Location Determination of a Mobile Device Using IEEE 802.11b Access Point Signals

Siddhartha Saha, Kamalika Chaudhuri, Dheeraj Sanghi, Pravin Bhagwat
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur, INDIA
{sidd,dheeraj,pravin}@cse.iitk.ac.in

***Abstract*— Wireless LANs are becoming increasingly popular today, particularly those based on IEEE 802.11b standard. We study the problem of determining the location of a mobile device, which is communicating through a WLAN. We exploit the fact that the strength of the signals that a device will receive from different access points will vary with location. We build a database of signal strength information for various locations, and use this information to determine which location a given test data comes from. The problem is complicated because RF Signals are affected by the noise, interference, multi-path effect and random movement in the environment. We find that in spite of this randomness, the signal information is sufficient to detect the position of a mobile device with certain error margin.**

I. Introduction

With the increase in mobile computing devices and wireless LANs, it has become important to determine the location of a device at any point in time. Several applications can be conceived of that can use this information [1]. Detecting the location is one of the first step towards building context sensitive smart devices [2]. For example, museums can use such a technology to build location-aware hand-held devices. As a visitor walks into a room, holding such a device, information about the exhibits in that particular room flashes on the screen. Indoor location aware systems may also be used to do guide users as well as robots in large and complex buildings, depending on the available granularity of the information.

The most popular WLAN today is based on the IEEE 802.11b standard [3]. In this paper, we present methods to determine location in such a WLAN. We adopt an experimental approach to the problem. We choose a building which has an 802.11b WLAN deployed. We record the strengths of the signals received by a client from multiple access points at different locations in the building. This information is then used to build a system which, given a sample of signal strength information, will determine the location at which the sample was recorded.

An advantage of this approach is that we need no extra equipment; the system uses components which are expected to be part of WLAN. This helps in making the system more cost effective than a system which uses extra sophisticated hardware devices.

II. Background and Related Work

Detection of the location of a mobile device has been addressed by several researchers in the past few years. Hightower *et. al.* [4], [5] give a summary of the advancement in location tracking systems.

The Cricket location-support system [6] uses a combination of RF and ultrasound. It uses devices called *beacons* and *listeners*. *Beacon* is a small device placed at a fixed location, usually ceiling-mounted, and publishes the location information on an RF signal along with an ultrasonic pulse. The *listeners*, usually attached to the mobile devices, uses the difference in *Time of Flight* of RF signal and the ultrasonic pulse from the same *beacon* to infer the location.

Ward *et. al.* [7] describe a location detection system developed at ORL. In this system, the mobile device is equipped with ultrasonic transmitter, transmitting at a fixed interval. The *time of flight* of these sound pulses from the transmitter to receivers placed at known positions is measured. From the pulse transit times, the distance from the receiver to transmitter can be obtained, from which, the transmitter's location can be found by multilateration [4].

Unlike these approaches, which uses specialized hardware to do location detection, we used no extra hardware.

There are essentially two ways in which a wireless LAN technology can be used to do location detection.

1) By measuring the time between transmitting a packet at the access Point and receiving it at the client.
2) By exploiting the fact that at different locations the signal strength received from the various access points is different.

The first alternative would require highly synchronized clocks at the sender and receiver, since RF signal traverses 3 meters in just 10 nanoseconds; hence it is impractical from an implementation point of view. Approaches that make use of the *Time of Flight* technique use ultrasonic transmitters and receivers, making the clock synchronization feasible due the much slower speed of sound than the speed of RF signal. It is the second approach that we have adopted in our system. There are again two ways in which we could have used the signal strength information. One way would be to use a signal propagation model and our information about the geometry of the building to compute the signal strength. Such an approach however is difficult to take; also it is rather unlikely to work in practice, except in some ideal or highly simplistic cases.

The second way, which we have adopted, is to build a database of signal strength data for a various locations and then use this training data to perform classification of a test data into one of these locations.

