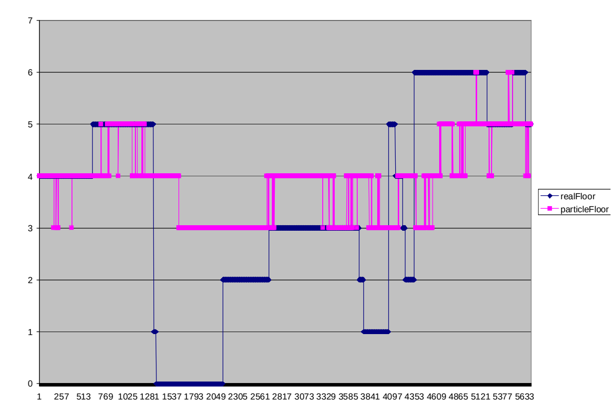

Figure 6. Floor estimation accuracy of binning with 20m-sized bins

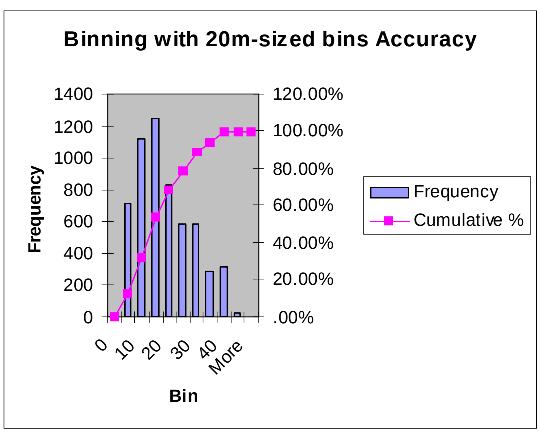

Figure 7. 2-D accuracy of binning with 20m sized bins

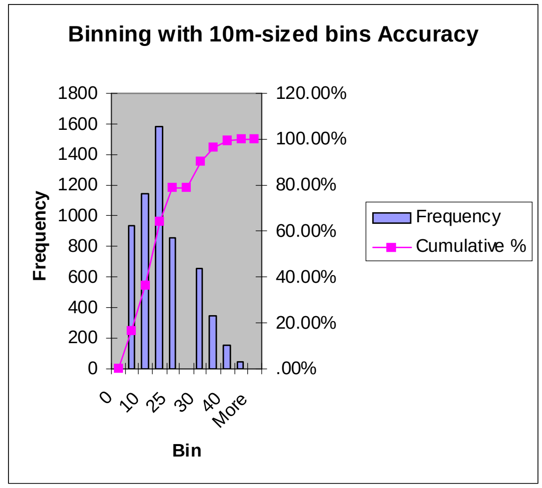

Figure 8. 2-D accuracy of binning with 10m

Binning based on distance and floor difference

Finally, we bin based on both 2-D distance from AP and floor. The results, shown in Figures 9, 10, and Table 10, are promising, as binning on 20 meters and per floor gives a much higher floor accuracy rate of 66%. Interestingly, 2-D error does not change much.

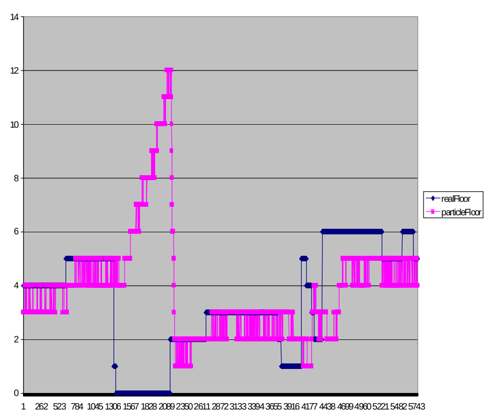

Figure 9. Floor estimation accuracy of floor and distance binning with 20m sized bins

