2015

Fault tolerant indoor localization using Wi-Fi

Anusha Chennaka
Iowa State University

Fault tolerant indoor localization using Wi-Fi

by

Anusha Chennaka

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering

Program of Study Committee:
Greg R. Luecke, Major Professor
Manimaran Govindarasu
Chris Harding

Iowa State University

Ames, Iowa

2015

DEDICATION

I would like to dedicate this thesis to my parents Raja Reddy Chennaka and Krishna Kumari Ambarapu for their loving guidance.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

AP	Access Point
IR	Infra-Red
ROC	Region of Consideration
RSS	Received Signal Strength
AIC	Akaike's Information Criterion
RF	Radio Frequency

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Greg Luecke, and my committee members, Dr. Manimaran Govindarasu and Dr. Chris Harding for their guidance and support throughout the course of this research.

In addition, I would also like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience.

Finally, thanks to my family for their encouragement and to my friends for their hours of patience, respect and love.

ABSTRACT

Precise Indoor Localization is a major component of numerous location based applications and services which perform indoor guidance and object tracking. There are many existing solutions which address the localization issue, but most of them do not provide a fault tolerant solution. In this work, we have developed a fault tolerant statistical method which leverages the existing infrastructure by using the readily available Wi-Fi Access Points. Our proposed method can be applied to any environment which has a Wi-Fi coverage and we do not assume the knowledge of the placement of the Access Points or any physical layout. Initially we map the signal strengths and the corresponding positions to obtain the RF distribution of the region and this is the offline phase. We develop different fault tolerant models and use an Android application for monitoring various Access Points to provide the status of the Access Points in the environment. During the online phase, we measure the signal strength at distributed locations in the environment and then, depending on the status obtained from the application, we use the appropriate scheme to obtain the corresponding locations. In specific we use a Maximum Likelihood Estimator to obtain the position from the previously recorded RF map. Further, we provide 95% confidence intervals for the location obtained by using a Bootstrap method. Our method, compared to other deterministic methods is more accurate and fault tolerant. We also provide the experimental results which validate the accuracy of our method in obtaining the user location.

CHAPTER 1. INTRODUCTION

Indoor Localization has gained popularity as a critical enabler for location based applications in the recent times. Location based information is necessary in many environments such as airport terminals and malls to provide the user navigation and promotion information. It also helps improve the business by studying the number of users visiting a particular store and study the rate of flow of passengers to obtain statistical information of the number of passengers flying through a particular city during a period. Moreover Indoor localization systems can also be used in automatic object location systems to detect objects in a warehouse or personnel in a huge location. The wide range of smart devices which have access to Wi-Fi and 4G LTE have triggered the development of these systems. For example, if you go to a mall, your smart phone could detect the location and then could automatically suggest you the restaurants nearby and also could fetch coupons for the stores. This seemingly easy mechanism involves location detection, indoor localization and also interaction with the database of the store to fetch valid coupons. This example portrays the gravity of Indoor Localization which is the problem we have addressed in this work.

Indoor Localization cannot rely on GPS since GPS doesn't work indoors. Hence an alternative means has to be pursued to achieve localization indoors. Various techniques using RFID, Ultrasound, IR and Bluetooth have been proposed. However all these techniques require additional infrastructure which poses a significant overhead. The recent commercial offerings such as Google Maps and ShopKick which help a user to position himself indoors have errors more than 5 meters or are only functional at the location of the stores. SkyHook offers an SDK which could be used to develop location based applications using the information of various

Wi-Fi APs it collects using wardriving. But, as their website mentions, the location obtained using their SDK is accurate only up to 10 meters.

In this work, we have addressed the problem of Indoor Localization by making use of the beacons emitted by the Wi-Fi Access Points (APs) and measuring their signal strength. A regression module was used to fit the records of Received Signal Strength (RSS) to the location to obtain signal strength distribution in the offline stage. The online stage consists of measuring the signal strength(s) at a particular location and using the previous signal strength distributions to obtain the position. Also, we have calculated the confidence intervals of the location estimated. And a fault tolerant module has been incorporated which caters to the various faults occurring in the Wi-Fi environment and thus can deal with the dynamic changes. By using our proposed method, the mean localization error was restricted to 2 meters.

1.1 Related work

There have been a plethora of papers in the past which have discussed different strategies in order to achieve indoor positioning effectively. The most popular work is RADAR[1] based on Wi-Fi technology. The authors have used two stages which they called off-line phase and real-time phase. In the offline-phase the average signal strength from the mobile clients are measured to build radio map recording locations, signal strengths and users' directions. During the next stage the readings were analyzed using the layout of the building plan and a line clipping algorithm. The nearest neighbor in signal space search methodology was used to compare multiple locations and pick the best one that matches the observed signal strength. The accuracy using the first method was found to be 2.94m. However this method doesn't include any mechanism which takes into account the dynamic changes in the environment.

Another popular work is Horus[4] which chooses the estimated location from the discrete set of radio map locations. Similar to RADAR, Horus also uses two phases namely online and offline. The offline phase employs a probabilistic method, the Bayes' theorem. It handles the correlation between different samples from the same access point and uses a perturbation technique to deal with small variations in the signal strength. Horus also uses a time average window to obtain continuous location estimation. The authors were able to achieve an accuracy of 2m. However, this technique is complicated as it has a large number of modules.

LANDMARC[6] is another Indoor localization technique using active RFID. The basic idea is to use expensive RFID readers at fixed positions and also inexpensive RFID tags as landmarks which can record the dynamic changes due to the movement of objects/people in the environment. An analysis of power level vs distance is carried out prior to deployment. The mechanism works by calculating the Euclidean distance between a reference tag and the tag to be tracked and employing k-nearest neighbor algorithm. It has an accuracy of 2m. But, the use of extra infrastructure poses a significant overhead.

The authors have eliminated the pre deployment effort completely in the paper Indoor Localization Without Pain[3]. They do not assume the physical layout and also positioning of the Access Points. The method makes use of Wi-Fi enabled devices carried by users for crowdsourcing. In addition, a GPS lock would be obtained at an entrance to extract a location fix and this information is relayed to EZ server. A propagation model has been used to represent the system. Since a propagation model is used, this scheme does not accommodate changes in the environment.

