AN INDOOR POSITIONING SYSTEM BASED ON ROBUST LOCATION FINGERPRINT FOR WI-FI AND BLUETOOTH

A.K.M. MAHTAB HOSSAIN

NATIONAL UNIVERSITY OF SINGAPORE

2009

AN INDOOR POSITIONING SYSTEM BASED ON ROBUST LOCATION FINGERPRINT FOR WI-FI AND BLUETOOTH

A.K.M. MAHTAB HOSSAIN

(*B. Sc., Bangladesh University of Engineering & Technology (BUET),*

M. Eng., Asian Institute of Technology (AIT), Thailand)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2009

Acknowledgement

I would like to dedicate this dissertation to my mother and my eldest sister. It had been a long journey and they were always a motivation for me. I am indebted to my supervisor Dr. Wee-Seng Soh for his continuous support and encouragement. His guidance and valuable suggestions have certainly improved the quality of my research work. I also take the opportunity to thank all my colleagues in the ECE-I^{2}R Wireless Communications Laboratory for their warm friendship and help. I gratefully acknowledge the financial support from the following entities: National University of Singapore for awarding the research scholarship throughout my candidature, and also the Ministry of Education of Singapore for funding our project.

Contents

Summary

A desirable indoor positioning system should be characterized by good accuracy, short training phase, cost-effectiveness (using off-the-shelf hardware), and robustness in the face of previously unobserved conditions. This dissertation aims to achieve an indoor positioning system that accomplishes all these requirements. First, the current signal strength based location fingerprints regarding two well-known RF technologies, namely, Wi-Fi and Bluetooth are elaborately discussed. As it will be explained, their RF signal parameters have specific purposes that render them inappropriate for consideration as location fingerprints. Subsequently, a *robust* location fingerprint, the *Signal Strength Difference (SSD)* is derived analytically, and then verified experimentally as well. A simple linear regression interpolation technique, and the application of *user feedback* to facilitate under-trained positioning systems have also been investigated. These techniques reduce the training time and effort. The results of two well-known localization algorithms ($\mathcal{K}$-Nearest Neighbor and Bayesian Inference) are presented when the proposed ideas are implemented.

List of Figures

List of Tables

List of Symbols

$\in$	is an element of
$\propto$	proportional with
e^x or $\exp\{x\}$	exponential function
$\ln x$	natural logarithm of x
$\log x$	base-10 logarithm of x
$E\{\cdot\}$	(statistical) mean value or expected value
$\lvert\boldsymbol{A}\rvert$	determinant of matrix $\boldsymbol{A}$
$\lVert\cdot\rVert$	Frobenius norm
$\prod_{i=1}^{N}$	multiple product
$\sum_{i=1}^{N}$	multiple sum
$\lvert a\rvert$	absolute value of a number
$N(\mu, \sigma^2)$	Gaussian random variable with mean μ and variance σ^2
$P(d_k)$	received signal strength at a distance d_k from the transmitter

List of Abbreviations

AP	Access Point
BER	Bit Error Rate
BT	Bluetooth
CDMA	Code Division Multiple Access
CRLB	Cramér Rao Lower Bound
DSSS	Direct Sequence Spread Spectrum
GUI	Graphical User Interface
LQ	Link Quality
MLE	Maximum Likelihood Estimate
MN	Mobile Node
NIC	Network Interface Card
OFDM	Orthogonal Frequency Division Multiplexing
p.d.f.	probability density function
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SNR	Signal-to-Noise Ratio
SQ	Signal Quality
SSD	Signal Strength Difference
TPL	Transmit Power Level
w.r.t.	with respect to

Chapter 1

Introduction

1.1 Overview

Location awareness is expected to be an integral part of future ubiquitous or pervasive wireless computing environment [1]. Pervasive or ubiquitous computing basically refers to the emerging trend towards numerous computing devices connected to an increasingly ubiquitous network infrastructure [2]. To reap the benefits of pervasive computing, location awareness becomes obligatory for many reasons, e.g., to access various location-based services that could be available. Therefore, indoor location determination for mobile nodes (MNs[1]) poses a significant challenge for the successful realization of such environment.

Location determination or *localization* refers to the procedure of obtaining location information of an MN with the help of a set of reference nodes (e.g., access points (APs[2])) within a predefined space. In the literature, this localization process can also be seen to be termed as *radiolocation* [3, 4], *geolocation* [5], *location sensing* [6, 7]

[1]This dissertation will use the term "MN" to indicate the people carrying devices, equipment, or other tangibles that need to be located.

[2]This dissertation will primarily use the term "APs" to indicate the reference nodes/points utilized for localization.

or *position location* [8], etc. A system deployed to estimate the location of an entity (e.g., MN) is called a *positioning system* or *location system*. An indoor positioning system generally refers to a network infrastructure within a building that helps to provide location information to any requesting end user. This location information can be reported in the form of a set of coordinates, or a combination of a floor number and a room number, or relative to some other reference objects' positions within the building. Note that the terms *location* and *position* will be used interchangeably throughout this thesis.

The application of indoor location information could range from helping firefighters to navigate through a building in an emergency situation to the more traditional assets/objects/personnel tracking. It also enables the users to become aware of many location-based services, e.g., sending the print jobs to the nearest printer, guidance services in a museum or exhibitions, targeted advertising, etc. In the field of robotics, a robot can navigate by itself with the assistance of an indoor positioning system [9]. Various smart home applications (e.g., automatically turning on/off different appliances to conserve energy depending on a user's location) are built upon location information as well. These are just a few examples from a wide range of applications that relies on indoor location information.

This chapter first presents the background of indoor localization and identifies some challenges associated with it. Next, the scope of the research, and the contributions are briefly discussed. Finally, the organization of this dissertation is outlined.

1.2 Background

As pointed out before, to reap the benefits of pervasive computing, the knowledge of a device's location with some degree of accuracy is obligatory regardless of its position (i.e., indoor or outdoor). The Global Positioning System (GPS) [10,11] solves

the localization problem in outdoor environments. However, it could not become the overwhelming solution for the localization problem as a whole, namely, because,

- GPS performs poorly in indoor environments because of its weak signal reception inside the buildings.
- Moreover, for small, cheap and low-power devices (e.g., sensors), it is not practical or feasible for them to be all GPS-enabled.

As a result, an alternative means is required to detect the MN's location in indoor environments. One way is to set up an infrastructure based on infrared [12], radio frequency (RF) [13, 14], ultra sound [13, 14], or ultra wide band (UWB) [15] technologies inside a building just for localization purpose. The measurements obtained from these sensors are converted into some metric such as distance or angle, which is subsequently utilized by the localization algorithm to estimate the MN's location. The widespread availability of wireless network infrastructure within homes, offices, and campuses opened the door for another alternative solution for indoor localization. It allows the design of an easily deployable low-cost positioning system. The wireless network interface card (NIC) which measures RF signal strength can be considered as a kind of sensor device. Location information is provided as a value-added service for such networks that are primarily set up for data communication.

Unlike outdoors, the indoor environment poses different challenges for location determination due to the multi-path effect and building material dependent propagation effect. Multi-path is a radio frequency phenomenon which is the result of radio signals traveling through multiple reflective paths from a transmitter to the receiver, and thereby, causes fluctuations of the received signal's amplitude, phase, and angle of arrival [16]. As a result, the RF signal strength measurement for wireless NIC, and the subsequent conversion of the metric (e.g., distance, angle, etc.) from it have not yielded satisfactory outcomes for localization algorithms [17]. On the contrary,

location fingerprinting technique that exploits relationship between any measurable physical stimulus (e.g., RF signal strength) and a specific location is shown to perform quite well [17]. This technique subsequently became very popular for positioning systems that utilize in-building communications infrastructure (e.g., Wi-Fi, Bluetooth, etc.) [18–24]. The deployment of such fingerprint based positioning system usually consists of two phases – *offline* training phase and *online* location estimation phase. These two phases are described briefly in the following.

During the offline phase, the location fingerprints (e.g., signal strength samples) at the selected locations of interest are collected, yielding the so-called *radio-map* [17]. In order to differentiate among various locations, the entire area is usually covered by a rectangular grid of points. During the online location determination phase, the signal strength samples received at the APs from the MN, or vice versa, will be sent to a central server. The server then uses some algorithm to estimate the MN's position, and reports it back to the MN (or the application requesting the location information). The most common algorithm used to estimate the location computes the Euclidean distance between the online measured sample and each fingerprint in the radio-map collected offline. The coordinates associated with the fingerprint in the radio-map that yields the smallest Euclidean distance is returned as the estimate of the MN's position.

From the above discussion, it is apparent that a fingerprint based indoor positioning system faces certain challenges:

- Since location information is provided as a value-added service on top of an existing network infrastructure using off-the-shelf hardware (e.g., wireless NIC), no custom sensor is manufactured as in the case of costly infrastructure-based localization discussed previously. Therefore, the positioning system cannot make any assumptions on the device types carried by the consumers, and it should be able to accommodate all the myriad types of devices (e.g., laptop, PDA, mobile

phone, etc.) that come with different hardware solutions.

- Fingerprint based positioning system is basically characterized by the exhaustive offline training phase, where the positioning system administrator strenuously collects the signal strength samples over the whole localization area. If the deployment area is quite large, this process would entail significant burden for the administrators. It could even hamper the proper installation of a positioning system if some areas are under-trained.

- Majority of the fingerprint based indoor positioning systems in the literature utilize Wi-Fi as the underlying network infrastructure because of its widespread availability. The promises of other underlying prevalent wireless technologies (e.g., Bluetooth) have been overlooked mostly.

1.3 Contributions

This dissertation is primarily a study of the RF signal strength based location fingerprints for wireless indoor positioning systems. Traditionally, the received signal strength (RSS) has been the ultimate choice as a location fingerprint for such systems. In this dissertation, we first review all the available RF signal strength parameters from a positioning system's perspective for two prevalent wireless technologies, i.e., Wi-Fi and Bluetooth. Note that, apart from the popular Wi-Fi, the prospects of various Bluetooth signal strength based parameters to serve as location fingerprints are investigated too.

The devices carried by consumers of location services are expected to come with different hardware solutions, even for the same wireless technology. As a result, a positioning system that relies solely on absolute RSS measurements to define location fingerprints would not perform well. Regardless of whether a device's signal

strengths as perceived by the APs are used to denote the device's location fingerprint (i.e., AP-based approach), or the reverse approach in which the APs' signal strengths as perceived by the device are used (i.e., MN-assisted approach), such fingerprints may differ significantly with the device's hardware even under the same wireless conditions [25–29]. This can easily be observed in existing popular wireless technologies, such as Wi-Fi or Bluetooth. The presence of power control in some wireless technologies further complicates the issue [25]. In this study, a robust location fingerprint, the *Signal Strength Difference (SSD)* is derived analytically and its effectiveness is proven experimentally as well. This particular location fingerprint's performance is shown to remain relatively unaffected with different devices' hardware variations compared to the traditional RSS. Next, the error bound of location estimation using the SSD measurements is analyzed. A novel characterization of the properties of this bound is presented that allows us to individually assess the impact of different parameters (e.g., number of APs, geometry of the APs, distance of the APs from the MN, etc.) on the accuracy of location estimates.

In the literature, the exhaustive offline training phase of the fingerprint based localization techniques is generally shortened utilizing interpolation techniques. For example, Li *et al.* [26] try to complete the radio-map database using interpolation of readings taken at other training points. The study in this thesis tries to relieve/shorten the exhaustive training phase in two ways. First, by exploiting the spatial similarity [30] of signal strength distribution, a weighted linear regression approach in order to obtain a better fit for the interpolated training points has been investigated. Second, the viability of a positioning system utilizing user feedback has been envisioned. Here, *user feedback* is defined as the information about a user's actual position indicated by the user to the system, either explicitly or implicitly.

There are certain assumptions which limit the scope of this research. For example, this study is limited to the investigation of stationary mobile devices. No mobility

tracking is considered. This study does not necessarily aim to find an optimal localization algorithm but some modifications to the baseline algorithms (e.g., the Euclidean distance technique) have been experimented with. Although this study includes Bluetooth in addition to the popular Wi-Fi technology, the hybrid approaches that combine multiple sensor technologies' data intelligently is beyond the scope of this dissertation.

The following is the summarized list of our contributions:

- Study and review all the available RF signal strength based location fingerprints for two well-known wireless technologies, i.e., Wi-Fi and Bluetooth.
- Proposed a robust RF signal strength based location fingerprint, namely, Signal Strength Difference (SSD), and verified its effectiveness over the traditional RSS as a location fingerprint both analytically and experimentally over different MNs' hardware variations.
- Analyzed the error bound of location estimation using the SSD measurements.
- Proposed two methods in order to shorten/relieve the exhaustive training phase typically seen in the fingerprint based positioning systems – i) weighted linear regression based interpolation techniques exploiting the spatial similarity of signal strength distribution, and ii) incorporating user feedback where a user indicates his/her actual position to the system, either explicitly or implicitly.
- Our ideas are implemented and tested with experimental testbeds based on both Wi-Fi and Bluetooth wireless technologies.

1.4 Organization

In Chapter 2, a literature survey of the indoor wireless positioning system is provided. Chapter 3 reviews the signal strength based location fingerprints of two well-known

wireless technologies, namely, Wi-Fi and Bluetooth, and points out their pitfalls regarding localization. In Chapter 4, a new robust location fingerprint is derived analytically and its performance is tested experimentally. Chapter 5 analyzes the Cramér-Rao Lower Bound (CRLB) of localization using the new robust location fingerprint which subsequently provides valuable insights in the positioning system design. In Chapter 6, two methods to shorten the exhaustive offline training phase typically seen in the fingerprint based positioning systems have been proposed. Finally, the conclusions and discussions of the future work are presented in Chapter 7.