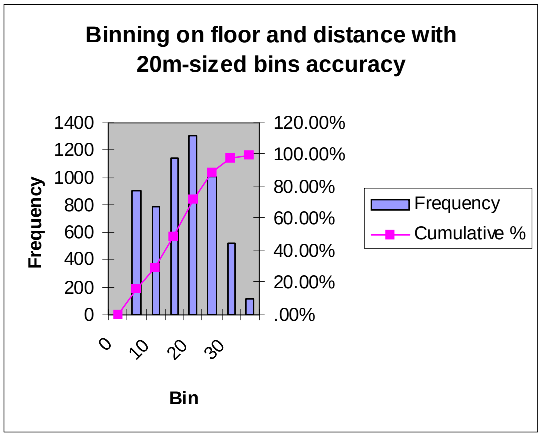

Figure 10. 2-D accuracy of binning on floor and distance with 20m-sized bins

2-D accuracy of floor and distance binning with 20m sized bins	
Mean	14.88514
Median	15.26413
Standard Deviation	7.948047

Table 1. 2-D accuracy of binning on both floor and distance with 20m sized bins

Map based Particle Filter

The map-based approach to localization is founded on the observation that we are not operating in an unknown environment; presumably the map of the area won't change much in terms of its structural layout, and so we can take this into account when estimating our location. Walls, for instance, tend to weaken signals that pass through them, and so if we are given a possible position, a beacon, and a WiFi signal from that beacon, we should be able to count the number of walls in-between the position and beacon on the map, and take that into account when we compute the likelihood of that position. We can also take into account other features of the area, specifically the location of wide-open spaces (such as the atrium in the Allen Center), and areas in which one can transition floors (stairways and elevators), and make use of them in the estimation process. In addition, it should require only a small amount of effort to setup; only specially marked maps and some training data are needed.

There are a few ways to store this kind of information, but perhaps the most straight-forward, most accurate, and easiest to use is simply to take an existing map of the area and mark the walls, stairwells/elevators and atria in certain colors, and that is the approach we take in this project. We use blue to indicate a wall, green to indicate a stairwell or elevator (the two are not distinguished in this project), and red to indicate the atrium. All other pixels in the image are white, indicating that they are not a wall, stairwell or atrium pixel, and no pixels are considered to be wall/stair/atrium combinations. All excess markings (such as room numbers) were removed from these maps. To improve the estimates produced by the particle filter we colored the areas on the maps outside the buildings as walls, as our sensor model forbids particles from existing inside of walls, which is discussed below. This does explicitly restrict our localization to the building itself, but we believe this is a reasonable limitation given the intended application. One of the maps used is shown below in Figure 11.

Figure 11. Map marked for map based particle filter

This map information is utilized in a particle filter to estimate our position; for this project we only apply the map information to a particle filter, although it could potentially be used in conjunction with other techniques instead. The particle filter seems to be the most intuitive choice, however, since we are given a particle, beacon and signal and asked how likely it is that this should occur; it is less clear how the map would be used if you didn't have a hypothetical position in mind to trace to from the beacon. The ways in which the map information is used in the particle filter's sensor and motion models is described below.

Map based sensor model

The sensor model of the particle filter is responsible for taking a particle and a set of sensor readings and generating a likelihood based on these; it comes up with a value indicating our confidence in the particle's position. The standard particle filter supplied by Place Lab is essentially a hand-made, and somewhat crude, decision tree which considers the distance covered and the signal strength, and produces a likelihood based on these. The map based approach we use takes into account wall, atrium and floor information, in addition to the distance and signal strength, and makes use of a more statistically sound method for determining the likelihood.

Whenever a new set of WiFi readings comes in, each particle in the particle filter has its likelihood recalculated based on the new WiFi readings, and it is in this calculation of the likelihood that we make use of the wall and atrium information. To get this likelihood, which has a value from 0 to 1, we find the likelihood for each of the individual readings (our confidence that we would get this particular signal from this particular beacon when our particle is at this particular position) and multiple them together to get the final likelihood for the particle.

$$L_p = \prod_i L_{p,i}$$

That is, the likelihood L_p of any particle p is the product of the likelihoods $L_{p,i}$ of that particle with respect to each beacon i. Calculating these individual likelihoods $L_{p,i}$ makes use of these signal characteristics by comparing them with the training data; this procedure is described below.