1.2 Fault Tolerance

Most of the previous works on Indoor localization have focused on improving the accuracy which is our primary goal. The secondary goal of our work is to make our method fault tolerant i.e., we must be able to localize even in the case of faults. We intend to make the system fail safe i.e., it continues to function even in the case of faults instead of failing completely. We also obtain a threshold of the number of faults which if exceeded would make the system degrade gracefully. If the number of faults is below this threshold, then the faults are masked. We use the terms faults and failures synonymously in the following sections.

We have followed the traditional fault tolerance procedure consisting of fault confinement, fault detection and fault masking steps. We assume that the faults/failures occurring at the Access Points are independent of each other which confines the faults. We do not consider instances such as power failures which affect all the APs at once. This helps confine the faults and also makes them independent of each other.

Fault detection has been achieved by means of deploying Wi-Fi enabled devices at different locations in such a way that the status of all APs is monitored. Fault masking could be achieved by means of altering the number of APs used in the Maximum Likelihood Estimator module. In other words, the failed APs would not be used during the analysis during the online stage. Though this would lead to an increase in the localization error, our method makes sure that the error is below a threshold.

Though the works have addressed key factors which determine the Localization process, most of them do not have an upper bound on the localization error. Moreover, almost all of them do not consider the case of AP failures/shutdown which is a common issue if public WLANS are being used. The most important differences between our model and the previous models are as follows.

- The need to study the complex theory of signal propagations has been eliminated.
- No prior information of the physical layout or distribution of the APs is necessary.
- A module that takes care of the faults occurring the environment has been incorporated.
- Confidence intervals which provide a range of the position along with the estimated position facilitate a means to calculate the maximum error in the estimation.

CHAPTER 2. OUR SCHEME

2.1 System Model and Assumptions

We deploy a Client-Server model in which the server is responsible for processing the RSS records and delivering the results to the clients. Clients are the Wi-Fi enabled user devices and form the interface through which the users interact with the server to obtain the coordinates of the current position. They perform the function of recording the Received Signal Strength (RSS) of different APs at the user location. The RSS records are then pushed to the database in the server which employs various methods to accurately calculate the position of the user.

We introduce various terms and the associated assumptions before we describe our scheme. A fingerprint is defined as the RSS (in dBm) along with the coordinates of the location at which the RSS is measured. For instance (-88, (2, 3)) represents an RSS record of value -88dBm at the location (2, 3). Here, (2, 3) represents a location which is 2m to the right and 3m above the reference point. The RF map is the collection of fingerprints at various locations. We construct RF maps for each of the APs discovered. The measurements are taken in the 2D region under consideration and this area is termed as Region of Consideration (ROC).Hence our method is limited to the 2D space and all the models described further are based on it. We define localization error as the Euclidean distance between the original location and the estimated location. Our scheme relies on two basic assumptions

- Full coverage is provided in the entire ROC by at least three APs at all times.
- The smart devices used by the people are Wi-Fi enabled.

We divide our scheme into an offline phase and an online phase. The offline phase comprises of data collection and data processing. The ROC is determined during this phase and a convenient point of reference is fixed. The signal strength is then obtained all over the ROC as a function of the distance from the reference to form the RF map of the region. Once the map is obtained, the next step is to find the minimum number of locations in ROC such that each AP's range would contain at least one location. In other words, the union of the set of the fingerprints at these locations would have the beacons from all the APs in the ROC. These locations are then utilized for continuous monitoring of the status of the APs. Since the ROC usually consists of some locations like offices, the smart devices/systems in the offices can be used to monitor the access points. An android application has been designed which sends regular updates of the APs and acts as a monitor for the environmental changes.

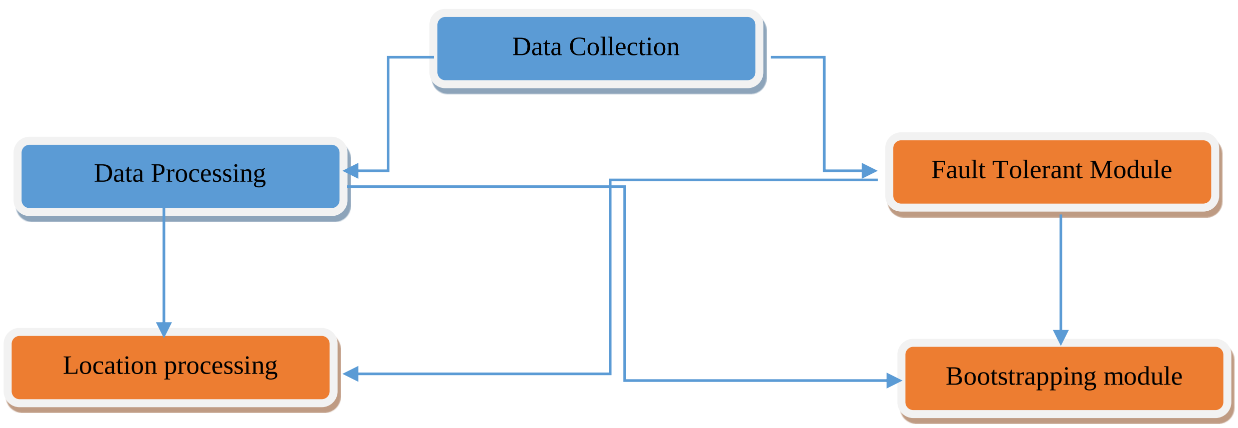

Fig. 1.1 Proposed scheme with Offline and Online phases

The online phase is completely client based and is responsible for reporting the location to the user. It consists of a location processing, fault tolerant and bootstrapping module. The user with a Wi-Fi enabled device measures the RSS at a location of interest and this data gets pushed to the database of the server. The fault tolerant module then submits the status of the various APs to the server and the data is then processed at the server. It requires the server to choose the method relevant to the status received and the location obtained is sent to the client. Also 95% confidence intervals are calculated for each location datum pushed to the server.

2.2 Offline Phase

2.2.1 Data Collection Module

This is the key step in our scheme which builds the RF map. Though it requires significant effort, once an RF map is obtained for an ROC, it can be used to localize any number of users. Another implication of this step is to obtain the “Monitor Points”, which are the locations to monitor the status of the APs. Initially, the ROC is decided upon and is equipartitioned into smaller regions. The RSS is then recorded at each of the smaller regions to give rise to a complete RF map. The RF map would be used in the data processing module to obtain a surface to be used in the online surface to infer the location corresponding to the fingerprints recorded. Also, it is handed over to the fault tolerant module. Thus this step is basically used to construct a training set which would be useful to obtain the signal strength distributions over the ROC.