Similar work has been done previously by Bahl and Padmanabhan [8], who have used IEEE 802.11b access point signals to detect the location of a user. They have presented two approaches to solve the problem. The first one is an empirical method, in which they have built a database of the signal strength of three access points at various points inside a building, and used this data to determine the location of a mobile user. In the second approach, they tried to model the radio signal propagation inside the building by taking into consideration the walls and objects in the building. But it was found that the results were better in the case of empirical model, where they used a nearest neighbor classifier. This can be primarily attributed to the complexity involved in accurately modeling radio propagation inside a building.

We have also adopted the empirical approach. However, we employ more sophisticated classifiers which take into account the distribution of the data.

Ladd *et. al.* [9] have also addressed the problem using the RF signal strength. But in their approach they have used signals from nine access points. We have done our experiments using only 3 access points and obtained good results in terms of accuracy of prediction but not as accurate as in [9]. But in order to do a meaningful comparison, we did another set of experiments, using eight access points in the building. This time we obtained results better than the results reported in [9].

III. Method

Our solution method is to build a database of signal strengths at several locations in a building; we then use this data to classify a test sample into one of these locations.

This method, however, has a number of challenges. The signal strength as received by a receiver at a spatial point is not constant, but varies with time. The variation in the signal strength is due to several factors - change in temperature, movement of people, and other disturbances. The signal strength also varies with change in the orientation of the receiver's antenna. In addition, the radio used in the normal 802.11b LAN Cards have some inaccuracies in their measurements of the signal strength. All these factors add up to a variation in the signal strengths at a place which is of the order of 5 to 7 dB. Hence we can detect locations using this data, only if such variation is much smaller than the variation in signal strengths due to change in location.

If we look at the signal strength distribution of two locations which are some distance apart (say, 20m), we do find that the signal strength distributions for all the channels are quite different and have little overlap. This suggests that the approach we are using has some promise.

A. Setup

We conducted our experiments at the ground floor of the Computer Center Building in IIT Kanpur. Three Access Points were installed in the building and these were configured to use channel numbers 1, 6, and 11 respectively, since these three channels are non overlapping.

Our data collection system comprised of a laptop, running Windows 2000. The laptop was equipped with a Lucent Orinoco Wireless LAN Card. For the purpose of our data collection, we used a sniffer utility. The sniffer is capable of capturing all packets in any channel in promiscuous mode. The sniffer works by switching to a set of given channels at user defined frequencies, and capturing all signals that it gets on that particular channel. We configured the sniffer to cycle through the channels 1, 6, and 11. The cycling frequency was set to 3 seconds, which means the sniffer would switch channels once a second. The Access Points were configured to send a beacon packet every 100 milliseconds. We used the signal strength of these beacon packets as our RSSI (Received Signal Strength Information). Hence, in one second, the sniffer would capture 10 such beacon packets from one access point.

With this set up, we collected signal strength information at several points in the Computer Center. At each point, we collected the data for four orientations - North, South, East and West, to study the effect of receiver orientation on the data. Data was collected for about 5 minutes at a particular orientation per spatial point. We collected data for 19 different training points.

We collected data for each point at different times of the day, on two days at an interval of a few weeks. Data was also collected at neighborhood points of these sample points.

B. Effect of Orientation

Observation of the signal strength information collected at different orientations shows that there is a slight variation of the data with changing orientation. This is an expected behavior as the antennas of the wireless cards are not perfectly isotropic. But, in most cases, the variation due to different orientation is not significant, the average signal strength remains within 3 to 4 dB. Since overall variance in the signal strength is quite less across orientations than across different locations, for the purpose of classification, we do not take into account the effect of orientation.

C. Preprocessing

After the data is obtained, it is preprocessed to remove the CRC error packets. We also remove the outliers from the training data. A data sample is said to be an outlier, if it lies two standard deviations beyond the mean of a given channel at a given location. It was found that very few (about 0.1%) of the sample points are outliers.

D. Classification

Determining a location is essentially a classification problem. The training set is the signal strength information at various points in the Computer Center. The test set is a sample of signal strength information at some location within the Computer Center. The problem is to classify the test sample into one of the predefined classes, based on the information obtained from the training data.

We have studied three methods for classification. The first is a nearest neighbor classifier; the second is a back-propagation neural network. The third classifier uses histogram matching. A brief overview of these classifiers is presented below.