Chapter 2

Literature Review

This chapter reviews the literature on wireless indoor positioning systems in order to provide a better understanding of the current research issues in this exciting field. First, in Section 2.1, a broad classification of the current indoor positioning systems is provided with some related examples for each. The description of some localization algorithms which are fundamental parts for accurate location estimation together with the examples of positioning systems that utilize them, appears in Section 2.2.

2.1 Taxonomy of Indoor Positioning Systems

The current research efforts for indoor positioning systems can largely be divided into two main categories:

- Those that make use of angle of arrival (AoA), time of arrival (ToA), and time difference of arrival (TDoA) methodologies. This family of localization techniques relies on specialized hardware (e.g., RF tags, ultrasound or infrared receivers, etc.) and extensive deployment of dedicated infrastructure solely for localization purpose [12–14, 31].

- Those that utilize the correlation between easily measurable signal characteris-

tics (e.g., RSS) and location. These location fingerprinting solutions try to build a positioning system on top of existing infrastructure (e.g., Wi-Fi or Bluetooth networks) [17, 18, 20, 32] in a cost-effective way.

Comprehensive surveys of the infrastructure-based positioning systems (i.e., the first category above) can be found in [5, 6]. Therefore, rather than delving into minute details of each of the forerunners of these types of systems, a subset of them is reviewed as examples in the following:

- Active Badge [12] is one of the pioneers for infrastructure-based indoor positioning systems. In this system, a small infrared (IR) badge is worn by each personnel to be tracked which emits a globally unique identifier every ten seconds. The network of sensors placed around the building detects it and reports to the location server. By inspecting which badge is seen by which room's sensor, it is possible to determine the location of a particular badge's owner. Since light is blocked by walls, IR location system has a relatively high room-level accuracy.

- Active Bat [13] improves over the room-level accuracy provided by Active Badge by using both RF and ultra-sound technologies. An array of ceiling-mounted ultra-sound receivers is deployed where the receivers are connected to the centralized positioning server via a wired network. The centralized controller sends out an RF request packet for the mobile "Bats", and simultaneously, sends a reset signal to the ceiling-mounted receivers. The receivers calculate the distance measurement starting from the time they receive the reset signal to the time they receive ultra-sound response pulse from the mobile "Bat", and computes the Bat's position by using multilateration (the localization algorithms are discussed in the next section). The system is shown to have 2 cm average accuracy.

- PinPoint's 3D-iD positioning system [33] is an indoor RF-based commercial product. A tag's location is determined by continuously broadcasting a signal

from an array of antennas at known cells' positions. When a tag receives a signal, it will immediately retransmit the message by shifting it to another radio frequency and encoding it with its own ID. The system controller measures multiple distances from the array of antennas using RF round-trip time and performs multilateration to estimate the location. The system has a 30 m range and offers 1 m to 3 m accuracy. It requires several transmitter cells per building and has expensive hardware.

- Ubisense [15] offers commercial solutions for location identification and tracking using UWB technologies. UWB has good multi-path resolution characteristics and obstacle penetration capability inside a room, compared to the other existing transmission media (e.g., IR or ultra-sound). Ubisense UWB positioning system requires fixed sensor infrastructure (i.e., networked units placed around the building) together with the tags carried by people or attached to the objects to be tracked. It measures both AoA and TDoA information of the tag's signals, enabling it to generate accurate 3D tracking information even when only two sensors can detect the tag. It is argued to offer accuracy in the range of 15 cm in 3D.

The main drawback of infrastructure-based positioning systems is the cost of infrastructure installation and the custom sensor badges or tags, which becomes significant for a large building with a lot personnel/objects to be located. Moreover, there are some technology specific shortcomings, e.g., the infrared or ultra-sound sensing signals cannot penetrate the walls and floors which are common inside most buildings.

The second category of the positioning systems which are overlaid on top of any existing wireless infrastructure (e.g., Wi-Fi, Bluetooth, etc.) can save the cost of dedicated infrastructure. Moreover, it utilizes RF signals which penetrate most of the indoor materials resulting in a larger range. The most common location fingerprint

RSS can be measured by the off-the-shelf hardware (e.g., wireless NIC). Therefore, Laptops, PDAs, and other handhelds with built-in RF support (e.g., Wi-Fi or Bluetooth) can be provided with location information without the need of any custom tag or badge. A subset of the forerunners of such indoor positioning systems is discussed as examples in the following:

- Place Lab [34] is a radio beacon-based approach to location, that can overcome the lack of ubiquity and high-cost found in the infrastructure-based location sensing approaches. The Place Lab approach is to allow commodity hardware clients like laptops, PDAs and cell phones to locate themselves by listening for radio beacons such as Wi-Fi APs, GSM cell phone towers, and fixed Bluetooth devices that already exist in the environment. These beacons all have unique or semi-unique IDs, e.g., a MAC address. Clients compute their own location by hearing one or more IDs, looking up the associated beacons' positions in a locally cached map, and estimating their own position referenced to the beacons' positions. Place Lab has a critical dependence on the availability of beacon locations; if Place Lab knows nothing about a beacon, being in range does not improve the location estimates. The *beacon database* plays an important role of serving this beacon location information to client devices. Many of these beacon databases come from institutions that own a large number of wireless networking beacons. Other sources of Place Lab mapping data are the large databases produced by the war-driving community [35]. Their list of beacon database can be found in [36].

- Location fingerprinting which was discussed in Section 1.2 became popular with RADAR [17] mainly because of the unavailability of appropriate radio signal propagation models for indoor environments. It also opened the door for many different approaches to be applied for indoor localization problem. RADAR

ties the average RSSs observed from the APs to a particular location which is termed as their location fingerprint. It found the user orientation and human being's movement inside the building to create random fluctuations of radio signals inside the building. Some other factors, e.g., temperature, air movement, and interference from other devices operating in the same frequency, were also seen to cause the RSS to vary at a particular location over time [37]. RADAR uses simplistic pattern matching algorithm (e.g., $\mathcal{K}$-Nearest Neighbor) to find the ultimate location estimate. Details of $\mathcal{K}$-Nearest Neighbor ($\mathcal{K}$NN) for location estimation are discussed in Section 2.2.5.

- Nibble [18] is one of the first systems to use a probabilistic approach for location estimation. Instead of being a deterministic constant value of average RSS vector, the location fingerprint becomes a conditional probability distribution of the observation vector of RSS and the location information. These distributions of the location fingerprints are either maintained via histogram [9, 18, 20, 29] or parametric estimation (e.g., normal distribution) [26, 27, 30]. With this form of location fingerprint, the Bayes' rule can be used to estimate the location. Details of Bayesian algorithms for location estimation are discussed in Section 2.2.6.

- Ekahau [22] is a commercial product which provides positioning support for Wi-Fi only. In addition to their custom Wi-Fi tags, they also support a few off-the-shelf NICs. To date, Ekahau's positioning engine software claims to be the most accurate location system based on probabilistic model of location fingerprinting techniques; they claim a one-meter average accuracy with a short offline training period [22].

- Skyhook [38] provides XPS, a hybrid positioning system, taking advantage of the relative strengths of several location technologies, e.g., Wi-Fi Positioning System (WPS), GPS, cellular tower triangulation. XPS is a software-only lo-

cation platform that can quickly determine the location of any Wi-Fi enabled MN with an accuracy of 10 to 20 m. The MN running an XPS client collects raw location data from the Wi-Fi APs, cellular towers and GPS satellites that continuously broadcast signals. This information is then sent to the XPS server which subsequently estimates the MN's location and returns the location information back to it. Skyhook's Wi-Fi and cellular database is arguably the largest and most extensive in the world. They claim to have scanned every single street in major metro areas worldwide, collecting Wi-Fi APs and cellular tower IDs. Skyhook's strength lies in the fact that they target to provide location services to a user in both indoor and outdoor scenarios using multiple technologies (e.g., GPS, Wi-Fi, etc.).

2.2 Localization Algorithms

In this section, the localization algorithms which form the core all the localization schemes classified above are elaborately discussed. Though some previous works [7, 39, 40] roughly touches upon the various localization or positioning techniques, they do not relate them to the existing protocols. Hightower and Borriello [6] provide a taxonomy of existing positioning systems and try to compare them regarding various performance metrics pertaining to any positioning system. Since location fingerprinting literature was not matured at that point, only RADAR [17] of that genre could be found in their survey. This section elaborately discusses the positioning methodologies, and also shows how the existing localization schemes (including various location fingerprinting solutions) adopt them.

2.2.1 Triangulation

Triangulation is one of the basic location estimation methods used in this field. From GPS to even recent schemes [14, 17, 41–43] make use of simple geometric properties to infer location estimates. Triangulation, in its simplest form, uses the distances (lateration) or angles (angulation) from some reference points to correctly infer the node's current location. Basically, three approaches to calculate the "spatial separation" between a reference point and the concerned node are popular in the literature: i) Time of Arrival (ToA) / Time Difference of Arrival (TDoA), ii) Angle of Arrival (AoA), and iii) Propagation Models. They are all elaborately discussed in the following.

2.2.1.1 A. Time of Arrival (ToA) / Time Difference of Arrival (TDoA)

In localization literature, both ToA and TDoA are used synonymously, though there is a subtle difference between them. ToA denotes the time elapsed for a signal to travel from/to a reference point to/from the node. It requires the node's clock to be synchronized with that of the reference point in order to infer exact "time of flight" of the signal. On the contrary, TDoA works by measuring differences in arrival times of a signal from a node at different reference points.

ToA is used in GPS technology to deduce the distances from GPS satellites. In order to measure the "time of flight" of the signals from satellites, the receiver clock has to be synchronized with satellite clocks. Practically, it is difficult to achieve, therefore, the receiver clock attributes a bias to the distance estimate from each satellite. Since all GPS satellite clocks are synchronized themselves, the receiver bias is the same for all satellite clocks. Consequently, if (x, y, z) is the receiver's coordinate and (x_k, y_k, z_k) denotes the k^{th} satellite's coordinate, the distance estimate from the k^{th} satellite can be written as,

$$d_k - b = \sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2} \tag{2.1}$$

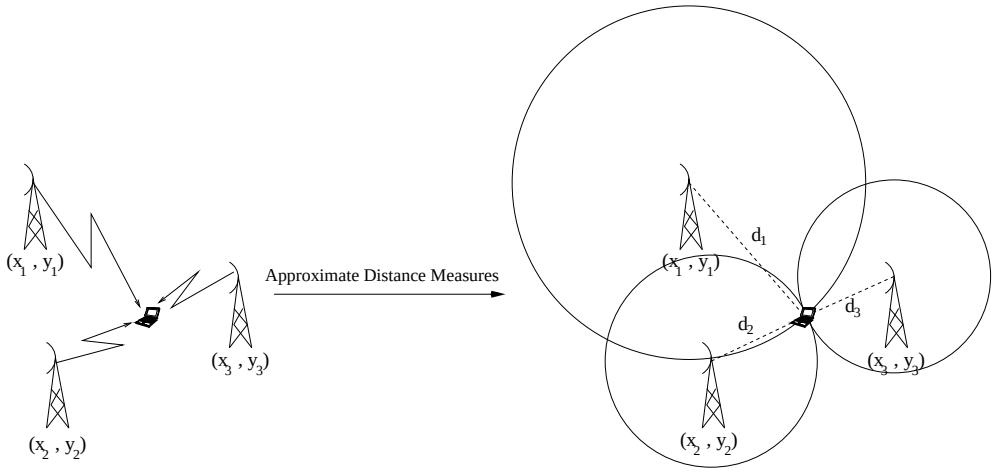


Figure 2.1: Location estimate in 2D for ideal case using lateration.

Here, b is the receiver bias component which is the same for each satellite. ToA estimates are always greater (never smaller) than true ToA values because of multi-path and other impairments. So the bias b is actually subtracted from the calculated distance estimate d_k in (2.1). There are four unknowns (i.e. $x, y, z,$ and b) in (2.1), therefore a receiver requires at least four satellites in view to infer its location (x, y, z).

Fig. 2.1 shows the most common way to infer a node's location once the distance approximations are made. Considering the k^{th} reference point as center, we get a system of circle equations of the following form,

$$(x - x_k)^2 + (y - y_k)^2 = d_k^2, \quad k = 1, 2, 3. \tag{2.2}$$

Subtracting the circle equations from each other, we get two quadratic equations with only two unknowns (i.e. $x,\ y$) and solving them to get the final location estimate is trivial.

$$M \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} d_1^2 - d_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ d_2^2 - d_3^2 + x_3^2 - x_2^2 + y_3^2 - y_2^2 \end{bmatrix},$$

where matrix M is denoted by,

$$M = \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_2) & 2(y_3 - y_2) \end{bmatrix}.$$

Note that, (2.2) is similar to (2.1) in 2D apart from the receiver bias.

In TDoA approach, differences of ToAs are used rather than absolute time measurement. Since the measured difference of distances traveled by the signal from two reference points is constant for a node, the locus of it can be translated into a hyperbola with the reference points at the foci.

$$d_{kl} = d_k - d_l = v \times (T_k - T_l), \text{ and}$$

$$d_{kl} = \sqrt{(x - x_k)^2 + (y - y_k)^2} - \sqrt{(x - x_l)^2 + (y - y_l)^2}. \tag{2.3}$$

where v is the signal's speed and $(T_k - T_l)$ denotes the time difference of the signal's arrival between reference points k and l. Equation (2.3) gives the locus of a node with foci at reference points k and l. The intersection of such hyperbolas with two or more pairs of reference points provides the estimated location of the node [44].