2.2.2 Data Processing Module

Data obtained from the collection module is fitted to surfaces to obtain signal strength distribution which could then be made use of in the online phase to find location corresponding to the RSS measured. A surface fitting module fits the RSS readings corresponding to each AP obtained to the position so that a function of RSS with respect to the position is obtained.
The LOESS method is used to fit the surfaces which is a non parametric method for estimating regression surfaces. LOESS is used to find a regression function g_i for each point in the set of points $\{x_1,y_1\},\{x_2,y_2\},\{x_3,y_3\}\ldots\{x_n ,y_n\}$ such that

$$y_i = g(x_i) + \in_i \tag{2.1}$$

Where $\in$ is the regression error.

If there are k fingerprints, the LOESS model on the training set can be expressed as in (1).

$$s_j = g(x_j, y_j) + \in_j \tag{2.2}$$

In (1), s is the signal strength distribution, g is the regression function, (x_j,y_j) are the coordinates of the location and $\in_j$ is the regression error. The degree of the polynomial g is a predefined parameter and in our case it is 2 since we use the quadratic model.

$$g(x_j, y_j) = \beta_{i,0} + \beta_{i,1}x + \beta_{i,2}x^2 + \beta_{i,3}y + \beta_{i,4}y^2 + \beta_{i,5}xy \tag{2.3}$$

L is the smoothing matrix which defines the linear relationship between fitted and observed dependent variable value.

$$\not{R} = LR$$

Where $\not{R}$ denotes the fitted RSS and R is the observed RSS.

LOESS fits each of the training points using the local points which is specified by the parameter k. k is given by

$$k = nq \tag{2.4}$$

where q is the smoothing parameter and n denotes size of the training set. It employs a tricube function to determine the weights for a point at (x_j,y_j). If d represents the maximum distance between (x_i,y_i) and (x_j,y_j) where (x_j,y_j) lies in the local neighborhood defined by k, then weight of (x_j,y_j) is given by

$$w(x_j, y_j) = (1 - \frac{(|(x_j, y_j) - (x_i, y_i)|)^3}{d^3})^3 \tag{2.5}$$

Using this equation and the quadratic equation of s, it minimizes the following quantity

$$Q = \sum w(x_j, y_j)(s_j - (\beta_{i,0} + \beta_{i,1}x + \beta_{i,2}x^2 + \beta_{i,3}y + \beta_{i,4}y^2 + \beta_{i,5}xy))^2 \tag{2.6}$$

Hence LOESS needs to estimate $\{\beta_{i,0}, \beta_{i,1}, \beta_{i,2}, \beta_{i,3}, \beta_{i,4}, \beta_{i,5}\}$.

An important assumption we make is that the regression error satisfies some normal distribution which leads to the conclusion that the signal strength s also is normally distributed.

$$\in = N(0, \sigma^2) \tag{2.7}$$

$$s = N(g, \sigma^2) \tag{2.8}$$

Where σ^2 is the variance.

Weighted least squares method is used to fit quadratic functions of the locations at the centers of the neighborhood. A smoothing parameter called span which represents the percentage of data points in each neighborhood controls the smoothness of the estimated surface. We make use of AIC[12] to obtain an appropriate span for the surface fit. AIC provides a measure of relative quality of a model for a given dataset. Three variations of the minimization criteria namely AIC_{C1},AIC_C and GCV are available for this purpose. We have used AIC_C in our case. The LOESS procedure evaluates the criteria for a sequence of smoothing parameter values and selects the value in the sequence which satisfies the criteria using an optim function in R.

$$AIC_{C1} = nlog(\sigma^2) + n\,\frac{\delta_1/\delta_2(n+v)}{\frac{\delta_1^2}{\delta_2}-2} \tag{2.9}$$

$$AIC_C = log(\sigma^2) + 1 + \frac{2(Trace(L)+1)}{n-Trace(L)-2} \tag{2.10}$$

$$GCV = \frac{n\sigma^2}{(n-Trace(L))^2} \tag{2.11}$$

Where n is the number of points

$$\delta_1 = Trace(1-L)^T(1-L) \tag{2.12}$$

$$\delta_2 = Trace((1-L)^T(1-L))^2 \tag{2.13}$$

$$v = Trace(L) \tag{2.14}$$

Where L is the smoothing matrix and σ^2 is an average residual sum of squares.

From the collection module, an RF map with k APs and n locations is obtained. The regression is applied to each fingerprint to obtain k regression surfaces corresponding to k APs. Thus the normal distributions corresponding to k APs are stored in the server to be used during the online phase.

2.3 Online Phase

2.3.1 Location Processing Module

This module takes the RSS readings of various APs as measured by the user devices and outputs the corresponding location estimate. A maximum likelihood estimator is deployed which works as follows. A likelihood function is a function of the parameters of the model and maximum likelihood estimation provides an unbiased minimum variance estimation where the estimates are normally distributed. Since log of a likelihood function is also an increasing function we use the log function in the parameter estimation.

The log likelihood function for the set of signal strength received from all APs at a location is given by(2) where $\emptyset$ denotes the standard normal density function, s_j is the distribution of jth AP obtained using the LOESS module and σ_j is the corresponding standard deviation. We use optimization as a method for maximizing this log likelihood function $l(x, y)$ which gives the position of a set of RSS readings.

$$l(x, y) = \sum \log(\emptyset\left(\frac{s_j - g_j(x,y)}{\sigma_j}\right)) \tag{2.15}$$

The above function when maximized gives the parameters, in our case the coordinates of the location.

For maximising the loglikelihood function, we have made use of SANN optimization which provides a global maximum.

2.3.2 Bootstrapping Module

The basic idea of bootstrapping[17] is to produce a large number of copies of the sample statistic and then obtain the sampling distribution of the surrogate population to study the behavior of the statistic over a large number of samples called bootstrap samples using a small population at hand. In our scheme we have used Bootstrapping to study the sampling distribution of the MLE estimator. Firstly, we resample the signal strength distribution obtained by regression fitting to obtain a large number, usually 2000 samples. These samples are then given as input to the MLE module which gives the position estimate corresponding to these samples. Then, a small percentage usually 2.5% is trimmed off from the lower as well as upper end of this range. The range of remaining 95% values is declared as the confidence limits of the corresponding population with level of confidence 95%.