When asked to calculate this likelihood $L_{p,i}$, we know the position of the particle, the position of the beacon and the signal strength, and wish to use these to characterize how likely it is that the reading could be heard with this strength at this position. From the particle's position and the beacon's position we can consult the map and see how many walls lie in-between, taking into account the different floor maps traversed in its path, and using similar methods we can see how many atrium pixels are encountered. We can also calculate the distance between the particle's position and the beacon's position, and the difference in floors between the particle and beacon.

With these five pieces of information (signal strength, distance, # of wall pixels on the path, # of atrium pixels on the path, and the floor difference) we can calculate a likelihood by binning; that is, these five features represent a point in a 5 dimensional space, and if we discretize his space, the point will fall into a single 5-dimensional bin. During the training process, described below, we populated these bins with training data points, and the number of training data points in a bin indicates how likely those characteristics are to be seen together. For example, signal strength should decrease with distance, so the number of training data points in a high-signal/large-distance should be fairly small. The probability is then calculated as the number of training data points in the particle's bin divided by the maximum number of points any bin has:

$$L_{p,i} = \frac{b_{p,i}}{b_{\max}}$$

where $b_{p,i}$ is the number of points in the bin of the particle in question, and b_{max} is the maximum number of particles contained in a single bin. Therefore, if we observe a common event, say a weak signal that comes from a significant distance and through several walls, it will correspond to a more populated bin, and therefore will result in a higher probability.

Each of the 5 dimensions is discretized by putting the values into ranges; for instance, distances of 0-7 meters fall into the first range, 8-14 fall into the second, 15-21 fall into the third, 22-28 fall into the fourth, 29-35 fall into the fifth, and anything higher than 35 falls into the sixth range. Each other dimension is likewise broken up into ranges (5 ranges for floor difference, 6 for the number of walls encountered, 2 for the number of atrium pixels encountered, and 9 for the signal strength ranges), and so the entire space becomes discrete. The actual selection of these ranges was done by hand, though one could imagine using more sophisticated techniques to optimize performance.

To further optimize the process we had the likelihood set to 0 for any particle whose position was inside of a wall or the open area of the atrium (except on the 1st floor); estimates generated in these regions are bound to be incorrect, and so it seemed prudent to set them at a disadvantage. This did not quite work out as we had intended; this is discussed in more detail in the analysis section.

The reasoning behind this method is that the training data should show us the statistical correlations between the number of walls encountered, amount of the atrium covered, floor difference, distance and signal strength. Rare combinations of these should have very few, if any, points in their corresponding bins, and thus generate low probabilities. The hope was that in addition to characterizing the signal's obvious interactions with walls, distance and such, it would pick up on some of the more subtle statistical interactions, the kind of thing that wouldn't happen if the probability were just calculated as a weighted sum of these values. It is also a fairly universal technique, which could easily be used in other environments, and which could easily be modified to include additional dimensions of information, were the system expanded in the future.

Map based motion model

The motion model of the particle filter governs how the particles move over time, both in terms of 2d movement and in terms of floor transitions. Although initially we had several ideas for how to use the map here, such as to reduce the chance of a particle transitioning through a wall, the only one that is currently in use is checking for whether the particle is on or around a stairwell before changing floors. While the other possible modifications to the motion model could possibly have improved accuracy, there was also the possibility that they would hurt it; reducing the chance of a particle moving through walls would prevent good estimates from going astray, but they may also prevent bad estimates from getting better, and result in particles getting locked in rooms and other such areas. Ultimately, there wasn't enough time to try as many variations as we would have liked, and so we settled on the best sounding approaches.

Results for the Map based particle filter approach

This approach did not live up its our expectations, as the mean error on the 2d estimation hovered around 20 meters, with a maximum error of 43 meters, and with the floor being estimated correctly only 34.8% of the time. This is an improvement over the accuracy of Place Lab's standard particle filter, whose error is around 30 meters, but it nonetheless falls short of our expectations. The results are shown below.

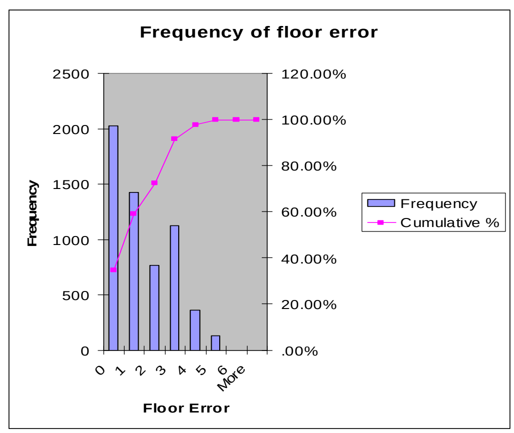

Figure 13. Floor estimation error for map based particle filter.