Cricket [14] is a different example of TDoA discussed above. Cricket positioning system works by measuring the time difference of arrival between RF and ultrasound pulses at the receiver sent concurrently from a *beacon* (i.e., reference point). The RF pulse basically works as a synchronizing signal between the *beacon* and the receiver in Cricket. Sound pulses travel 343.4 m/s in 20^oC air, whereas, light pulses have velocity $299,792,458$ m/s [45]. When a Cricket receiver receives the first bit of an RF pulse sent from a *beacon*, it starts calculating the time until it receives the ultrasound pulse from the same *beacon*. Suppose, our node is 5 meters away from a *beacon*. Then, theoretically, the node would receive RF and ultrasound pulses from it after 17 and

14560280 nanoseconds, respectively. So, in this case, the theoretical distance estimation of Cricket would be, $343.4 \times (14560280 - 17) \times 10^{-9}$ or 4.9999942 meters, which is equal to the actual separation between the *beacon* and the node.

"Time of flight" measurement is the most accurate compared to the other distance estimation methods, although, there are challenges in separating the main signal's arrival time from its reflections [13, 14].

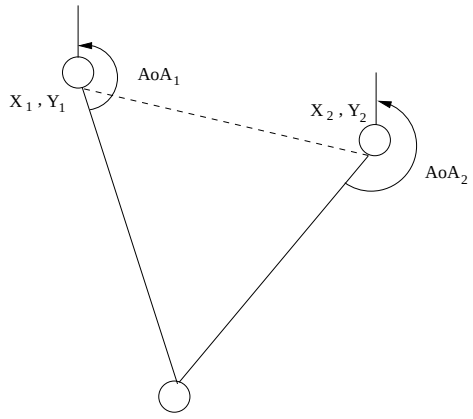
2.2.1.2 Angle of Arrival (AoA)

Based on the properties of some receiving antennae (e.g., phased antenna array), the originating signal's angle can be inferred. Solving linear equations of the form, $y - y_k = m_k \times (x - x_k)$ where $k \geq 2$, gives the ultimate estimate of the node's position (x, y) in 2D [46]. Here, (x_k, y_k) denotes the coordinates of the k^{th} reference point, and m_k specifies the slope of the line joining the node and the k^{th} reference point which is deducible from the arrival angle of the emitted signal (Fig. 2.2(a)). Note that, angles from only two reference points ($k = 2$) are enough to solve the linear equations in order to find a unique location estimate.

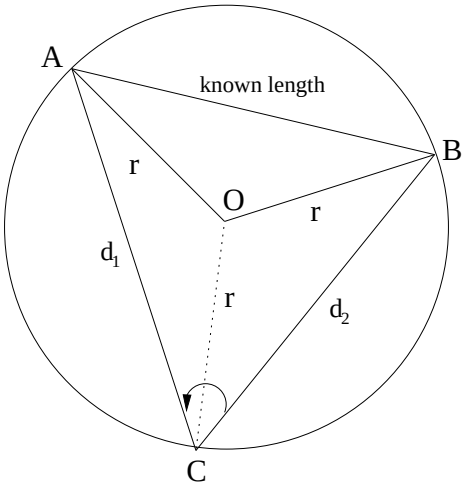
Fig. 2.2(b) helps to geometrically derive the location estimate quantity for the same scenario where it is actually converted into a lateration problem. From the angle of arrival information, the angle at point C of Fig. 2.2(b) could be comprehended, i.e., $\angle ACB = \angle AoA_2 - \angle AoA_1$. We know from circle properties that, the angle subtended by AB at its center is twice the angle subtended by it at point C, i.e., $\angle AOB = 2\angle ACB$. Now, applying cosine law in $\triangle OAB$, radius r of the circumscribing circle can be obtained as,

$$\cos \angle AOB = \frac{r^2 + r^2 - AB^2}{2r^2}$$

The other two angles $\angle ABC \ \& \ \angle CAB$ can be comprehended with the additional



(a) The coordinates of the reference points {i.e., (X_1, Y_1) & (X_2, Y_2)} are known – so are the emitted signals' angles from them (i.e., $\angle AoA_1$ & $\angle AoA_2$).



(b) Equivalent figure of 2.2(a): A & B denote the positions of the reference points and C denotes the node's position – $\angle ACB$ can be inferred from $\angle AoA_1$ & $\angle AoA_2$.

Figure 2.2: Location estimate using angle information in 2D (the originating signals' angles are represented w.r.t. magnetic north).

information of the *bearings* [47] of the reference points to each other. Since the coordinates of the reference points are known, these *bearings* are not hard to calculate. Then similar application of the circle property and cosine law for $\triangle OAC$ and $\triangle OBC$ respectively, yield the distance measurements d_1 and d_2 from the two reference points. To unambiguously infer a node's location, distance estimates from three or more reference points are usually required as previously explained.

2.2.1.3 Propagation Models

The emitted radio signal strength from the reference point decreases with distance. Based on various propagation models [16], we can deduce the received signal at a given distance. For example, considering free-space propagation model, a radio signal attenuates by $1/d^2$ when it reaches a node at a distance, d. So, if we know the transmitted power of the original radio signal, we could find the received signal strength using the path-loss equation of the free-space propagation model [16]. Conversely, if

we can measure the received signal strength at a node without knowledge about its distance from the source, we may subsequently infer the distance by making use of the same model. Finding the appropriate propagation model is a challenge, especially in indoor environments, because, RF signal suffers from multi-path effect, refraction, and reflection from objects with different properties which cause the attenuation of the signal to correlate poorly with distance. To combat this phenomenon, some works try to derive propagation models pertaining only to a specific indoor environment. For example, SpotON's [41] indoor propagation model is entirely based on empirical data. Nonetheless, RADAR [17] came up with Wall Attenuation Factor (WAF) model based on the number of obstacles (e.g., walls) separating the transmitter and receiver. They approximated the value of WAF parameter by conducting experiments measuring signal strength between transmitter and receiver when they had line-of-sight and also, while they were separated by walls. Unfortunately, RADAR's propagation model did not perform as accurate as their empirical method.

Apart from these three basic techniques to deduce the distance between a reference point and the node to be located, other approaches also exist. For example, DV-Hop [42], Amorphous [43] and Self-Configurable [48] localization are proposed mainly for ad-hoc networks to provide coarse-level granularity, and they use number of hops to reach a node as an indication of its distance away from the reference points.

2.2.2 Proximity to a Reference Point

The family of coarse-grained localization schemes try to estimate locations of the nodes on a broader scale. Instead of trying to make near-perfect estimate of distance from a reference point, these schemes may infer the node to be collocated with a reference point, if the node hears beacons from it. In general, coarse-grained localization schemes try to measure a node's closeness to a reference point of known position.

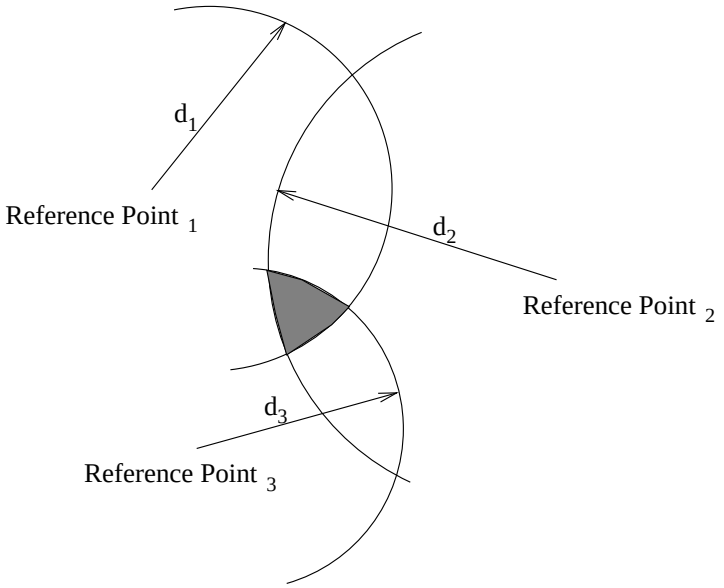


Figure 2.3: The node's estimated position resides inside the shaded region rather than yielding a unique intersection point.

When more than one reference point reports the node to be in their vicinity, some simple computationally inexpensive techniques (e.g. Centroid [49], APIT [50]) are utilized to infer the location:

- The Centroid scheme [49] defines a *connectivity* metric which indicates the closeness of a node to a particular reference point. During a certain time interval, all the reference points send a predefined number of beacons. The *connectivity* metric is defined as the number of beacons received by the node from a particular reference point to the number of beacons sent by it during a time interval. The final location estimate is the centroid of all the reference points for which, the *connectivity* metric is above a certain threshold.

- Approximate point-in-triangulation or APIT [50] takes the Centroid scheme a step further and gives center of gravity of the overlapping areas created by triangles (triangle vertices are reference points) as the node's ultimate position. Only those triangles where the node is supposed to be inside are considered.

Though APIT tries to improve on the overall localization error, it suffers from InToOutError (i.e. the node is inside a triangle but the APIT test shows otherwise) and OutToInError (i.e. the node mistakenly assumes to be inside a triangle) which affects its performance.

In short, these techniques incur less complexity in both the nodes and the infrastructure accommodating them, at the expense of larger localization error. Sometimes, a node may not be detectable by three or more reference points or the reporting stations may be collinear. So the fine-grained distance approximation methods (e.g., triangulation) may not apply. In these cases, the systems using proximity techniques can at least provide some coarser approximations. For example, Cricket [14] receiver basically uses lateration to infer its position. It requires the receiver to hear announcements from four *beacons* or reference points (not three) to correctly deduce its position. Speed of sound comprises the fourth unknown there, as it varies with temperature, humidity, etc. [51]. Once the receiver fails to receive announcements from four *beacons*, Cricket reverts back to proximity measures and gives the centroid of the receiving *beacons'* coordinates as its own position.

2.2.3 Gradient Descent Method

Sometimes geometric interpretation to calculate intersection of circles as discussed in 2.2.1 does not provide a unique solution (see Fig. 2.3) [3]. This may result due to the distance approximation errors incurred while using ToA/TDoA, AoA or propagation models. A more robust algorithm like the gradient descent approach, can eliminate this shortcoming. From Eq. (2.2), the performance measurement function considering the k^{th} reference point can be obtained as,

$$f_k(X) \quad = \quad d_k - \sqrt{(x - x_k)^2 + (y - y_k)^2}$$

$$= \quad c(\tau_k - \tau) - \sqrt{(x_k - x)^2 + (y_k - y)^2}$$

where c is the speed of light, and the node's transmitted sequence at time τ is received by the k^{th} reference point at time τ_k. There can be many types of objective functions, but, for simplicity, let us consider the following objective function to be minimized [4],

$$F(X) \quad = \quad \sum_{k=1}^{K} f_k^2(X)$$

where K is the number of reference points and $X \ = (x, y, \tau)^T$. Successive location estimates are updated according to the following formula,

$$X_{i+1} \ = \ X_i - \eta \bigtriangledown_X F(X_i) \tag{2.4}$$

where η is a small constant, used to maintain stability in search for optimal X by ensuring that, the operating point does not move too far along the performance surface. X_i specifies the i^{th} estimate and $\bigtriangledown_X F(X_i)$ denotes the gradient of the performance surface at i^{th} iteration which is defined as,

$$\bigtriangledown_X F(X_i) \ \equiv \ \bigtriangledown_X F(X)|_{X_i} \ = \ \begin{pmatrix} \dfrac{\partial F}{\partial x} \\ \dfrac{\partial F}{\partial y} \\ \dfrac{\partial F}{\partial \tau} \end{pmatrix} |_{X_i}$$

$$= \begin{pmatrix} 2\displaystyle\sum_{k=1}^{K} \dfrac{f_k(X_i) \times (x_k - x_i)}{\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}} \\ 2\displaystyle\sum_{k=1}^{K} \dfrac{f_k(X_i) \times (y_k - y_i)}{\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}} \\ -2c\displaystyle\sum_{k=1}^{K} f_k(X_i) \end{pmatrix}$$

The recursion in (2.4) continues until $\|\eta \bigtriangledown_X F(X_i)\| \leq \epsilon$, where ϵ is a predefined maximum permissible error.

2.2.4 Smallest Vertex Polygon

Smallest Vertex Polygon (SVP) [21,52] is a simple algorithm to infer location estimate for fingerprint based positioning systems. During a runtime signal measurement, if a number of locations w.r.t. a reference point's offline training database seem likely according to the *bracketing* heuristic [52], then all such locations constitute the candidate set regarding that particular reference point. Subsequently, a number of distinct vertex polygons are formed where each vertex is from a different reference point's candidate set. Suppose, the search for candidate set results in M potential locations for each of the K reference points. So, a total of M^K distinct vertex polygons are realized. Among them, SVP is the one having shortest perimeter and its centroid denotes the final location estimate. The idea behind such an algorithm was to allow a fair contribution from all the reference points.

2.2.5 Nearest Neighbor in Signal Space

Nearest Neighbor (NN) algorithm is first utilized in RADAR [17] to tackle the localization problem, and subsequently being used by other works relying on signal pattern matching techniques ([21], [52], [27], etc.). This algorithm returns the location entry from the location fingerprint database which has the smallest root mean square error in signal space with the given runtime measurement at the unknown location. $\mathcal{K}$-nearest neighbor ($\mathcal{K}$-NN) is a variant of the basic nearest neighbor algorithm where $\mathcal{K}$ location entries are searched instead of returning only the best match. The final location estimate is obtained by averaging the coordinates of the $\mathcal{K}$ locations found. The value of $\mathcal{K}$ has usually been chosen empirically in the literature. RADAR's experimental results

show that, $\mathcal{K}$-averaging has some benefit over the basic nearest neighbor algorithm for smaller $\mathcal{K}$'s, but for large $\mathcal{K}$, their accuracy degrades rapidly as points irrelevant to the true location are also included in the averaging.