2.3.3 Fault Tolerant Module

This is the critical part of the system which focuses primarily on providing fault tolerance in real world environments where the APs could fail or exhibit erroneous behavior. We introduce different fault models which capture such behavior and evaluate the way our proposed scheme functions in each case. The basic idea is obtained from [14]. An important assumption we make is that the faults occur after the offline phase which causes a discrepancy between the environment during the offline phase and that during the online phase. In order to deal with the various faults we propose the following technique. After the data collection phase, we run through the RF map and find locations in the map, usually offices in a mall/terminal which can accommodate a system/mobile device to act as a monitor. If a particular location is denoted by (x_i,y_i) and the corresponding RSS beacons received from j^{th} APs as s_{ij}.$\{s_i\}$ denotes the set of beacons received from all APs at the location i. We need to find the least number of locations n such that $\{s_1\} \cup \{s_2\} \cup \{s_3\} \ldots \cup \{s_n\}$ gives the complete set of APs in the ROC. These locations would be called as Monitors. We then deploy a mobile application in the smart devices of the people at these locations which provides the status of the all access points in the ROC. The Android application scans for the beacons emitted by the APs every 10 seconds and sends the beacon data received to a central server at regular intervals. The database of the central server reflects the status of the various APs in the ROC. An important assumption that we make is that none of the faults occur at the Monitors. We describe the models next and the proposed mechanism to deal with each model.

2.3.3.1 AP failure model

The model corresponds to the case where the APs used in the offline phase are not available during the online phase. Possible reasons might be power outages, system maintenance or permanent shut down of the AP. In this case the RF map created in the offline phase doesn't provide accurate results. If there are no beacons corresponding to a particular AP for a considerable amount of time, say for one hour, we assume that the AP has failed and modify the MLE to reflect the changes corresponding to the failed AP. So we use the signal strength distributions obtained from the regression module corresponding to the APs which are functioning at that point as one of the inputs to the Estimator. The other input would be the readings of the APs recorded by the users at his/her location. This method might reduce the accuracy if the number of APs falls below a threshold.

2.3.3.2 False Negative Model

Another scenario is when the beacons from the AP are not detected during the online phase even if the APs are functioning. This might be due to the movement of any structure/furniture such that it obstructs the signals from the AP. We go with our initial assumption that no faults occur at the monitors and hence they reflect the true status of the environment, i.e., in this case the beacons would be received at the monitors and would indicate the true state of the APs. So even if a reading at a location doesn't contain a record corresponding to a particular AP, we give the same set as an input to the estimator. This might affect the accuracy of localization though.

2.3.3.3 False Positive Model

In this model, the beacons from the APs not received during the offline phase would be detected. This might be due to a change in the configuration of the coverage by adding new APs. In this case the beacons received from the APs that are not a part of the offline phase signals, are not used for localization. This is because, the training dataset doesn't contain the fingerprints corresponding to the APs newly discovered which implies that they are not a part of the initial set of APs.

CHAPTER 3. EXPERIMENTAL ANALYSIS

Our experimental test bed is located in the basement of the Howe hall of Iowa State University. We have used a region which measured 12m by 10m as our ROC divided into square regions of dimensions 1m by 1m. The ROC has been colored black in Fig. 1.We have used a free space to build the map initially and then have moved furniture into the area during the online phase to study the effect of reflections due to various objects in the environment. A Lenovo Thinkpad T440S model laptop running Windows 8 equipped with AirPcap Classic adapter has been our capture hardware. Wireshark has been used in conjunction with AirPcap adapter to capture the beacons along with the associated RSS. An android application which can be run on any Android device has been developed which scans for the Wi-Fi beacons every 10 seconds and then pushes it to a server. A WAMP server has been used for this purpose which runs on a Lenovo laptop. The entire statistical programming has been done using R Studio. The Android SDK was used for developing the Android application [18].

3.1 Offline phase

In this phase, we collect the beacon information at 1m intervals spanning the entire ROC. We limit our Localization process to 2.4GHz band. The AirPcap adapter makes available the RSS of the beacons received in units of dBm. Power in mW can be converted to dBm using the conversion formula 10*log10(mW). Next, we obtain the fingerprints of the entire ROC in the format (dBm,(x,y)) with the person standing in front of the laptop. The collection process is carried out with no change in the person’s orientation over the ROC.

The process is repeated for the APs in the three channels namely 1,6 and 11 corresponding to 2412 GHz,2437 GHz and 2462 GHz bandwidths. At each location we capture

5000 beacons and average the RSS to obtain a mean RSS. We use the mean RSS for the rest of the sections. So at the end of the data collection stage we have RF maps corresponding to each AP.

The next step is then carried out to determine the positions of the monitors. The RF maps are then studied to obtain the locations for deploying the monitors which might be the offices of employees with their mobiles running the android application to regularly monitor the status of the APs. In our case we have the application running at the position colored red indicated in the map in Fig. 3.1.

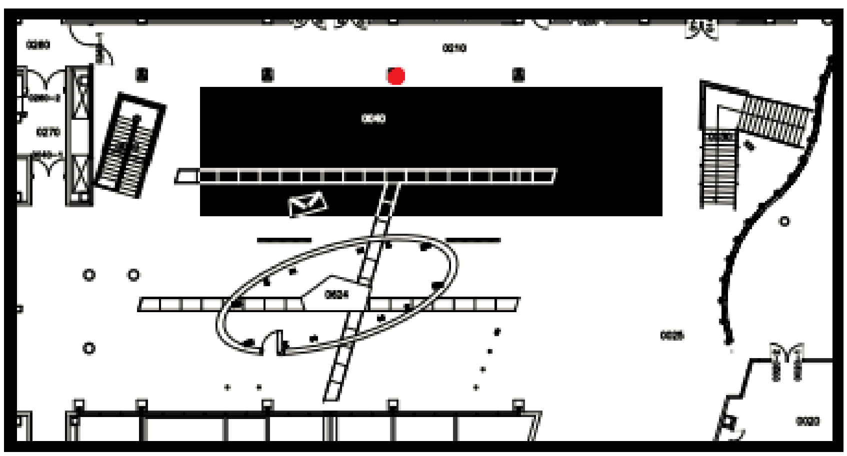

Fig. 3.1 Floor plan of the Howe Hall Basement

We then use the data processing module to fit the fingerprints to a surface, so that we have signal strength distribution in the ROC for each AP. This gives us n surfaces corresponding to n APs. The LOESS regression module has been used to fit the readings to the positions to obtain the curve with the span obtained by minimizing the error by using the AIC[12].