1) Nearest Neighbor Classifier: Consider a sample of the signal strength information for a given time interval. The sample will contain information corresponding to the three channels or three access points. The 3-tuple (c_1, c_2, c_3), denoting the average of the first, second, and third channels, respectively, can be thought of as a point in 3-D space.

In this classifier, using the training data, we first calculate the mean of signal strength for each channel $M_j = (c_{j1}, c_{j2}, c_{j3})$ at each location. This constitutes the signal strength profile of a given location. Now given a test sample, we calculate the mean of the three channels $M' = (c'_1, c'_2, c'_3)$. Then our classifier would determine that the test data belongs to location j if

$$dist(M_j, M') <= dist(M_k, M'), \forall k \neq j \tag{1}$$

In other words, our classifier chooses that class whose center is nearest in feature space to the mean of the current data, where nearest is defined in terms of some distance measure. Here, we use is the Euclidean distance.

Modification to the nearest neighbor classifier

One problem with the nearest neighbor classifier is that it blindly chooses the class whose center is nearest in feature space to the mean of the current data, without any consideration of the distribution of the data. Sometimes we have test data nowhere close to the training data. This classifier will still give a location as the result of the classification.

To solve this problem, we have modified the classifier slightly such that it reports such data points as unclassifiable. We consider the profile for a given location as consisting of the values, $M_j = (c_{j1}, c_{j2}, c_{j3}), and S_j = (s_{j1}, s_{j2}, s_{j3})$. M corresponds to the mean of the data samples for the three channels respectively, whereas S corresponds to the standard deviation values for these three channels. Given a test sample, we calculate the mean of the three channels, $M' = (c'_1, c'_2, c'_3)$. Then our classifier would determine that the test data belongs to location j if M' lies within the rectangular parallelopiped with M_j as center and $4s_{j1}$, $4s_{j2}$, and $4s_{j3}$ as the length of the sides, respectively. More mathematically, the condition reduces to

$$c_{j1} - 2s_{j1} <= c'_1 <= c_{j1} + 2s_{j1} \tag{2}$$

$$c_{j2} - 2s_{j2} <= c'_2 <= c_{j2} + 2s_{j2} \tag{3}$$

$$c_{j3} - 2s_{j3} <= c'_3 <= c_{j3} + 2s_{j3} \tag{4}$$

If a sample point M' lies within no such region, we say that we are unable to classify it. If a sample point lies within more than one such region, we break the tie by choosing the region whose center is nearest to M'.

2) Neural Network: Our second classifier is a Back-propagation neural network [10].

A neural network is a collection of neurons arranged in a particular manner to form a network, where the output of some neurons feed into the input of some others and so on. A Back-propagation neural network is a multi-layer feed forward neural network. Such a network typically consists of several layers - a set of source nodes comprising the input layer, one or more hidden layers and an output layer of neurons. The term "feed forward" indicates that input connections go only in one direction, that is, from input to hidden, hidden to output etc, and not the other way round.

Training of such a network is done by adjusting the synaptic weights such that given a particular input, the network provides a particular output. For a back-propagation network, training is done by using an iterative algorithm called the error back-propagation algorithm.

The Back-propagation neural network we used had 3 inputs, one for the mean of the signal strengths of each channel. It had a hidden layer with 20 nodes and an output layer consisting of N nodes, where N is the number of our training points, one for each location in our training set. In this case, we had $N = 19$. The output of the network is thus a n-dimensional vector, which gives us the probability of the input data belonging to each class.

We had trained the neural network by signal strength samples from each of these N training location, after preprocessing the data to remove the outliers. Each vector used to train the neural network is the average signal strengths obtained from each access point.

For testing process, we take the average of signal strengths obtained from different access points to construct the input vector, which is then fed into the neural network to obtain the output.

3) Classification By Histogram Matching: In this classifier, we match the distributions of the signal strength information directly and come up with a classification result. We first construct the histograms of the signal strength for each location in the training data. Given a test sample, we construct the histogram of the test data and compare the two histograms directly, by finding out the overlap between them. We assign the test sample to the class with which it has the highest overlap in their histograms.