2.2.6 Probabilistic Methods

The probabilistic approach models the location fingerprint with conditional probabilities and utilizes the Bayesian inference concept to estimate location [18, 20, 22, 26, 27, 53]. It does not follow the deterministic approach to represent the location fingerprints as a vector of mean RSSs like the nearest neighbor algorithm discussed above. Consequently, the location fingerprint becomes a conditional probability distribution of the form $Pr(O|L)$ where O denotes the observation vector of RSS at location L. For each location, $l \in L$, we can estimate the likelihood function $Pr(O|L_l)$ from an offline training set consisting of samples of location fingerprints observed at that position. In localization literature, there are generally two methods for representing the likelihood function: i) the parametric approximation and ii) the histogram approach.

- Roos *et al.* [53] suggested a kernel method to approximate the probability density function of the RSS from an AP at a particular location. However, the most popular parametric estimation is the Gaussian model as can be seen from many existing works (e.g., [26, 27, 30]):

$$Pr(\mathsf{O}_k|L_l) = \frac{1}{\sigma_{kl}\sqrt{2\pi}} \exp\{(x - u_{kl})^2/2\sigma_{kl}^2\},$$

  where μ_{kl} and σ_{kl} denote the mean and standard deviation of RSS from the k^{th} AP at location $l \in L$. These parameters can be obtained from the offline radio-map database. The rationale behind choosing such Gaussian model approximation is usually vindicated through experimental findings [26, 27, 30].

- The histogram representation [9, 18, 20] is essentially a fixed set of bins where each bin holds the frequency of occurrence of RSS samples that falls within the range of that particular bin. The bin's range is calculated from an adjustable number of bins and the known values of minimum and maximum RSS values. The larger the number of bins, the better the histogram can approximate the probability density function of RSS.

A slightly more sophisticated way to determine $Pr(O|L)$ is presented in [9] where two different conditional probabilities are calculated from two different histogram representations and are multiplied together. The first conditional probability represents the frequency count of a particular access point's collected samples given a location L. In other words, this probability indicates how often the system visualizes the particular access point at that location. The second conditional probability represents the distribution of RSS from that access point given the same location.

According to Bayes rule, a posterior distribution of each location $l \in L$ can be formed as the following,

$$Pr(L_l|O) = \frac{Pr(O|L_l)P(L_l)}{Pr(O)} = \frac{Pr(O|L_l)Pr(L_l)}{\sum_{m=1}^{|L|} Pr(O|L_m)Pr(L_m)}, \tag{2.5}$$

where $|L|$ is the total number of discrete locations and $Pr(L_l)$ denotes the prior probability of being at location L_l which can be set as a uniform distribution, assuming every location is equally likely. As the denominator $\sum_{m=1}^{|L|} Pr(O|L_m)Pr(L_m)$ does not depend upon the location variable l, it can be safely treated as a normalizing constant whenever only relative probabilities or probability ratios are required. Upon observing a particular fingerprint (e.g., O^*), the position (x, y) of the MN can subsequently be calculated as, $x = \sum_{l=1}^{|L|} x_l \cdot Pr(L_l|O^*)$ and $y = \sum_{l=1}^{|L|} y_l \cdot Pr(L_l|O^*)$.

In general, upon observing a particular fingerprint (e.g., O^*), Bayes decision rule will select position L_m over position L_n using (2.5) when:

$$\begin{aligned} Pr(L_m|O^*) &> Pr(L_n|O^*) \\ \Rightarrow Pr(O^*|L_m)P(L_m) &> Pr(O^*|L_n)P(L_n). \end{aligned} \tag{2.6}$$

Upon the assumption of uniform distribution of $Pr(L_l)$'s, (2.6) can further be simplified as $Pr(O^*|L_m) > Pr(O^*|L_n)$ which comprises of the likelihood conditional probabilities only. Maximum Likelihood Estimate (MLE) chooses the location L_l^* which obtains the maximum value of the posterior probability:

$$L_l^* = \arg\max_{L_l} Pr(L_l|O^*).$$

Since probabilistic models incorporate additional information compared to the deterministic representation of location distribution, they are expected to provide better performance on location estimation. However, location systems utilizing probabilistic models usually require a large training set in order to map the conditional probabilities appropriately.

2.2.7 Neural Networks

Neural Network methods were introduced in localization problem with the view that RSS fingerprints are too complex to be analyzed mathematically and may require subtle non-linear discriminant functions for classification. Battiti *et al.* [24] utilized multilayer perceptron (MLP) neural network [54] in solving the indoor localization problem. They noticed slow convergence period of their neural network during the offline training phase and also emphasized on the need for a large training set to train the neural network properly. The problems of over-training or over-fitting also accom-

pany these types of classification algorithms [54]. Since the neural networks avoid characterization of the statistics of the location fingerprints, they rarely provide any insight information on the underlying mechanism of indoor positioning system. The performance of such algorithms is found to be almost similar to the simplistic pattern matching techniques (e.g., $\mathcal{K}$-NN) in localization literature [24].

2.2.8 Support Vector Machines

The support vector machines (SVMs) algorithm has its root in statistical learning theory introduced by Vapnik [55]. The strength of SVMs lies in its ability to be trained correctly through a relatively small training set and creating sufficient structure for data classification without memorizing or over fitting the training samples [23].

Here, the SVMs algorithm is described briefly without divulging into mathematical details. Avid readers may consult [56] for an excellent mathematical description of the technique. In order to apply SVMs into the localization problem, first, the vectors of location fingerprints are mapped into a higher dimensional space called feature space by using a *kernel* function to perform the vector transformation [23]. Subsequently, the SVMs algorithm creates an optimal separating hyperplane or decision surface in that feature space and uses the hyperplane to perform classification. The separating hyperplane is not unique in general, and is optimal when it has the largest possible distance from the closest training point or a maximal margin. A hyperplane is any codimension-1 vector subspace of a vector space, or equivalently, a hyperplane V in a vector space W is any subspace such that W/V is one-dimensional. However, the application of SVMs did not improve the localization performance compared to other techniques (e.g., $\mathcal{K}$-NN) [23]. Also, from the theoretical modeling perspective, the SVMs may be too complex to provide useful information into designing a positioning system.

2.3 Summary and Conclusions

Indoor localization techniques that are built on top of already existing network infrastructure have gained popularity because of their ubiquity and cost-effectiveness compared to other infrastructure-based positioning systems. These systems generally use location fingerprinting techniques that utilize the correlation between a location and its signature (e.g., RSS), and apply mostly pattern matching techniques. The unavailability of appropriate radio propagation model inside the building made them unable to apply the localization algorithms (e.g., lateration) commonly seen in the infrastructure-based systems. Consequently, the reported accuracy of the fingerprinting techniques falls short of their counterparts. However, finer accuracy is only one of the many characteristics of a positioning system that are expected to be met. It is generally agreed that a desirable indoor positioning system should be characterized by good accuracy, short training phase, cost-effectiveness (preferably using off-the-shelf hardware), and robustness, in the face of previously unobserved conditions. This work herein aims to achieve a positioning system that accomplishes all these requirements.

Since one of our goal is to design a *cost-effective* positioning system, we opt for fingerprinting techniques that utilize the existing network infrastructure, and use off-the-shelf hardware (e.g., laptops, PDAs, phones, etc.) carried by the users that have built-in wireless technologies. Due to the widespread availability of Wi-Fi and Bluetooth in such devices, we select these two as our underlying technologies to provide positioning service. The majority of the fingerprinting techniques in the literature that are based on Wi-Fi, make use of the basic RF-based signal strength parameter (i.e., RSS) as a location fingerprint; although other signal strength parameters (e.g., Signal-to-noise ratio (SNR)) are also available. A comparative study of such RF-based signal strength parameters of Wi-Fi is required from a positioning system's point of view. Only a few works in the localization literature have divulged into Bluetooth based posi-

tioning systems. There is clearly a need for a comprehensive study of Bluetooth signal strength parameters from a positioning system's perspective as well. The shortcoming of the popular RSS location fingerprint has been apparent in existing literature [25–29] across different hardware solutions for the same wireless technology. This limits RSS's usage as location fingerprint if the positioning system aims to accommodate heterogeneous hardware solutions (e.g., laptops, PDAs, phones, etc.). This study addresses this issue and subsequently, proposes a *new* location fingerprint which is shown to be *robust* across different hardware solutions. This dissertation also proposes two different approaches in order to *shorten* the exhaustive training phase typically seen in the fingerprint based positioning systems.

The major performance metric usually seen in the localization literature is the positioning accuracy which is a form of error measurement. *Accuracy* of the location information is reported as an error distance between the estimated location and the MN's actual location. However, some works also report the percentage of successful location detection within a particular accuracy which is termed as *precision*. In short, the term "accuracy" generally indicates the grain size of the location information provided, while the term "precision" specifies how often we are expected to attain that accuracy [6]. For example, if a location system can determine positions within 3 meters for about 90 percent of the measurements, that particular system qualifies to be 90% precise in providing 3-meter accuracy. The positioning systems which deploy dedicated infrastructure just for positioning services, are able to provide centimeter-level accuracy. For example, the accuracy and precision of Active Bat positioning system is quite impressive at 9 cm for 95% of locations [13]. Depending on the testbed environment and how the offline training phase is conducted, the accuracies offered by the systems which are overlaid on top of existing infrastructure are noticed to have varied slightly. An average accuracy in the range of 2 to 3 m is generally reported in most localization literature for such systems [20, 24, 37, 53]. To date, Ekahau's po-

sitioning engine software [22] claims to be the most accurate location system based on probabilistic model for location fingerprinting techniques; they claim a one-meter average accuracy with a short training period. However, Ekahau's claim of one-meter average accuracy draw mixed reactions from the research community who performed tests with their software (e.g., [21]).

Our research limits the scope of the localization algorithms to two well-known techniques, namely, the nearest neighbor and Bayesian inference. This is in accordance with the view that our ideas are quite generic and can be productive irrespective of the choice of algorithms. Moreover, the other complex algorithms, e.g., neural networks and support vector machines, do not provide favorable results compared to the simplistic pattern matching algorithms (e.g., $\mathcal{K}$-NN) [23]. Both neural networks and SVMs are non-parametric classifiers [57] that do not assume any knowledge of the distributions of the location fingerprints. As a result, they basically cannot provide insight on how to improve the positioning systems beyond the complexity of their generic learning machines.

Chapter 3

Review of Location Fingerprints

In this chapter, we investigate the properties of the signal strength based parameters for two well-known RF technologies, namely, Wi-Fi and Bluetooth. This analysis is required to understand the underlying features of location dependent signal strength parameters' patterns. As we will explain, the available RF signal strength parameters of Wi-Fi and Bluetooth have specific usage according to their own respective technologies, which may render them inappropriate for consideration as location fingerprints. For Wi-Fi, there has been extensive study regarding its signal strength parameters (e.g., RSS). However, such knowledge is generally aimed towards communications capability [58, 59] or receiver design [60]. An understanding of the signal strength parameters for location fingerprinting in order to improve the design of positioning systems is still lacking.

A comprehensive study of Bluetooth signal strength parameters from a positioning system's perspective is missing as well. Very few works have actually contemplated a Bluetooth positioning system (e.g., [61, 62]). Positioning systems that are solely based on Bluetooth however reported coarse accuracy mainly because of the choice of an inappropriate signal strength parameter as location fingerprint. To the best of our knowledge, no work has delved into inspecting the intricacies related to

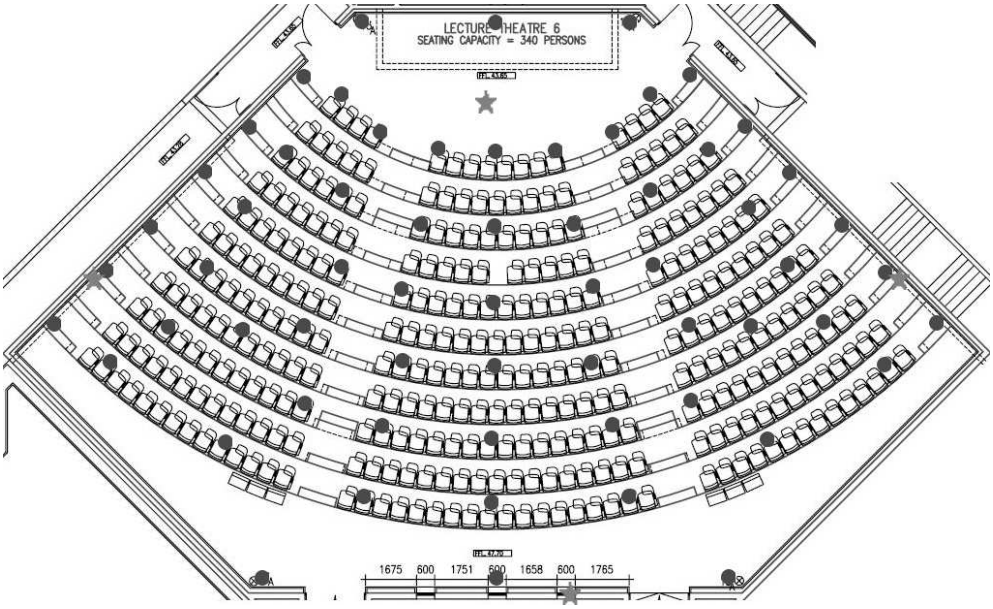


Figure 3.1: Our first experimental testbed – the training locations which we use as training data are marked as shaded circles.

choosing an appropriate location fingerprint in detail for Bluetooth.