3.2 Online phase

We have users moving at random in the ROC and measuring the RSS using the Wi-Fi enabled device. This is then pushed to the WAMP server. The RSS readings are then provided as an input to the location processing module. In our case, the server can calculate the position of one user at any given time. The reading would be in the form of (SSID, BSSID, RSS, Frequency, Timestamp). The BSSID would give the details of the AP address and the RSS is the signal strength in dBm. At each location 100 samples are collected and then the mean of the readings is taken. The set of readings in the form of (BSSID, Mean RSS) are fed to the Estimator which uses the previously calculated Signal Strength distributions to perform Maximum Likelihood Estimation. Also provided to the server is the status from the Monitors which is essential to provide fault tolerant solution to the localization problem.

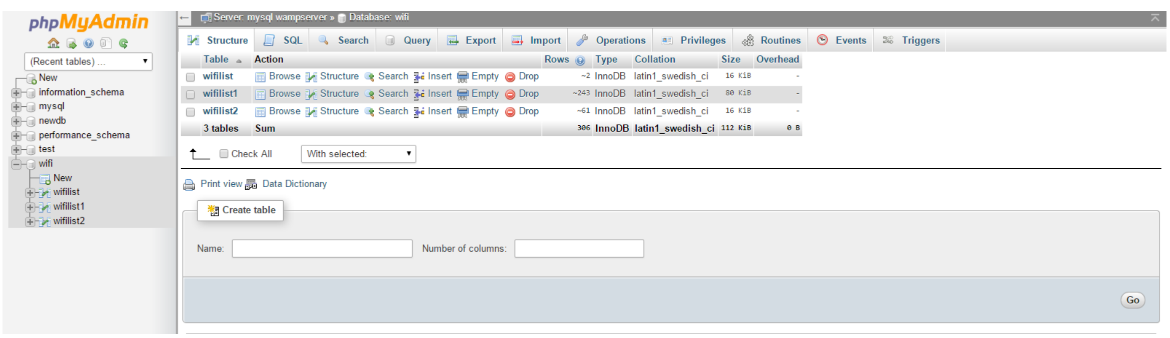

Fig. 3.2 Screenshot of the MySQL database

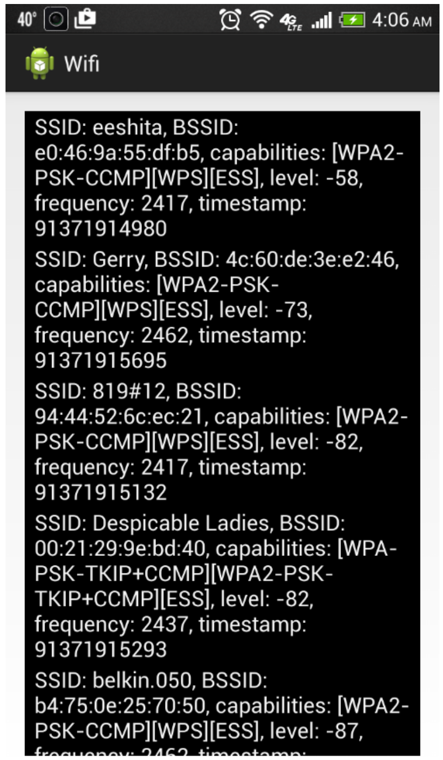

Fig. 3.3 Screenshot of the Android application

3.2.1 Temporal Analysis

In this case we have selected random locations in the ROC and measured the RSS at these locations during three different times. The first set of readings (T1) was taken during the offline phase along with the training data. The second set of readings (T2) was taken after arranging furniture in the test bed past midnight when the number of people in the ROC is the minimum. The third set of readings (T3), was taken during a busy time, 10 minutes prior to the beginning of the class. From Fig. 3.4, we see that the though there is some variation in the results obtained from different sets, the variation is not very significant. Fig. 3.5 shows the mean localization error for the three cases. Thus we can conclude that the method is temporally robust as long as the APs function.

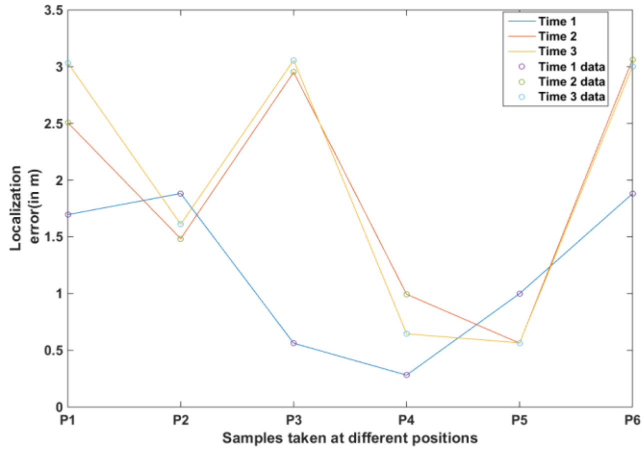

Fig. 3.4 Localization error at different positions in the ROC during three different times

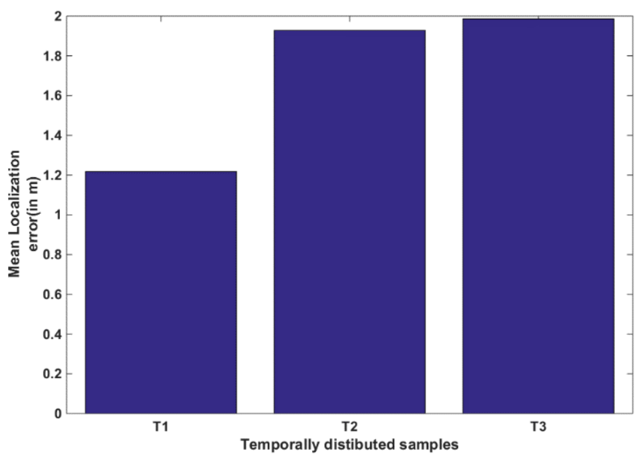

Fig. 3.5 Mean Localization error of temporally distributed samples

3.2.2 Effect of the number of APs

We have obtained the fingerprints all over the ROC and it was observed that there were 8 APs which are responsible for providing Wi-Fi coverage. Our objective in this case was to obtain the minimum number of APs which are needed to maintain the localization error up to a certain threshold. We fixed the threshold to be 2m, hence we find the minimum number of

APs needed to participate in the localization process in order to restrict the localization error to 2m. From the results interpreted in Fig. 3.6, we observe that the mean localization error depends on the number of APs. It sharply reduces from 3.74m to 1.52m when the number of APs increases from 2 to 3 but thereafter, a significant change has not been observed. However another acute change occurs when all the 8 APs are employed thereby reducing the mean error to 1.38m. Hence in our ROC we find that the minimum number of APs required to limit the mean localization error to 2 m is 3.