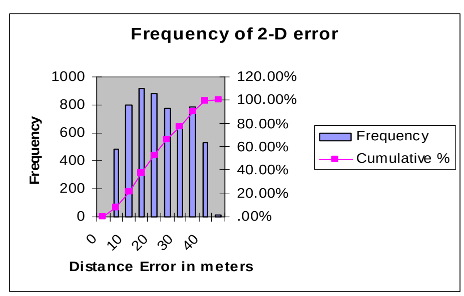

Figure 12. 2-D estimation error for map based particle filter.

particle2dError	
Mean	19.77317
Standard Error	0.139246
Median	19.58416
Standard Deviation	10.63567

Table 2. 2-D estimation error of the map based particle filter

Sources of error with the Map based particle filter

Taking advantage of the surrounding environment seemed to be a very promising idea, especially the ideas of taking into account the number of walls penetrated by the signal, and in which areas it is safe for a particle to transition. Yet the results were not as impressive as we had initially hoped, and so it becomes necessary to consider the limitations of the method, and assumptions of ours that were perhaps not justified. The first limitation that comes to mind is the limitation inherent in modeling radio waves, but this will be addressed later; to begin with we will consider some less obvious, but perhaps just as important, issues.

One such problem is that of symmetry, which is quite abundant in the Allen Center. Consider the situation shown in the image shown below. Walls are shown in blue, the access point in teal, and the user to be localized in orange. Even if the WiFi signal could be modeled perfectly, there is no way of knowing whether the correct estimate is the actual location or a similar one further to the right in the image; the area is symmetric in that sense, and so the number of walls passed through and the distances involved will be the same. In reality there will be other access points around to shed some light on the situation, but given rather chaotic WiFi signals and large distances involved, it could still be difficult to interpret this information, even for a well modeled radio wave system.

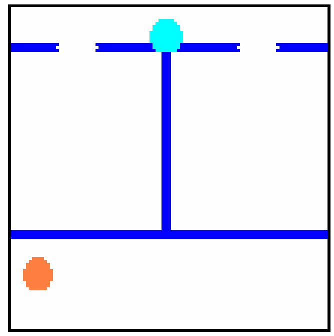

Figure 14. Image illustrating potential symmetry problem for map based particle filter

There are potential issues with the maps themselves; the maps we used were accurate in terms of the layout, but wall lines are not necessarily drawn with their correct relative thicknesses, and certainly don't take into account material composition. The assumption that all walls should be treated equally (because we can't differentiate the thicknesses and materials anyway) is no doubt a source of error, as radio wave propagation is certainly tied to the medium it passes through, not just in terms of weakened signals, but also in terms of reflections. The maps used also do not take into account other objects, such as furniture, computers and people, all of which will in reality impact the spread of the waves involved. And of course, the maps do not represent other factors that may affect the wave propagation, factors such as temperature and pressure no doubt contribute to the effect, perhaps in some small way, but perhaps in ways not so small.

There is the question of whether WiFi signals themselves are predictable enough to use for localization. That is, even if our maps were updated to take into account all of the factors of the environment, ranging from wall composition to reflective properties, to objects and people in the way, could we then produce amazingly accurate estimates? Elnahrawy et al. argue that not only are all current approaches to the problem of using WiFi signals for localization limited in accuracy, but that using WiFi signals for localization is a fundamentally limited approach, and that much more sophisticated hardware, or modeling techniques, may be required to see any improvement beyond a certain point.

Nonetheless, certain changes could no doubt be made to bring our results closer to the ground truth. Frequently, however, such changes may do more harm than good. In the sensor model for the map based particle filter tracker we set to 0 the likelihood of any particle that is inside of a wall or in the atrium (except on the ground floor); these estimates are certainly incorrect, and so it makes sense to remove them. However, the results are actually worse when this change is in effect, which we found both surprising and disappointing. One explanation I can see for this is that it may prevent particles from moving as freely as they should: particles won't be entirely trapped by walls, but their propagation will be lessened. In addition, they will also be unable to cross the atrium. This makes sense in that a person wouldn't be able to, but the particles themselves are estimates that need a large degree of freedom in order to settle into the more likely areas, and restricting their movement in this way was apparently too restrictive.