Before illustrating the existing RF signal strength parameters' pitfalls regarding localization for both Wi-Fi and Bluetooth, we first describe our experimental setup. We have three experimental testbeds where one is located inside a lecture theater, and the other two are set up within two different research laboratories of our university campus. The rationale behind choosing a few testbeds was to emulate different practical scenarios. Next, we elaborately describe all our measurement setups where the experiments are conducted.

3.1 Measurement Setup

As mentioned previously, we have three different experimental testbeds:

- The first is a Wi-Fi and Bluetooth testbed located inside a lecture theater of our university campus which is shown in Fig. 3.1. We shall refer to this testbed as

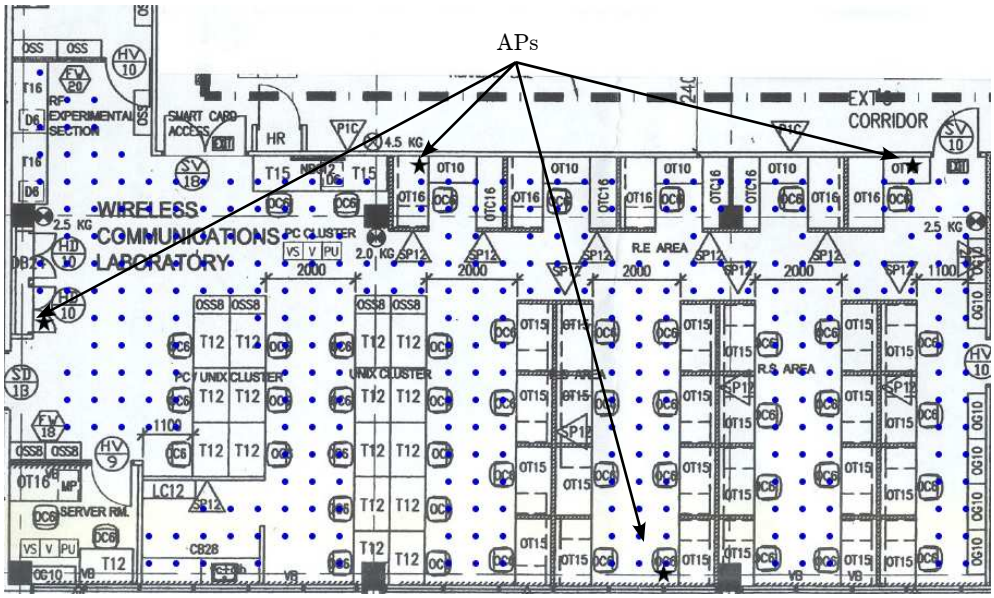


Figure 3.2: Our second experimental testbed – all the training locations are marked as shaded circles.

"Testbed 1" throughout the thesis. In this testbed, we have used four Aopen MP945 Mini PCs to serve as our APs which are placed near the ceilings. The locations of these APs are marked as stars in Fig. 3.1. Each MP945 is incorporated with Ranger's BT-2100 Class 1 Bluetooth adapter which scans for Bluetooth packets and is also installed with Aopen WN2302A mini-PCI WLAN adapter in order to passively detect Wi-Fi devices. Each Mini PC or AP is connected to our university's intranet for communicating with the server by means of a wired LAN connection. All our mini PCs ran the latest (at the time of experiment) openSUSE Linux distribution with the latest libpcap libraries [63] and BlueZ protocol stack [64].

- The second ("Testbed 2") is solely a Bluetooth testbed located within a research laboratory of our university campus which is shown in Fig. 3.2. Similar to Testbed 1, we have used the Aopen MP945 mini PCs incorporated with Ranger's BT-2100 Class 1 adapter to serve as our APs. The locations of these APs are

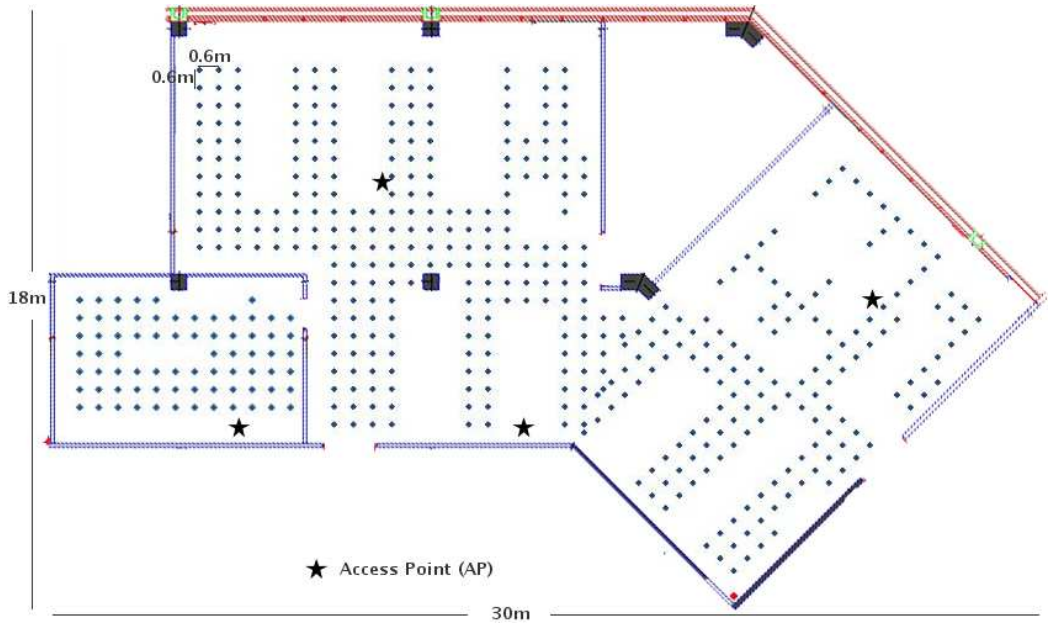


Figure 3.3: Our third experimental testbed – all the training locations are marked as shaded circles.

marked as stars in Fig. 3.2.

- The third ("Testbed 3") is a Wi-Fi testbed which is located inside another research laboratory of our university campus (see Fig. 3.3). Apart from the Cisco APs which provide wireless connectivity in that building, we have also used Linksys WRT54G router and Ekahau T201 tags to serve as our APs. Unlike the previous two testbeds where the signal strengths are actually measured at the AP side, here, MN will be responsible for that operation.

The various Wi-Fi and Bluetooth devices which are used as APs and MNs in our testbeds are listed in Table 3.1. Note that, our first two testbeds emulate the *AP-based* positioning system where the signal strengths are actually measured at the AP side, whereas the third testbed follows an *MN-assisted* approach where the MN itself retrieves the signal strength information.

Table 3.1: The list of Wi-Fi and Bluetooth devices used as MN and AP in our experimental testbeds

	Technology	MN Devices	AP Devices
Testbed 1	Wi-Fi	Intel PRO/Wireless 3945ABG Samsung SWL-2455	Aopen WN2302A
	Bluetooth	Ranger's BT-2100 (Class 1) HP iPAQ 6315 PDA (Class 2)	Ranger's BT-2100
Testbed 2	Bluetooth	Ranger's BT-2100 Billionton's USBBT02-B (Class 2) Acer n300 PDA (Class 2) Motorola V3xx Phone (Class 2)	Ranger's BT-2100
Testbed 3	Wi-Fi	Intel PRO/Wireless 2200BG Atheros AR242x 802.11abg	Linksys WRT54G Cisco Aironet 1200 Ekahau T201 Tag

3.1.1 Data Collection Procedure

In our three testbeds, there are 106, 337 and 466 training points or grids, respectively. The training process involves placing the mobile device at each training point, and collecting data. In the first two testbeds, we adopted the approach whereby our APs are the ones that collect RSS information while in the third, the MN itself retrieves the RSS. Our front-end of the signal strength collection program is a Java Graphical User Interface (GUI) which allows to load the map and click on the location to be trained conveniently.

For the case of Wi-Fi data collection in Testbed 1, the program is written using the libpcap libraries [63] where the WLAN adapter at the mobile device sends probe requests continuously for some period so that the APs can gather enough packets. For the case of Bluetooth, we actually log onto the mini PCs using Secure Shell (SSH) and make the APs issue Bluetooth inquiries which the mobile device responds to. The Bluetooth signal strength information retrieval program is written utilizing the HCI API of BlueZ protocol stack [64]. In either case, the packet information is transferred

to our central server's database from the APs (i.e., mini PCs). These signal strength collection programs are invoked externally from the Java program when we click on the locations to be trained on the map. Note that, our Bluetooth adapters provide the absolute RSS values of the inquiry response packets, rather than the RSSI values as stipulated by the Bluetooth Core specification [65]. In Testbed 1, we have chosen 44 testing points which are completely different from our training locations (not shown in Fig. 3.1). The central server is also responsible for calculating the location estimate during the testing phase. In Testbed 3, we have utilized tcpdump [66] to capture the signal strength information at the MN. We first put the MN's NIC into "monitor mode" and then run tcpdump where it snoops all the 802.11 packets from the air. Later on, we ran some scripting programs on the tcpdump's actual output to retrieve the required RSS information from our desired APs.

Based on our three experimental environments, Table 3.2 summarizes the measurement scenarios used to collect the signal strength data which will be used in subsequent analyses. We noticed that the number of data points collected at each location for Wi-Fi using our own program in Testbed 1 is quite less compared to the tcpdump program utilized in Testbed 3.

3.2 Wi-Fi Location Fingerprints

3.2.1 Received Signal Strength (RSS)

Strictly speaking, RSS is supposed to be the signal power observed at the receiver, and is usually measured in dBm. Consequently, it is expected to vary among different transmitter-receiver pair configurations which will be discussed elaborately in the next chapter. Moreover, the 802.11 network interface cards (NICs) do not provide the RSS readings directly. Instead, a typical NIC only provides the RSSI parameter, in the form

Table 3.2: Experimental design and measurement factors

Factors	Testbed 1	Testbed 2	Testbed 3
Testbed Type	Lecture Theater	Research Lab.	Research Lab.
Testbed Area	540 m^2	214 m^2	382 m^2
Measurement Time	Afternoon/Eve.	Afternoon/Eve.	Afternoon/Eve.
Measurement Span	7 days	10 days	10 days
Measurement Device	APs	APs	MN
Measurement period per location	5 minutes	4 minutes	1 minute
Distance between Locations	Non-uniform 0.6 m or more	Uniform 0.6 m	Uniform 0.6 m
Number of Locations	106	337	466
Software Tool	Java GUI, Libpcap (Wi-Fi), BlueZ (Bluetooth)	Java GUI, BlueZ (Bluetooth)	Java GUI, Tcpdump (Wi-Fi)

of an 8-bit unsigned integer that incorporates quantization error as a result of the A-to-D conversion of the measured signal strength into RSSI. In order to use the RSS as a location fingerprint, which is done traditionally for Wi-Fi based localization, it must be translated from the given RSSI. The 802.11 standard does not mandate how RSSI should be calculated from the sampled RSS. As a result, different vendors tend to have their own formulas or conversion tables for the mapping from RSS to RSSI, and vice versa [59]. In addition, the RSSI is intended for internal use by the NIC, e.g., to determine whether the channel is clear to send, or to decide whether it should attempt to roam. As these operations neither require high precision nor accuracy of the measured RF power, some NIC vendors may choose to provide RSSI readings with limited range, granularity, and accuracy. For example, Cisco has 101 RSSI values and its maximum RSSI is mapped to an RSS of -10 dBm, while Symbol has 32 RSSI values and its maximum RSSI is only mapped to an RSS of -50 dBm. The variations among different transmitter-receiver pairs, together with the non-standardized ways of defining RSSI, have several implications on the localization accuracy. Logically, we

would expect a NIC that can offer a larger set of RSSI values (and hence RSS values) to be able to differentiate among more locations, which could potentially yield better accuracy. We also should not impose a restriction on the devices that can be used with the localization algorithm, especially if widespread usage is desired. Finally, it is also expected that the localization accuracy could deteriorate when the mobile device's NIC is made by a different vendor than the one used during the training phase. The latter undesirable effect is often shielded in existing localization literature, since they typically use the same transmitter-receiver pair during both training and testing phases.

3.2.2 Signal Quality (SQ)

The 802.11 standard describes SQ as "PN (Pseudo Noise) code correlation strength" which gives a measure of channel condition based on the bit error rate (BER) observed at the receiver. Similar to RSSI, SQ measurements are most likely to be inconsistent among different vendors since they may be implemented according to the vendors' own guidelines (e.g., how many "wrong" bit positions may qualify for 50% SQ is likely to be inconsistent among different vendors). Moreover, the SQ parameter's definition is unclear in the 802.11 standard for non-DSSS modulation types (e.g., OFDM used in 802.11g) since only DSSS uses a PN code. All these factors contribute to the omission of SQ as location fingerprint, and till date, no work has actually implemented their positioning system based on it.

3.2.3 Signal-to-noise ratio (SNR)

SNR is generally defined as the ratio of the received signal strength to the power of the ambient RF energy in Wi-Fi nomenclature. Since many Wi-Fi cards do not report SNR [59], its use as a location fingerprint is less common. It is also expected to be a less stable location fingerprint compared to RSS, since it is calculated from RSS

in the first place, with additional uncertainty contributed by the ambient interference which tends to be more random. Prior works have also vindicated this claim. For example, RADAR [17] found more fluctuations in SNR measurements compared to RSS at the same location for a particular NIC. Nibble [18], which uses SNR as a location fingerprint, also reported poorer accuracy compared to its RSS counterparts.