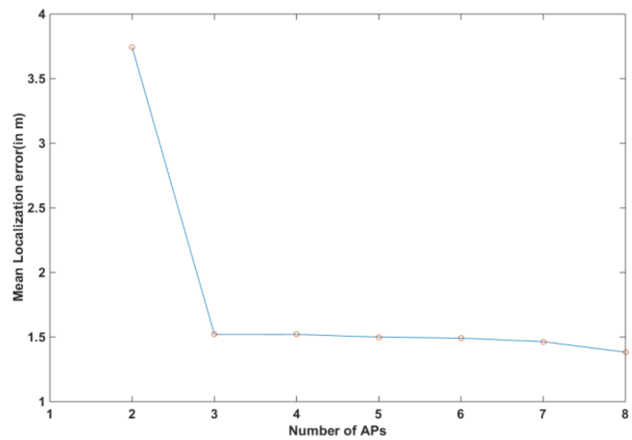

Fig. 3.6 Relation between the number of APs and the Mean Localization error

3.2.3 Location determination along a path

We select a path which spans the entire ROC to observe the distribution of localization error over the path. A total of 16 locations were selected in such a way that the new set and the original training set are mutually exclusive. From Fig. 3.7, we observe that the error at the edges of the ROC is very high when compared to that at other points. This can be explained since the regression uses the neighborhood to obtain the surface plot and the number of neighborhood locations at the edges is low.

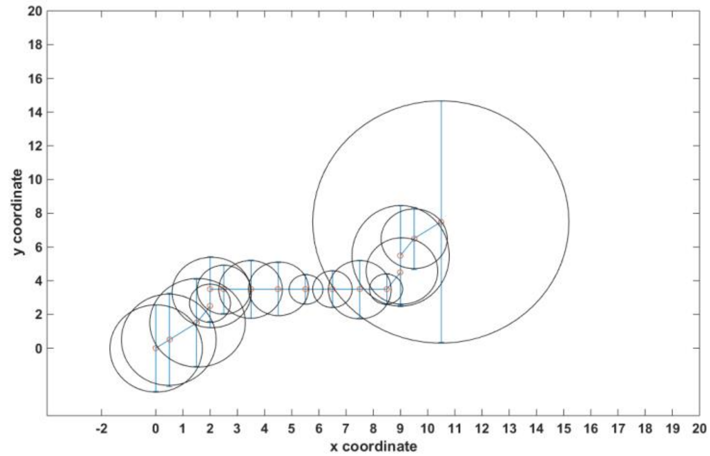

Fig. 3.7 Localization error along a path with 8APs

3.2.4 Failure of an AP

This case uses the same path which has been established in the previous case. To simulate the situation of 1 AP failure, we choose an AP and substitute the readings corresponding to it across all the positions along the path with zeros. We then give this set of readings as input to the estimator. This procedure is then repeated for the rest of the APs i.e., we assume that each AP fails once and then construct of training sets and use them to obtain the position. Then the mean localization error is obtained at each position. As discussed earlier, if there were a failure it would be detected by the Monitor and the estimator would be notified about it. In our case, to simulate this fault tolerant behavior, we drop the fingerprints corresponding to the failed AP and hence prevent the inclusion of this AP in further estimation. Fig. 3.8 represents the results corresponding to the fault tolerant and non fault tolerant cases. We observe that there is a commendable difference between the performance of our localization scheme with and without fault tolerance. Some outliers however have been observed.

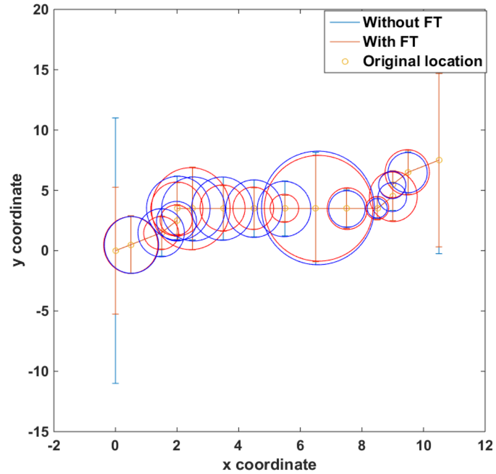

Fig. 3.8 Localization error in the case of an AP failure

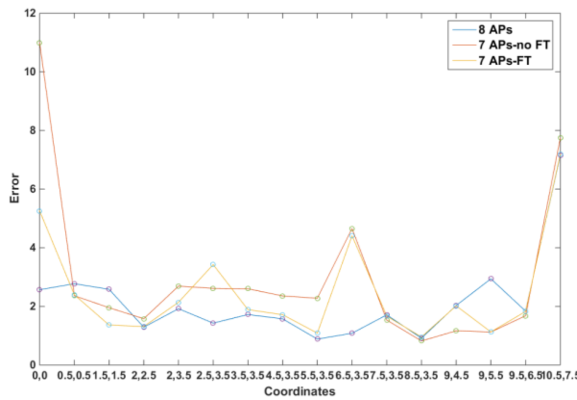

Fig. 3.9 Comparison of localization error with 8 APs and 7 APs

3.2.5   Failure of two APs

In this case, we use six APs for localization in the fault tolerant case and substitute the readings of the other two APs with zeros. Since we obtain a large number of sets consisting of two APs, we select the 4 APs, 2 with maximum coverage and 2 with minimum coverage. We then take all possible combinations of these 4 APs which gives 6 sets of training data. We use these sets to obtain the locations of the points as explained before and plot the mean localization error at each location. The difference in the error between the two cases is quite significant than the previous case. Thus it could be inferred that as long as a single fault at AP occurs, it is masked and any number of faults higher than 1 would increase the localization error. However the system would degrade gracefully as depicted in the graph below. Since our model assumes that faults don't occur at the monitors, it can detect the situations of AP failure. If the beacon from a particular AP is not recorded at a particular location, the database is checked to see if the AP is alive, if it is then it is assumed as a false negative and the original training set is used. Else, the training set excluding the failed AP is used to process the location.