Mathematically, let $F_1(t_i)$, $F_6(t_i)$, $F_{11}(t_i)$ are the normalized histograms of the training data at location t_i for channels 1, 6, and 11 respectively. And let $G_1(s)$, $G_6(s)$, $G_{11}(s)$ be the normalized histograms of the sample test data s for channels 1, 6, and 11. Let A_{1i} be the common area under $F_1(t_i)$ and $G_1(s)$. Similarly define A_{6i} and A_{11i}. Then for the test sample s, we assign it the cluster k for which the quantity $A_{1i} * A_{6i} * A_{11i}$ is maximum for $i = k$, and i varies over all the clusters.

We have noticed that before applying this test, a preprocessing of the histogram by a moving window average slightly improves the result.

IV. RESULTS

A. Experiments

To assess the performance of the classifiers, we conducted a number of experiments. We divided the the ground floor of the Computer Center building into 19 clusters according to the 19 training points from which we had collected the training data. These training points were scattered more or less evenly throughout the ground floor, which consisted of two narrow corridors, one wide corridor, and 6 big rooms used as computer labs.

We conducted two different sets of experiments. In the first one, we chose a cluster randomly and measured the signal strength at the same position from which the training data came for that cluster. In the second set of experiments, we randomly picked up a location which usually is located within 3 meters of a training point, and collected the signal strength information there. Based on this signal strength information, our system determined in which cluster this point belongs to. We repeated both the experiments several times. For each experiment, we collected data at the test locations for about three minutes. We wanted to study the time duration for which we need to collect the signal strength data to do the location detection reliably.

B. Error Measures

We introduce here our performance measures for the system. To measure *precision*, we use *Error Distance*. This is defined as the spatial distance between the original point to which the data belongs, and the point which is reported by the classifier. The *Average Error Distance* is defined as the sum of all the *Error Distances* averaged over all the runs of the experiment. We ignore the experiments which produced a *Unable to Classify* result.

Mathematically, let N_T be the number of trials in an experiment, and N_C be the number of trials for which the classifier yields an "Unable to Classify" value. Then the *Average Error Distance* is defined as

$$A_e = \frac{\Sigma_{i=1}^{N_T - N_C} dist(l_i, y_i)}{N_T - N_C} \quad (5)$$

Here l_i is the original location of the test data used for the i^{th} trial and y_i is the value of the location obtained from the classifier for the i^{th} trial, and $dist(x_1, x_2)$ denotes the distance between the points x_1 and x_2.

We also measure *accuracy* of our location determination. *Accuracy* is defined as the percentage of times our classifier determines the correct cluster for a test data sample over all the runs of the experiment.

1) The Ideal Classifier: Since we are doing cluster based location detection, our system will return one of the 19 training locations. Hence for the cases where we are picking our test location a few meters away from the training point, our classifier will definitely incur some error value. To measure the severity of these error values, we introduce a concept of an *ideal classifier*. This classifier always determines the correct cluster for any given test data. Hence

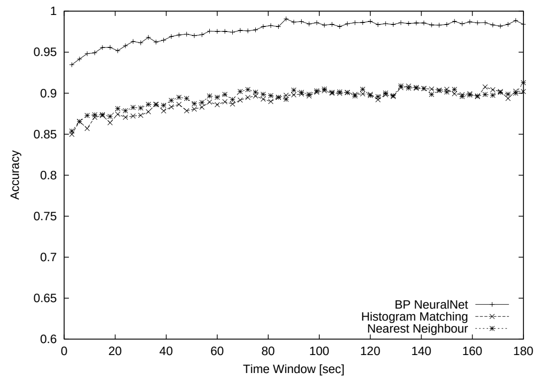

Fig. 1. The Accuracy obtained by various classifiers when the test location is one of the training locations.

the *accuracy* of this classifier is always 100%. In the first set of experiments, since we are collecting the test data from the point from which we had collected the training data, the *Average Error Distance* of the ideal classifier will be zero. But for the second set of experiments, the *Average Error Distance* of this ideal classifier will be non zero.