3.3 Bluetooth Location Fingerprints

3.3.1 Received Signal Strength Indicator (RSSI)

In the Bluetooth standard, the RSSI is an 8-bit signed integer that denotes whether the received power level is within or above/below the Golden Receiver Power Range (GRPR) [65]. A positive or negative RSSI (in dB) means that the received power level is above or below GRPR, respectively, while a zero implies that it is ideal (i.e., within GRPR). Next, we proceed to investigate the RSSI's relationship with distance, and consequently, infer how it might affect positioning systems. Let $P(d_1)$ and $P(d_2)$ denote the upper and lower GRPR thresholds of the intended receiver, and assume that these power levels are detected at distances d_1 and d_2, respectively, from the transmitter. According to the free-space propagation model,

$$P(d_1) \propto \frac{1}{{d_1}^2} \text{ and } P(d_2) \propto \frac{1}{{d_2}^2}, \text{ giving } \frac{P(d_1)}{P(d_2)} = \frac{{d_2}^2}{{d_1}^2}, \tag{3.1}$$

where the proportionality constant is the same. If we consider 20 dB path loss between these two distances, which is approximately the nominal GRPR range, we get

$$10 \times \log \frac{P(d_1)}{P(d_2)} = 20. \tag{3.2}$$

Combining (3.1) and (3.2), we finally obtain

$$\frac{d_2}{d_1} = 10. \tag{3.3}$$

The above calculation implies that the RSSI remains at 0 when the transmitter-receiver separation ranges between d_1 and d_2, although they differ by a factor of 10. Hence, we may not be able to differentiate over a wide area if we rely on RSSI for localization. To aggravate the problem, Bluetooth devices may request the transmitter to perform power control, so as to keep its received power level within GRPR. Suppose the devices choose to perform power control over a range of 20 dB (the margin may be even larger according to Bluetooth specification). If we add this quantity to the 20 dB GRPR range, it means that we can no longer discriminate path losses of 40 dB. Following the same analysis as before, it can be seen that, a device that is only 10 cm away may not be distinguishable from one that is 10 m away. This wide range is unacceptable for indoor localization. Hence, RSSI is argued to be a poor candidate for Bluetooth positioning systems.

3.3.2 Link Quality (LQ)

LQ is derived from the average bit error rate (BER) seen at the receiver, and is constantly updated as packets are received. For our experiments, we have chosen Ranger's BT-2100 Bluetooth USB adapters, which use BlueCore4-ROM chips from Cambridge Silicon Radio (CSR). Since LQ is an 8-bit unsigned integer, it can only assume 256 different values to represent various BER conditions. The CSR chips report LQ with finer BER resolution when BER is small, but as the BER increases, the resolution becomes coarser [67]. According to Bluetooth specification, a link is only considered workable if its BER is at most 0.1%. Therefore, the CSR chipsets map LQ values below 215 with a coarser BER resolution, as the link is already considered undesirable. This in

turn has adverse effects on localization because the LQ parameter may not provide desirable distinct signatures when the links are error-prone.

Another important consideration is that, the LQ conversion algorithms among different chipsets made by different vendors may differ; therefore, prior works [21, 52] that have relied on the mobile device's perceived LQ as location fingerprints may actually suffer from performance degradation if they were to use devices that contain chipsets from different vendors other than the one used during the training phase.

3.3.3 Transmit Power Level (TPL)

TPL is an 8-bit signed integer which specifies the Bluetooth module's transmit power level (in dBm). The power control feature is introduced into Bluetooth devices in order to facilitate energy conservation, and also to combat interference. The step size for power adjustments ranges between 2 and 8 dB. Upon the receipt of a power control request message, the TPL is increased or decreased by one step.

According to Bluetooth specification, Class 1 devices are advised to perform power control even when the power is below -30 dBm. However, for the convenience of analysis, we assume here that the minimum selectable power is -30 dBm. In this scenario, Class 1 devices can thus vary its power over a range of 50 dB, since the maximum attainable power for Class 1 devices is $+20$ dBm. If we consider the minimum step size for power control (i.e., 2 dB), then there can be at most $50 \div 2 = 25$ different TPL values for distinguishing unique locations, which are quite limited.

Our CSR adapters offer updated RSSI measurements once every second. Therefore, if it takes four power control steps to eventually reach a stabilized TPL for a specific location, the overhead can be as long as 4 seconds (ignoring transmission and processing delays), which contributes to the overall latency of such a positioning system.

3.3.4 Inquiry Result with RSSI/RSS

Every inquiry that is sent and replied by a device will be transmitted at a device-specific default power setting. As a result, the RSSI fetched through an inquiry is free from the side-effect of power control as explained earlier. Hence, the inquiry-fetched RSSI is expected to provide finer measurements than the connection-based RSSI, although it still suffers from the GRPR-related zero-RSSI problem. On the other hand, if the inquiry results are provided as absolute RSS, as is the case in the chipset we used, GRPR-related problem is diminished.

3.4 Experimental Findings

Since most of the fingerprinting techniques are built on top of a Wi-Fi infrastructure, many works have experimented with the available signal strength parameters (mainly RSS) as location fingerprints and noticed their pitfalls regarding localization [18, 25, 26, 28–30]. However, no prior work has actually investigated the prospects of Bluetooth's various signal strength parameters available as location fingerprints. Few works [52, 61, 62] have used either RSSI or LQ as location fingerprint which is ill-suited for localization purpose as evident from our analysis in the previous section. Next, we present the results from our experiments for a complete understanding of the Bluetooth's signal strength parameters from a localization perspective. All the experiments pertaining to this section have been conducted inside our Bluetooth testbed (i.e., Testbed 2).

3.4.1 Signal parameters' correlation with distance

For this experiment, we carefully chose five different grid positions where we took readings from each of the 3 APs, thus resulting in 15 data points. We adopted this

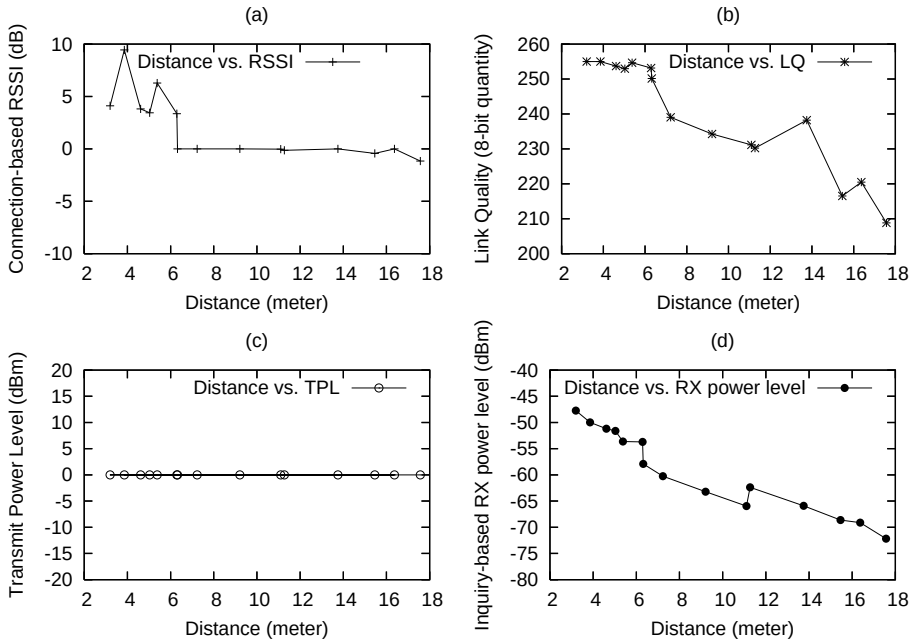


Figure 3.4: Relationship between various Bluetooth signal parameters & distance.

methodology, rather than choosing 15 distinct distances from a single AP, because we wanted to correlate distance with signals originating from APs that were placed at different locations and surroundings.

In our experiments, we discovered that the Bluetooth wireless signal strengths tend to vary quite significantly depending on the user's orientation. Therefore, for every chosen grid position, we took 30 readings from every AP for each of the four different orientations. We then calculated the average of these 120 readings to obtain the signal parameter's value for that particular AP at the specific grid position. Since we know the distances of all grid positions from any AP, the signal strength values are simply mapped against the corresponding distances to generate Fig. 3.4. In order to acquire the connection-based status parameter readings (i.e., RSSI, LQ, and TPL), we maintained connections at the HCI level from the APs to our mobile host.

From Fig. 3.4, the following observations can be made:

- As anticipated in our earlier analysis, RSSI turns out to correlate poorly with

distance, as shown in Fig. 3.4(a).

- Fig. 3.4(c) shows a horizontal straight line for TPL values. This is because our Class 2 adapter at the mobile host which uses Broadcom's BCM2035 chip does not support power control feature. As a result, the TPL at the AP remained at its default value, which happens to be 0 dBm for the Bluetooth adapter used.

- From Fig. 3.4(b), we see that LQ correlates with distance much better than RSSI and TPL, although the LQ readings obtained at smaller distances show very little variation. Note that these readings were taken at the AP side, rather than at the mobile host side, as the LQ perceived at our mobile host was always 255 at any grid position, which is the highest possible LQ value. This is due to our Class 1 APs' large transmit power. The measurements at the AP side, on the other hand, show variations because our mobile host uses a Class 2 adapter.

- Our BT-2100 Class 1 adapters provide absolute RX power level through inquiry, instead of the relative RSSI values as suggested by Bluetooth specification. As the parameter "Inquiry Result with RSSI" also suffers from the GRPR-related zero-RSSI problem (just like the "connection-based RSSI"), we believe that making RX power level available should augur well in terms of distance. Fig. 3.4(d) certainly establishes this claim since the RX power level shows the best correlation with distance, compared to the other three signal parameters.

3.4.2 Effect of GRPR on RSSI

Fig. 3.5 illustrates the adverse effects of wider GRPR on the reported RSSI. From the figure, it is seen that BT-2100's RSSI readings (GRPR $\approx$ 80 dB) showed little variation compared to our Broadcom's adapter, which has a narrower GRPR. Because of the combined effect of large GRPR and power control, BT-2100's RSSI readings

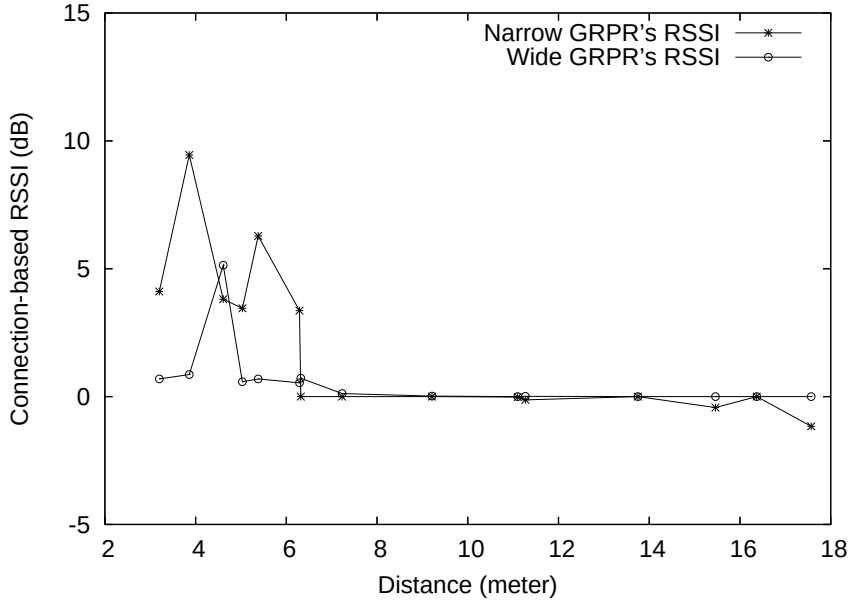


Figure 3.5: Connection-based RSSI for two Bluetooth adapters with different GRPR.

always remained at or above 0. On the contrary, Broadcom's adapter gave negative RSSI values at greater distances, although we did not have many such grid positions owing to our testbed's size.

3.4.3 TPL Consideration

For this experiment, we recorded the stabilized TPL values as well as the stabilization time periods for each AP's signal at specific grid positions using BT-2100 at the mobile host side. Fig. 3.6(a) indeed shows very few discrete transmit power levels, in harmony with our analysis in Section 3.3.3. Moreover, the time periods required to reach these stabilized TPL values are also quite significant, as revealed in Fig. 3.6(b). Both these attributes make TPL a poor candidate for localization purpose.