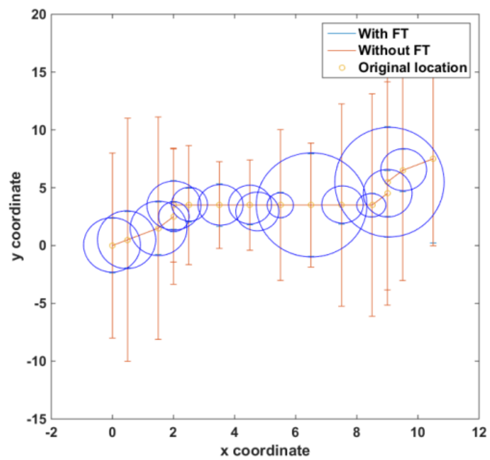

Fig. 3.10 Localization error in the case of two AP failures

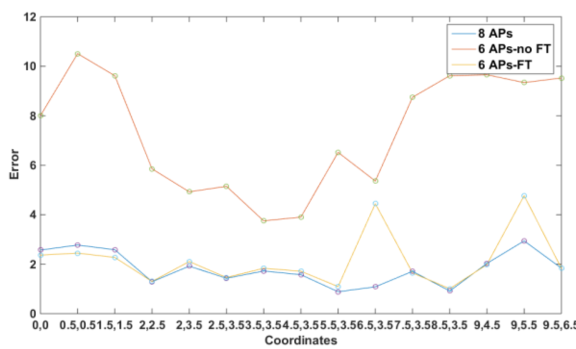

Fig. 3.11 Comparision of localization error with 8APs and 6 APs

3.2.6 Different hardware

We have verified the validity of our method on different devices by comparing the Localization error along a path in the ROC when the online phase was carried out using an HTC One X+ running on Android Version 4.2.2 and a Lenovo laptop running on Windows 8 version. Fig. 3.12 shows a comparison of means of errors when the online phase readings were recorded using the previously mentioned hardware devices.

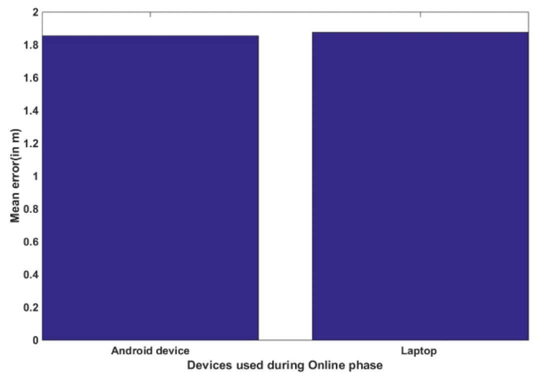

Fig. 3.12 Mean Localization error with an Android device and a laptop

3.2.7 Bootstrap Analysis

In order to obtain the 95% confidence intervals for the location estimates, we resample the signal distributions obtained by regression to obtain 2000 samples. These samples form the new training set and are used to find the position of each set of readings collected during the online phase. Thus for each location readings, we have 2000 location estimates. The next step is to find the 2.5% and 97.5% quantiles which give the lower and upper bounds for the 95% confidence intervals. We have repeated the process for 7 locations in the ROC and the results have been presented in Table 3.1.

TABLE 3.1 CONFIDENCE INTERVALS

Original Location in (m,m)	**Location estimate in (m,m)**	**Lower bound (m,m) in**	**Upper bound (m,m) in**
(4,7)	(3.456226,7.08)	(1.21,5.297)	(2.545,7.68)
(5,7)	(6.37,10)	(4.962,8)	(6.475,8)
(6,7)	(4.405,8)	(3.892,8)	(7.32,6.417)
(7,7)	(6.44,7.05)	(5.683,6.14)	(8.66,7.78)
(9,7)	(8.77,6.836)	(6.784,6.644)	(8.66.7.78)
(10,7)	(8.01,8)	(4.876,8)	(8.9,4.99)
(11,7)	(11,8)	(11,6.494)	(11,8)

CHAPTER 4. CONCLUSION

We have achieved a fault tolerant solution in the case where the faults which occur are limited to AP failures and reception of false negatives and false positives. We intend to address the case where there is relocation of one or more AP(s). Can the monitors detect the relocated APs? If yes, can a relevant fault tolerant solution be provided? Also, we would like to extend our work to the 3D space which would be more relevant to the multi floored indoor spaces. Another issue to be considered is to come up with a method which eliminates the errors at the edges of the ROC due to limited neighborhood. A basic idea to address this issue would be to take help of GPS as the edges usually consist of windows/exits and then use this in conjunction with our localization scheme.

In this work, we have discussed a scheme to localize a user indoors without the need of a physical layout or information regarding placement of the APs. Further, a procedure to deal with various fault models has been discussed. We have successfully localized a user in the ROC within 2m of the actual position in spite of the dynamic environment. A proof of the localization error along a path spanning the ROC also has been provided. Also, we have shown that the minimum number of APs needed to have the localization error less than 2 m as 3. Thus we have provided a fault tolerant solution to the Indoor Localization problem which is a significant contribution to the research going on in this area.

APPENDIX A

R CODE

PARAMETER CALCULATION SATISFYING AKAIKE'S INFORMATION CRITERION

```
#x is the loess object

loess.aic<-function(x){

  span<-x$pars$span

  n<-x$n

  traceL <- x$trace.hat

  sigma2 <- sum( x$residuals^2 ) / (n-1)

  delta1 <- x$one.delta

  delta2 <- x$two.delta

  enp <- x$enp

  aicc <- log(sigma2) + 1 + 2* (2*(traceL+1)) / (n-traceL-2)

  aicc1<- n*log(sigma2) + n* ( (delta1/delta2)*(n+enp)/(delta1^2/delta2)-2 )

  gcv  <- n*sigma2 / (n-traceL)^2

  result <- list(span=span, aicc=aicc, aicc1=aicc1, gcv=gcv)

  return(result)

  }

bestLoess <- function(model, criterion=c("aicc", "aicc1", "gcv"), spans=c(0.05,0.95)){

  criterion <- match.arg(criterion)

  f <- function(span) {

  mod <- update(model, span=span)

  loess.aic(mod)[[criterion]]
```

```
 }
 result <- optimize(f, spans)
 list(span=result$minimum, criterion=result$objective)
 }
```