C. Results

Below, we summarize the results of the experiments alongwith the observed accuracy and the precision of the classifier.

1) Accuracy: Figure 1 shows the accuracy observed with different classifiers when test location was one of the training location. The accuracy improves slightly as the amount of time we collect signal strength information at the test location. The Nearest Neighbor and the Histogram based classifier perform equally well, both registering an accuracy of around 90%. But the performance of both these classifiers are surpassed by the Neural Network based classifier, which recorded an accuracy of close to 97% to 99%. This behavior is expected, because the relative positioning of the signal strength from three access points with respect to each other is captured only by the Neural Network based classifier.

Figure 2 shows the accuracy recorded by different classifiers when the test location is different from training locations. Here also the performance of the Nearest Neighbor and the Histogram based classifiers are almost equal at around 85%. The Neural Network based classifier records a successful classification rate of around 90%. This accuracy is obtained when the minimum distance of two points from which we collected data was 3.12 meters.

Another interesting point to be noted in the results is that the accuracy does not change significantly with the increase in sampling time.

2) Precision: Figure 3 shows the Average Error Distance as a function of sampling time when the test location is one of the training locations. There is a gradual decrease in the Average Error Distance for all the classifiers up to a sample size of one minute, after which it stabilizes. Here also the Neural Network based classifier records Average

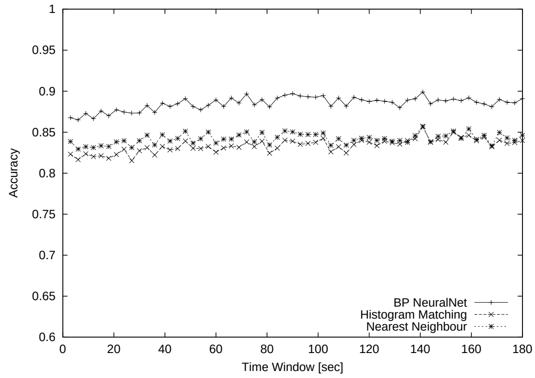

Fig. 2. The Accuracy obtained by various classifiers when the test location is not one of the training locations.

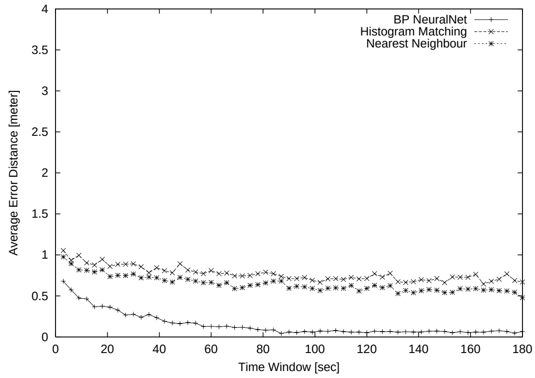

Fig. 3. The Average Error Distance obtained by various classifiers when the test location is one of the training locations.

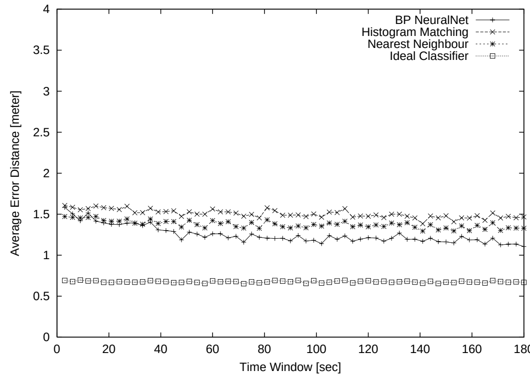

Fig. 4. The Average Error Distance obtained by various classifiers when the test location is not one of the training locations.