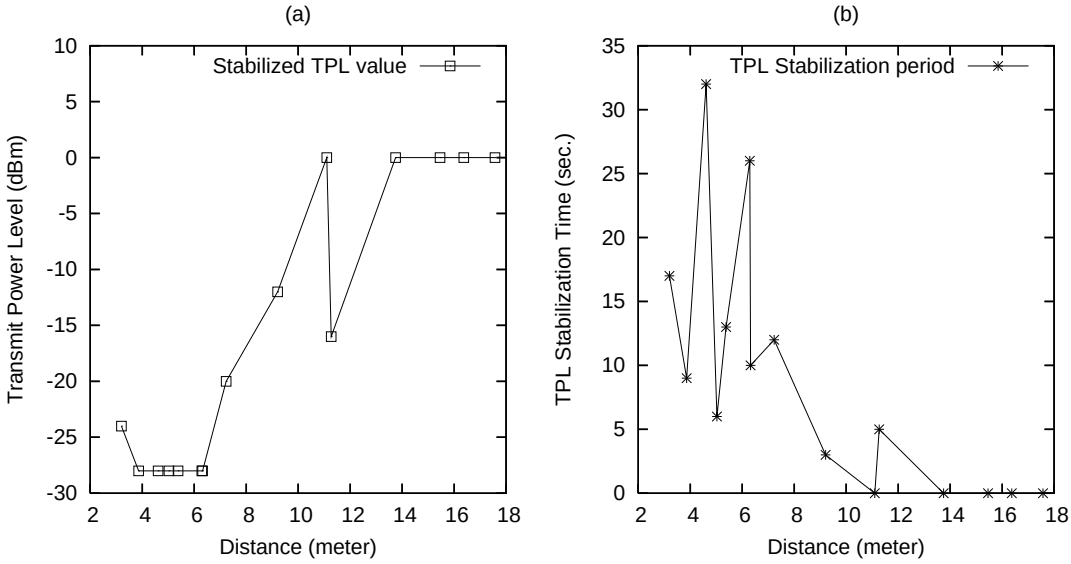


Figure 3.6: Stabilized TPLs & time periods to attain them.

3.5 Summary and Conclusions

In Table 3.3 and 3.4, we summarize the suitability of the available signal parameters as location fingerprints for Wi-Fi and Bluetooth, respectively. We emphasize on the point that each signal parameter has particular usage according to its own respective technology. Consequently, their inherent characteristics may render them inappropriate to be considered as location fingerprints which we have investigated thoroughly here. Based on our analysis, RSS seems to be the most viable option as location fingerprint compared to all the other signal parameters available for Wi-Fi. However, RSS has been observed to vary at the same location depending on different devices' hardware even under the same wireless conditions [25–29], which ultimately has adverse effect on fingerprinting solutions. From our analysis and experimental findings, it is apparent that RSS has the most potential compared to the other currently available Bluetooth location fingerprints. However, it may also not be robust when different devices are considered (e.g., Class 1, 2, etc.). Next, we deduce our robust location fingerprint, the *Signal Strength Difference (SSD)*, and prove its superiority over RSS, both analytically and experimentally.

Table 3.3: A qualitative overview of the characteristics of Wi-Fi technology's available signal parameters and their pitfalls regarding localization

	Main Purpose	Issues Regarding Localization
RSS	• RSS is translated from RSSI which is meant for internal use by the Wi-Fi NIC. • RSSI is utilized in Wi-Fi's CSMA-based MAC protocol to give an idea about whether the channel is clear to send, or to decide when a device should attempt to roam.	• Most popular RF location fingerprint. • Inconsistent among different transmitter-receiver pair configurations. • Different vendors' varying interpretations also have adverse effect on this fingerprint. • The above two effects collectively degrade its robustness as a location fingerprint.
SQ	• Gives a measure of the channel condition based on the BER observed at the receiver. • Defined as the "PN code correlation strength" in 802.11 family which uses DSSS modulation schemes.	• Inconsistent among various vendors. • Undefined in 802.11 standard for variants of 802.11 (e.g., 802.11g). • No localization research so far tried to contemplate it as a location fingerprint.
SNR	• Gives a relative measure of the RSS compared to the ambient RF energy.	• Existing works [17, 18] reported this fingerprint to be more inconsistent than RSS. • Positioning systems based on it reported poor accuracy so far.

Table 3.4: A qualitative overview of the characteristics of Bluetooth technology's available signal parameters and their pitfalls regarding localization

	Main Purpose	Issues Regarding Localization
RSSI	• Provides a relative measure of an established connection's RSS. • Helps in power-control.	• RSSI's correlation with distance suffers because of both GRPR and the power control feature. • Positioning systems based on RSSI have so far affirmed poor accuracy [62].
LQ	• Gives a measure of the perceived BER of an established link. • Mainly used for adapting to changes in the link's state, notably to support CQDDR (Channel Quality Driven Data Rate).	• When the links are error-prone, LQ may not provide desirable distinct signatures. • Inconsistent among different vendors. • To date, positioning systems based on LQ have reported coarse accuracy [62].
TPL	• Denotes the transmitter's output power of an established link in dBm.	• Varies depending on the Bluetooth class. • Can assume only a few distinct values. • Incurs latency to a location system based on it.
RSS	• Since Bluetooth operates on TDMA-based MAC protocol, the reporting of RSS is not mandatory, as indicated in the Bluetooth Specification [65].	• Inquiry-based RSSI is free from power control effects, but it still suffers from GRPR-related zero-RSSI problem. However, if RSS can be obtained instead of RSSI, it becomes free from that drawback as well.

Chapter 4

Robust Location Fingerprint

In existing localization literature based on fingerprints, the signal strength samples are either collected at the APs, or at the MN that needs to be located. The AP-based approach has the advantage of detecting locations of a wide range of MNs without requiring any modification of the latter, e.g., the MN need not download any additional software solely for localization purpose. On the other hand, the MN-assisted approach could better ensure the security and privacy of the MN. In both approaches, the samples' signal strength values collected over a small time-window are generally averaged to obtain the traditional RSS location fingerprint. This RSS location fingerprint has certain implications:

- It is influenced by a particular transmitter-receiver pair's hardware-specific parameters, such as antenna gains. Consequently, having a different transmitter-receiver pair compared to the training phase would likely produce a different RSS signature at the same location [26].

- Moreover, if the MN-assisted approach is used, the RSS fingerprint is likely to be different across mobile devices made by different vendors, not just due to the differences in their hardware, but also due to the vendors' own interpretations of

RSS as discussed in Section 3.2.1.

In this chapter, we show that, rather than utilizing the absolute signal strength (RSS) as location fingerprint, the differences of signal strengths perceived at the APs or at the MN would actually provide a more stable location signature for any mobile device irrespective of its hardware used. We contend that, in this way, the transmitter-receiver pair's hardware effect is mitigated. In Section 4.1, the robust location fingerprint, SSD, is first explained in detail. We then list in Section 4.2 some related works that address the same issue of hardware variations of the MN. Experimental results are presented in Section 4.3, while a summary of our findings of the robust location fingerprint, SSD, appear in Section 4.4.

4.1 Signal Strength Difference (SSD) – a robust location fingerprint

Suppose $P(d)$ and $P(d_0)$ denote the received signal strengths at an arbitrary distance d and a reference distance d_0 from the transmitter, respectively, for a particular transmitter-receiver pair. Here, we assume that the mobile device is the transmitter, while the AP is the receiver. From the log-normal shadowing model [16], we get,

$$\left[\frac{P(d)}{P(d_0)}\right]_{\text{dB}} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{\text{dB}}. \tag{4.1}$$

The first term on the RHS of (4.1) defines the path loss component (β is the path loss exponent), while the second term reflects the variation of the received power at a certain distance ($X_{\text{dB}} \sim N(0, \sigma^2_{\text{dB}})$). Eq. (4.1) can be rewritten as,

$$P(d)|_{\text{dBm}} = P(d_0)|_{\text{dBm}} - 10\beta \log\left(\frac{d}{d_0}\right) + X_{\text{dB}}. \tag{4.2}$$

Depending on the hardware used at both the AP and the mobile device, the perceived power at a reference distance (i.e., $P(d_0)$) varies, as a result of hardware-specific parameters, such as antenna gains. Therefore, the average RSS at a distance d is also hardware-dependent. This explains why RSS is not a robust location fingerprint, despite the fact that it is commonly used in the existing literature.

Rather than using absolute RSS values as location fingerprints, the difference of the RSS values observed by two APs can be used to define a more robust signature for a transmitting mobile device; we shall term this difference as *Signal Strength Difference*, or *SSD*. To explain analytically, let $P(d_1)$ and $P(d_2)$ denote the RSSs of a mobile device's transmitted signal as perceived at two different APs (AP_1 and AP_2) which are at distances d_1 and d_2 from the mobile device, respectively. We assume that, all the APs are of the same type, i.e., their hardware have the same properties. Consequently, using (4.2), we can write the following for AP_1 and AP_2 respectively:

$$P(d_1)|_{\text{dBm}} = P(d_0)|_{\text{dBm}} - 10\beta_1 \log\left(\frac{d_1}{d_0}\right) + [X_1]_{\text{dB}}, \tag{4.3}$$

$$\text{and} \quad P(d_2)|_{\text{dBm}} = P(d_0)|_{\text{dBm}} - 10\beta_2 \log\left(\frac{d_2}{d_0}\right) + [X_2]_{\text{dB}}. \tag{4.4}$$

Combining (4.3) and (4.4), we obtain,

$$\left[\frac{P(d_1)}{P(d_2)}\right]_{\text{dB}} = -10\beta_1 \log\left(\frac{d_1}{d_0}\right) + 10\beta_2 \log\left(\frac{d_2}{d_0}\right) + [X_1 - X_2]_{\text{dB}}. \tag{4.5}$$

Eq. (4.5) denotes SSD's expression which is free from $P(d_0)$, thereby, specifies a more robust location fingerprint than absolute RSS. If we assume the path loss exponent to be the same for the particular indoor environment (i.e., $\beta_1 = \beta_2 = \beta$), (4.5) can be further simplified as,

$$\left[\frac{P(d_1)}{P(d_2)}\right]_{\text{dB}} = -10\beta \log\left(\frac{d_1}{d_2}\right) + [X_1 - X_2]_{\text{dB}}. \tag{4.6}$$

Based on the above analysis, we claim that SSD provides significant improvement over traditional RSS in denoting the location fingerprint when the signal strength samples are collected at the APs. Next, we explain it in a more detailed way. We also inspect whether SSD is superior to RSS in the case of MN-assisted localization where the signal strength samples are actually collected at the MN.

4.1.1 SSD for AP-based localization approach

Consider the same example scenario as above but with the assumption that the close-in reference power, i.e., $P(d_0)$ of (4.2), can be evaluated using the free space propagation model as follows [16],

$$P(d_0)|_{\text{dBm}} = 10\log\left(\frac{P_{\text{MN}}G_{\text{MN}}G_{\text{AP}_i}\zeta_{\text{MN}}^2}{16\pi^2 d_0^2 L}\right), \tag{4.7}$$

where P_{MN} is the MN's transmitted power, G_{MN} is the MN's antenna gain, G_{AP_i} is the i^{th} AP's antenna gain, L is the system loss factor, and ζ_{MN} is the transmitted carrier's wavelength (same unit as d_0).

Using (4.7), both (4.3) and (4.4) can be rewritten respectively as,

$$P(d_1)|_{\text{dBm}} = 10\log\left(\frac{P_{\text{MN}}G_{\text{MN}}G_{\text{AP}_1}\zeta_{\text{MN}}^2}{16\pi^2 d_0^2 L_1}\right) - 10\beta_1\log\left(\frac{d_1}{d_0}\right) + [X_1]_{\text{dB}}, \tag{4.8}$$

$$\text{and } P(d_2)|_{\text{dBm}} = 10\log\left(\frac{P_{\text{MN}}G_{\text{MN}}G_{\text{AP}_2}\zeta_{\text{MN}}^2}{16\pi^2 d_0^2 L_2}\right) - 10\beta_2\log\left(\frac{d_2}{d_0}\right) + [X_2]_{\text{dB}}. \tag{4.9}$$

Here, the APs' antenna gains (i.e., G_{AP_1} and G_{AP_2}) and the miscellaneous losses (L_1 and L_2) would be the same because of our previous assumption which mentions that the APs are of similar hardware properties. As a result, combining (4.8) and (4.9) yields SSD's expression of (4.5). Consequently, we claim that while RSS may vary using different mobile devices as can be seen from (4.8) or (4.9), SSD is free from that drawback as evident in (4.5).

4.1.2 SSD for MN-assisted localization approach

We consider the same example scenario as above, except that the signal strength is now measured at the MN rather than at the APs. Subsequently, (4.8) and (4.9) take the following forms, respectively,

$$P(d_1)|_{\text{dBm}} = 10\log\left(\frac{P_{\text{AP}_1}G_{\text{AP}_1}G_{\text{MN}}\zeta^2_{\text{AP}_1}}{16\pi^2 d_0^2 L_1}\right) - 10\beta_1\log\left(\frac{d_1}{d_0}\right) + [X_1]_{\text{dB}}, \quad (4.10)$$

$$\text{and } P(d_2)|_{\text{dBm}} = 10\log\left(\frac{P_{\text{AP}_2}G_{\text{AP}_2}G_{\text{MN}}\zeta^2_{\text{AP}_2}}{16\pi^2 d_0^2 L_2}\right) - 10\beta_2\log\left(\frac{d_2}{d_0}\right) + [X_2]_{\text{dB}}. \quad (4.11)$$

The APs' antenna gains (i.e., G_{AP_1} and G_{AP_2}) and the miscellaneous losses (L_1 and L_2) are the same as discussed in Section 4.1.1. The ζ's will be different if the APs operate in different channels but this difference is usually not very significant [68]. The samples gathered at the MN are mainly derived from the probe replies that come from the APs [26]. Since these replies are generally sent using some default power setting, we can have the approximation, $P_{\text{AP}_1} \approx P_{\text{AP}_2}$. Under these conditions, the SSD's expression in (4.5) can be obtained by combining (4.10) and (4.11), and thereby, denotes a more robust location fingerprint compared to RSS.