LIKELIHOOD ESTIMATION

```
#param is the seed for the optimization
#n is the number of APs
#k is the array of RSS readings recorded in the online stage
#s is the training set
LoL <-function(param,k,s,n)
{
 d<-data.frame(x=param[1],y=param[2])
 m<-array(1:n)
 se<-array(1:n)
 TheS<-array(1:n)
 TheSD<-array(1:n)
 l<-array(1:n)
 p<-0
 for (i in 1:n){
  b1<-bestLoess(loess(s[i,]~x+y))
  b<-loess(s[i,]~x+y,span=b1$span)
  m[i]<-predict(b,d,se=TRUE)$fit
```

```
   se[i]<-predict(b,d,se=TRUE)$se
   TheS[i]<-loess(s[i,]~x+y)$s
   TheSD[i]<-sqrt(se[i]^2 + TheS[i]^2)
   l[i]<-log(dnorm(k[i],mean=m[i],sd=TheSD[i]))
#log of the likelihood function
   p<-p-l[i]
#sum of the log likelihood functions
  }
  return(p)
}
```

MAXIMUM LIKELIHOOD FUNCTION

```
MLE<-function(param,k,s...)
{
  return(optim(param,LoL,s,k,...))
}
```

BOOTSTRAP FUNCTION

```
bootsignal<-NULL
#this for loop calculates 2000 training sets using the mean and sd of the original training set
for(i in 1:n)
{
  b1<-bestLoess(loess(s[i,]~x+y))
  b<-loess(s[i,]~x+y,span=b1$span)
  a<-predict(b,se=TRUE)
```

```
  m<-a$fit
  sd<-b$s
  tp<-matrix(rnorm(length(m)*2000,mean=m,sd=sd),byrow=TRUE,nrow=2000)
  bootsignal<-c(bootsignal,list(tp))
}
#Here we are calculating the confidence intervals for 8 APs
posn<-array(1:2000)
for(bno in 1:2000)
{
   d<-array(1:864,dim=c(8,108))
   d[1,]<-bootsignal[[1]][bno,]
   d[2,]<-bootsignal[[2]][bno,]
   d[3,]<-bootsignal[[3]][bno,]
   d[4,]<-bootsignal[[4]][bno,]
   d[5,]<-bootsignal[[5]][bno,]
   d[6,]<-bootsignal[[6]][bno,]
   d[7,]<-bootsignal[[7]][bno,]
   d[8,]<-bootsignal[[8]][bno,]
   v<-optim(c(0,0),LoL,gr=NULL,k,d)
}
dposn<-array(1:2000)
for(i in 1:2000)
{
```

```
  dposn[i]<-posn[[i]]$value
}
#calculate the lower and upper quantiles
quantile(dposn,probs=c(0.025,0.5,0.975))
```

REFERENCES

[1] P. Bahl, V. Padmanabhan, and A. Balachandran, “Enhancements to the RADAR User Location and Tracking System,”Technical Report Technical Report MSR-TR-2000-12, Microsoft Research, February 2000.

[2] P. Bahl and V. N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system,”Proceedings of the IEEE International Conference on Computer Communications (INFOCOM’00), pp. 775–784, March 2000

[3] K. Chintalapudi, A. Padmanabha Iyer, and V. N.Padmanabhan,“Indoor localization without the pain”, Proceedings of the sixteenth annual international conference on Mobile computing and networking (MobiCom ’10), pp. 173–184, 2010.

[4] M. Youssef and A. Agrawala, “The Horus WLAN Location Determination System,” Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, MobiSys, pp 205-218.

[5] T. S. Rappaport. Wireless Communications-Principles and Practice,Prentice Hall Communications Engineering and Emerging Technologies,Series, 2 edition, 2005.

[6] Ni, Lionel ; Liu, Yunhao ; Lau, Yiu ; Patil, Abhishek, “LANDMARC: Indoor Location Sensing Using Active RFID,” Wireless Networks, 2004, Vol.10(6), pp.701-710

[7] T. Roos, P. Myllymki, H. Tirri, P. Misikangas, and J. Sievnen, “A probabilistic approach to WLAN user location estimation,” International Journal of Wireless Information Networks, vol. 9, 2002.

[8] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo, “Zero-configuration, robustindoor localization: Theory and experimentation,” Wireless Networks, 2010, Vol.16(2), pp.405-420

[9] S. Yang, J. Yi, and H. Cha. Hcrl, “A hop-count-ratio based localization in wireless sensor networks,” SECON, 2007.

[10] Souvik Sen , Romit Roy Choudhury , Srihari Nelakuditi, “SpinLoc: spin once to know your location,” Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications, February 28-29, 2012.

[11] Anshul Rai , Krishna Kant Chintalapudi , Venkata N. Padmanabhan , Rijurekha Sen, “Zee: zero-effort crowdsourcing for indoor localization,” Proceedings of the 18th annual international conference on Mobile computing and networking, August 22-26, 2012.

[12] H. Akaike, “A new look at the statistical model identification,” IEEE Transactions on Automatic Control, Vol. AC-19,No.6,December 1974.

[13] William S. Cleveland and Susan J. Devlin, “Locally weighted regression: An approach to regression analysis by local ftting,” Journal of the American Statistical Association, Vol. 83, No. 403,pp.596-610, 1988.

[14] C. Laoudias, M. P. Michaelides, and C. G. Panayiotou, "Fault tolerant positioning using WLAN signal strength fingerprints," in International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2010, pp. 1-8.

[15] Garcia-Valverde, Teresa ; Garcia-Sola, Alberto ; Hagras, Hani ; Dooley, James A. ; Callaghan, Victor ; Botia, Juan A, “A Fuzzy Logic-Based System for Indoor Localization Using WiFi in Ambient Intelligent Environments,” IEEE Transactions on Fuzzy Systems, Aug. 2013, Vol.21(4), pp.702-718

[16] Gao, Chunwang , “A statistical indoor localization method for supporting location-based access control,” Graduate Theses and Dissertations Digital Repository @ Iowa State University 2009

[17] Singh, K., Xie, M., “Bootstrap: A Statistical Method,” http://www.stat.rutgers.edu/home/mxie/rcpapers/bootstrap.pdf

[18] T.Ravi, ”How to connect Android with PHP,MySQL,” http://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/