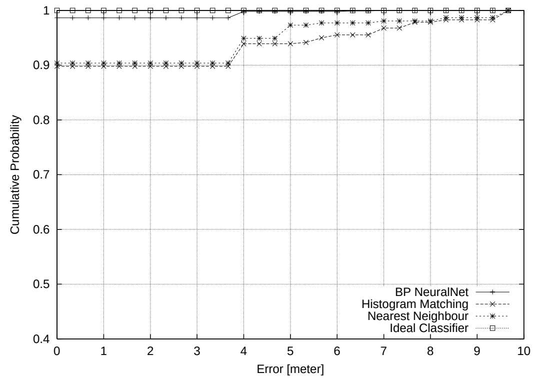

Fig. 5. Cumulative Error Distance Distribution when the test location is one of the training locations.

Error Distance near zero and outperforms the other two classifiers. Figure 4 shows the results obtained when the test location is different from training locations. Here we also plot the Average Error Distance of the Ideal classifier that we described earlier. The Neural Network based classifier performs slightly better than the other two classifiers. Also, the average error distance does not change much with the sampling time.

Figures 5 and 6 shows the cumulative Error Distance Distribution of the classifiers in the two sets of experiments. When the test location is one of the training locations, the Nearest Neighbor was accurate within one meter with 95% probability. The Ideal Classifier is, of course, 100% precise. When the test location is not one of the training locations, the Neural Network Based classifier recorded an Error Distance of less than 1.5 meters with 83% probability, whereas the Ideal Classifier gave an Error Distance of 1.5 meters with 92% probability. The other two classifiers gave an Error Distance of just over 2 meters with a probability of 83% to 84%.

D. Experiments with several access points

Recently, Ladd *et. al.* [9] have addressed the same problem and have reported good results. But they have used nine access points in a building which is only slightly larger than our computer center building. Using nine access points in a building of that size is not desirable. There are 11 channels in which an access point can operate, and amongst them, only the three channels, namely 1, 6, 11 operate on a mutually exclusive frequency range. All other channels have frequency overlap with neighboring channels [3]. As a result, using 9 access point will not only increase the cost of hardware infrastructure, but will also have adverse affect to the wireless network performance due to collisions between signals from overlapping channels. Signals from a large number of access points will definitely give more information to use in location tracking, but that information may also bring along undesirable performance issues and high cost. Still, in order to make a useful comparison, we conducted another experiment. We conducted this experiment on the first floor of the Computer Science department.

We have used eight access points. We exhaustively covered the corridor with our training points, a total of 24 of them, separated by 2-3 meters. Test locations were selected randomly throughput the corridor. We performed the same set of tests as we had done in our earlier experiments. Below we present the results.

The effect of having eight access points was immediately apparent when all the three classifiers reported 100% accuracy and zero average error in the first set of experiment, where the test location is also one of the training locations.

In the second set of experiments, when the test location

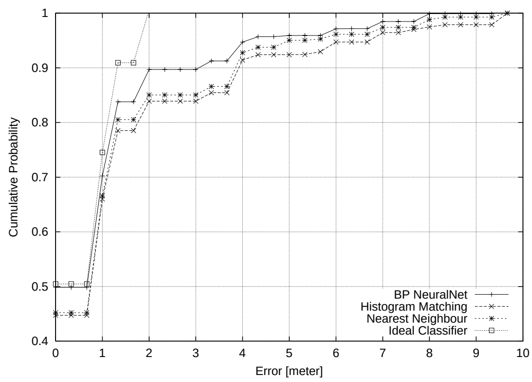

Fig. 6. Cumulative Error Distance Distribution when the test location is not one of the training locations.

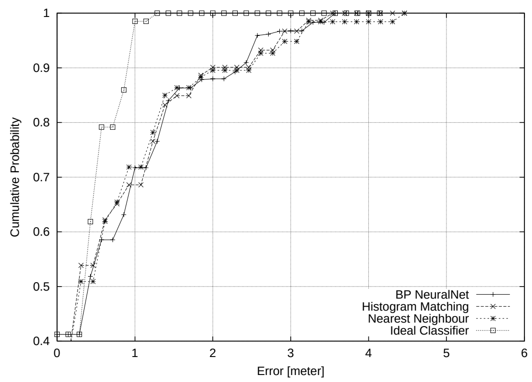

Fig. 8. Cumulative Error Distance Distribution with eight access points, when the test location is not one of the training locations.