DISCUSSION AND CONCLUSION

We have mixed feelings about our results. On the one hand, they can broadly be characterized as poor. However, in trying out different schemes we have been able to isolate important factors that may influence a designer's decision on what scheme to use. Our negative results also indicate opportunities for improvement. We feel that every scheme has its downsides and therefore the initial results from our alternative schemes should not discourage future work in finding low-cost and low-effort localization schemes.

FUTURE WORK

There are numerous modifications to the schemes that we could try. We did not investigate the effects of smoothing on The most promising would to be use additional sensors. These sensors have the potential to greatly impact the usefulness of a particle filter in combination with a map-based approach. Changes in pressure could indicate a change of floor while exact pressure could indicate exact floor. An accelerometer could indicate actual movement so that particles spread out only when the client is actually moving. Having a microphone capable of ultrasound detection could lead to highly precise room-level positioning so even if only some rooms are instrumented with ultrasound beacons, a scheme could overcome the accumulation of error.

REFERENCES

[Bat] Andy Ward, Alan Jones, Andy Hopper. *A New Location Technique for the Active Office*. IEEE Personal Communications, Vol. 4, No. 5, October 1997, pp. 42-47.

[CB] Anind K. Dey and Gregory D. Abowd, CybreMinder: A Context-Aware System for Supporting Reminders, Symposium on Handheld and Ubiquitous Computing, Bristol, UK, 2000.

[Gru] Marco Gruteser and Dirk Grunwald. *Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking*. ACM/USENIX International Conference on Mobile Systems, Applications, and Services (MobiSys) 2003.

[HEM] Andreas Haeberlen, Eliot Flannery, Andrew M. Ladd, Algis Rudys, Dan S. Wallach, and Lydia E. Kavraki, *Practical Robust Localization over Large-Scale 802.11 Wireless Networks*,
MOBICOM 2004, Philadelphia, PA, September 2004.

[Inf] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz. *Voronoi tracking: Location estimation using sparse and noisy sensor data*, Ubicomp 2003, Seattle, WA, October 2003.

[Lim] Eiman Elnahrawy, Xiaoyan Li, and Richard P. Martin, *The Limits of Localization Using Signal Strength: A Comparative Study*, In Proceedings of The First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON 2004), Santa Clara, CA, October 2004.

[LL] Julie Letchner and Benson Limketkai. *Localization Using WiFi Strength*. CSE561 Course project, December 2003.
http://www.cs.washington.edu/education/courses/cse561/03au/projects/letchner_limketkai.pdf

[OpKn] Donald J. Patterson.Lin Liao. Krzysztof Gajos, Michael Collier, Nik Livic, Katherine Olson, Shiaokai Wang, Dieter Fox, and Henry Kautz, *Opportunity Knocks: a System to Provide Cognitive Assistance with Transportation Services*, Ubicomp 2004, Nottingham, UK, October 2004.

[PL] Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Tim Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello and Bill Schilit. *Place Lab: Device Positioning Using Radio Beacons in the Wild*. IRS-TR-04-016.
http://www.placelab.org

[Rad] P. Bahl, V. Padmanabhan. *RADAR: an in-building RF-based user location and tracking system*. INFOCOM 2000, March 2000, pp: 775 – 784.

[Rob] Dieter Fox, Sebastian Thrun, Wolfram

Burgard, and Frank Dellaert. *Particle filters for mobile robot localization*. Sequential Monte Carlo Methods in Practice, New York, 2001.

[Spot] Jeffrey Hightower, Roy Want, and Gaetano Borriello, *SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength*, UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, Feb. 2000.

[WA] Tony Offer and Chris Palistrant. *Wireless Active Location Resolver with Ultrasound*. CSE 477 Course project.
http://www.cs.washington.edu/education/courses/cse477/04sp/projectwebs/cse477m/