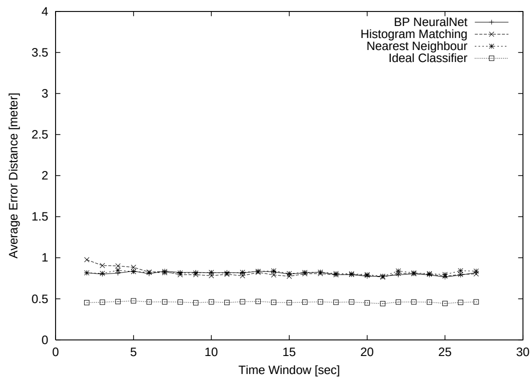

Fig. 7. The Average Error Distance with eight access points, when the test location is not one of the training locations.

is selectedly randomly in the corridor, the three classifiers behaved almost the same way, with the neural network based classifier performing slightly better than the other two. Figure 7 shows the average error obtained by different classifiers in this case. Figure 8 shows the most important result of this experiment. It shows the cumulative error probability distribution of the three classifiers. The neural network based classier gives an error of less than one meter with 72% probability, less than 2.6 meters with over 95% probability, and with almost 98% probability, it could predict the location within 3.3 meters. This is a significant improvement over the result shown in Figure 6, where, the classifier could reach a probability of 98% with an error margin of 7 meters. This is also better than the result obtained by [9].

V. Conclusions

From the results, it is evident that position can be determined in an indoor environment using IEEE 802.11b access point signal strengths. The accuracy of location determination depends on how close the two points to be distinguished are and signals from how many access points are being used. Using more number of access points surely makes it easy for a location detection system, simply because the amount of information that can be obtained increases with the addition of each extra access point, and the probability of two distant points having the same signal strength profile decreases. But it introduces problems related to network performance and hardware cost. There has to a trade-off between performance, cost and accuracy of the location detection system. The system we developed works quite well with a minimal number of access points and excels in performance if given a infrastructure with large number of access points.

Another conclusion from our experiments is that accuracy and precision do not change significantly with the increase or decrease in the sampling time. This suggests that we need to take the sample of signal strengths at a given location for only a small amount of time to achieve reasonable accuracy. This makes our method of position detection quite practical. This also means that this method can be applied to indoor location tracking - that is, determining the position of a moving user in a building, provided the user is not moving too fast.

References

[1] R. Want and B. Schilit, “Expanding the horizons of location-aware computing,” *IEEE Computer*, vol. 34, no. 8, Aug. 2001.

[2] H. Gellersen, A. Schmidt, and M. Beigl, “Adding some smartness to devices and everyday things,” in *Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 2000.

[3] Bob O’Hara and Al Petrik, *The IEEE 802.11 Handbook, A Designer’s Companion*, IEEE Press, 1999.

[4] Jeffrey Hightower and Gaetano Boriello, “Location sensing techniques,” Tech. Rep. UW-CSE-01-07-01, University of Washington, Computer Science and Engg. Department, 2001.

[5] Jeffrey Hightower and Gaetano Borriello, “Location systems for ubiquitous computing,” *IEEE Computer*, vol. 34, no. 8, pp. 57–66, Aug. 2001.

[6] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The cricket location-support system,” in *Proc. of 6th ACM MOBICOM*, Boston, MA, Aug. 2000, pp. 32–43.

[7] A. Ward, A. Jones, and A. Hopper, “A new location technique for the active office,” *IEEE Personal Commun.*, vol. 4, no. 5, pp. 42–47, Oct. 1997.

[8] Paramvir Bahl and Venkata N Padmanabhan, “Radar: An in-building rf based user location and tracking system,” in *Proc. of IEEE INFOCOM 2000*, Mar. 2000, pp. 775–784.

[9] Andrew M. Ladd, Kostas Bekris, Algis Rudys, Guillaume Marceau, Lydia E. Kavraki, and Dan S. Wallach, “Robotics-based location sensing using wireless ethernet,” in *Proc. of 8th ACM MOBICOM*, Sept. 2002.

[10] Simon Haykin, *Neural Networks: A Comprehensive Foundation*, Pearson Education Inc, 1999.