4.2 Related Work

The effects of different devices' hardware variations on RF location fingerprint have gained little attention in the localization literature so far. As discussed before, existing works generally use the same mobile device during the training and testing phases, thereby, invoking similar setups (i.e., transmitter-receiver pair) in both cases. However, [25–29] have observed that the location fingerprints (i.e., RSSs) produced by using different mobile devices vary quite significantly from one another even under the same wireless conditions. Haebarlen *et al.* [26] try to accommodate various de-

vices by having a benchmark training database taken with only one device. For other devices, they require a set of linear RSS conversion formulae, which translate the RSSs of those devices into the benchmark device's RSSs. These linear conversion formulae are obtained by laboriously experimenting with each supported device to discover its RSS relationship with that of the benchmark device. Kjærgaard [28] follows a similar approach. Tao *et al.* [25] utilize signal strength difference as a location fingerprint like our approach. Their motivation was to find the locations of rogue machines with different hardware configurations and varying transmitting powers, and they have only provided experimental results based on the idea. They did not provide any intuition or analysis about why the differences in signal strengths could work successfully in their scenarios. On the contrary, our work gives both the detailed analysis and the experimental results as to why the SSD could be regarded as a *robust* location fingerprint.

There are two other techniques in the literature that could mitigate the effect of MN's hardware variations without any additional calibration steps like the aforementioned works [26, 28] – Hyperbolic Location Fingerprint (HLF) [29] and Ecolocation [69]. HLF [29] uses logarithm of signal strength ratios between pairs of APs. However, they do not give any analytical basis as to why it mitigates the hardware variation effects. Their log signal strength ratios are actually just the RSS differences in the log scale. Taking log of (4.3) and (4.4), and combining, it can be seen that the resulting expression is not totally free from $P(d_0)$, unlike our SSD's expression (4.5). Ecolocation [69] uses ordered sequence of RSS measurements rather than the absolute RSSs to constitute a unique location fingerprint. If $P(d_i)$ and $P(d_j)$ denote the RSSs at AP_i and AP_j, which are at distances d_i and d_j from the MN, respectively, then a *constraint* of the sequence is defined as,

$$P(d_i) > P(d_j) \Rightarrow d_i < d_j. \tag{4.12}$$

First, the constraint set for each grid point is calculated using the RHS of (4.12). Only the locations of *reference nodes* (i.e., APs) are required in this phase – no signal strength collection surveys are necessary. During location determination phase, the ordered sequence of RSSs collected at the APs is translated into the ordered sequence of distances using (4.12), and subsequently matched against the constraint set of each grid point calculated beforehand. The centroid of the grid points where the maximum number of constraints are matched is returned as the location estimate. We believe that, owing to MNs' hardware variations and varying transmission powers, both $P(d_i)$ and $P(d_j)$ should be affected in a similar way. Therefore, the constraint (4.12) is expected to remain intact over different MNs. Consequently, Ecolocation could be robust against MNs' hardware variations as well.

4.3 Experimental Results and Findings

We first list the assumptions that we have made for our experiments in Section 4.3.1, and then discuss our results in subsequent sections.

4.3.1 Assumptions

- In this dissertation, whenever we have used RSS as location fingerprint for certain experiments, we have assumed it to be normally distributed at a particular location. Though some works defy this phenomenon, others lend support to it [30]. We denote the RSS location fingerprint to be normal distributed which is defined by only two parameters. One instance of RSS distribution at a particular location inside Testbed 1 is shown in Fig. 4.1. Similar to [26], our experimental results also suggest that it is a reasonable approximation, as significant improvement cannot be achieved even if we were to utilize histogram representations of RSS. However, we have used the histogram representation for HLF and the

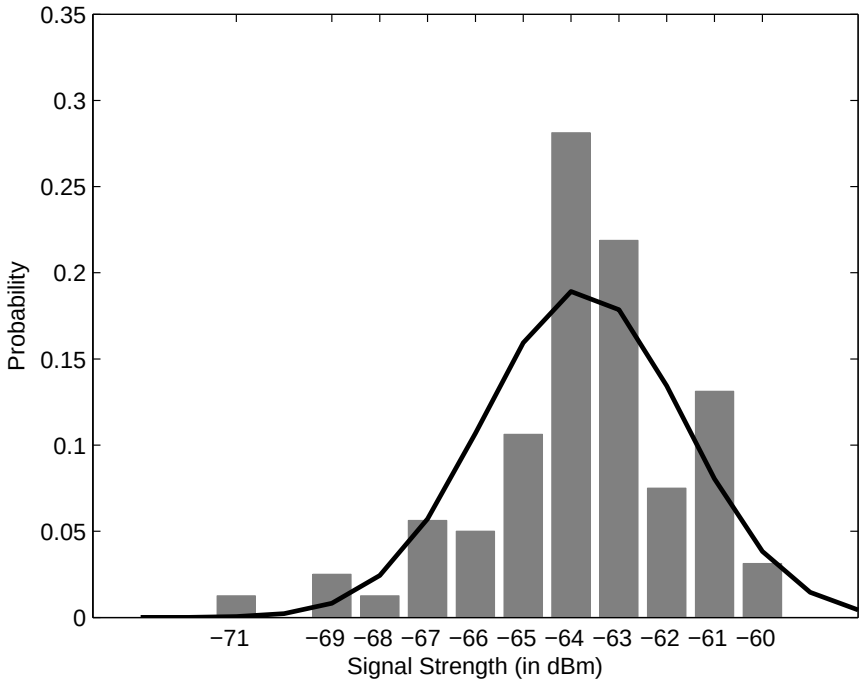
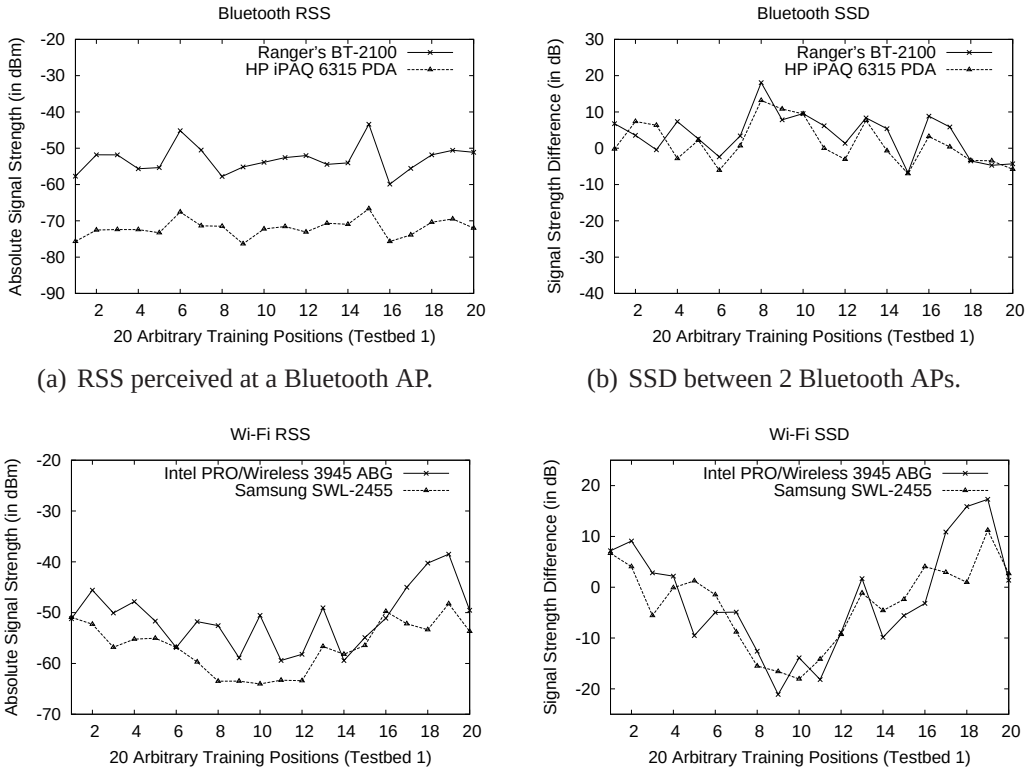


Figure 4.1: Histogram of received signal strength (RSS) at a particular training point regarding an AP and its Gaussian approximation.

histogram's bin size is selected to be 0.02 as suggested by [29].

- We have chosen two well-known algorithms in the localization literature, namely, $\mathcal{K}$-NN in signal space [17] and Bayesian Inference [26], in order to test our ideas. These two algorithms are discussed in Section 2.2.5 and 2.2.6, respectively. Our key intention is to show that our ideas are quite generic and can be helpful irrespective of the choice of algorithms. For the $\mathcal{K}$-NN algorithm, we choose the value of $\mathcal{K}$ empirically, similar to prior works [17]. Based on our experimental findings, we select $\mathcal{K} = 4$. While applying Bayes formula, the priori probabilities are assumed to be uniformly distributed.

- In order to apply probabilistic models, one assumption that has widely been used is the independence of RSS values of different APs [18, 20]. This assumption is justifiable for a well-designed network where each AP runs on a non-overlapping



(a) RSS perceived at a Bluetooth AP.

(b) SSD between 2 Bluetooth APs.

(c) RSS perceived at a Wi-Fi AP.

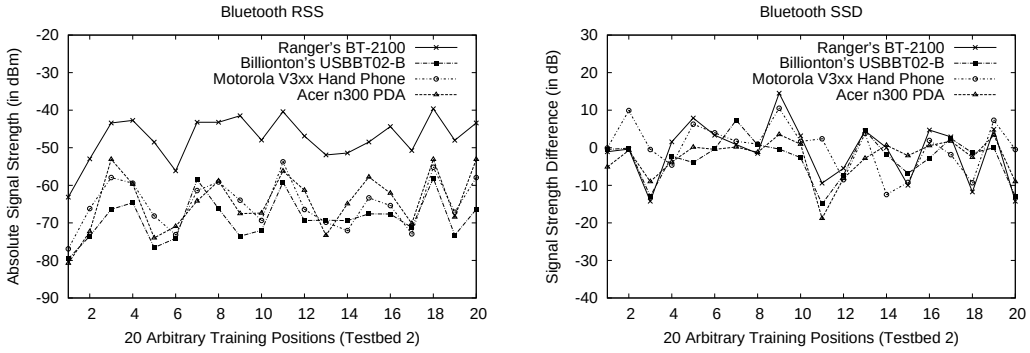
(d) SSD between 2 Wi-Fi APs.

Figure 4.2: RSS and SSD considering 2 different devices (a laptop and a PDA) incorporated with both Bluetooth and Wi-Fi capability (Testbed 1).

channel. Kaemarungsi and Krishnamurthy have performed experiments in [30] to evaluate the correlation factor among the APs' RSS values in the presence of interference and they have strengthened this claim as well. Thus, we have also adopted their vindication.

4.3.2 Justification of SSD as a robust fingerprint

For this experiment, we have chosen various mobile devices which were listed in Table 3.1 in order to see their effects on both RSS and SSD location fingerprint. In Testbed 1, we have selected two different devices (a laptop and a PDA) and measured their signal strengths at the APs (i.e., mini PCs). Our laptop is installed with an Intel PRO/Wireless 3945 ABG Mini PCI WLAN adapter whereas the WLAN card used in



(a) RSS perceived at a Bluetooth AP. (b) SSD between 2 Bluetooth APs.

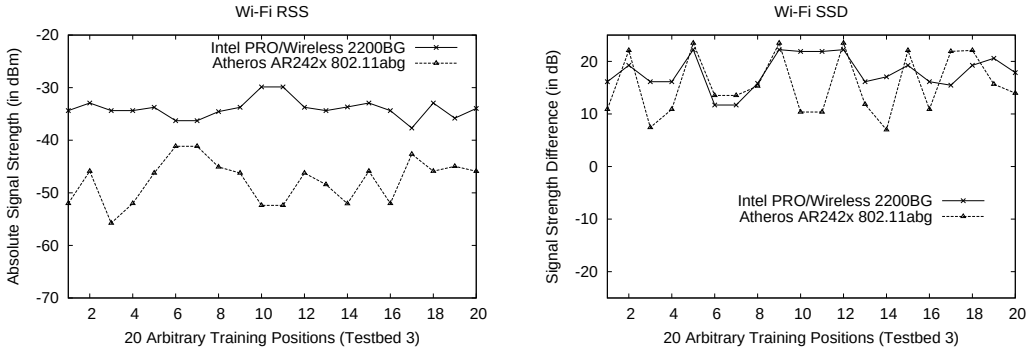
Figure 4.3: RSS and SSD considering 4 different Bluetooth devices (Testbed 2).

our PDA is Samsung SWL-2455 802.11b. As for Bluetooth, our HP iPAQ PDA has an integrated Class 2 Bluetooth chip, whereas a BT-2100 Class 1 Bluetooth USB adapter has been plugged into the laptop during the experiments.

We have picked 20 random training points and stationed the devices at those locations, while ensuring that we have collected enough samples at the APs for both devices. Fig. 4.2(a) and 4.2(c) are drawn with the RSS readings seen by a particular AP, whereas Fig. 4.2(b) and 4.2(d) plot the difference between the RSS values seen at two different APs.

We repeat similar experiments for our Testbed 2 where four different Bluetooth devices are used. The Acer n300 PDA, Motorola V3xx phone and USBBT02-B adapter are the three Bluetooth Class 2 devices while the Ranger's BT-2100 is a Class 1 adapter. All these devices are stationed at the various training locations in order to measure their signal strengths at the APs. The RSS at a particular AP and the SSD between two different APs for 20 such locations are depicted in Fig. 4.3(a) and 4.3(b), respectively.

For Testbed 3, we conducted the signal strength survey by plugging two different Wi-Fi NICs (Intel PRO/Wireless 2200BG and Atheros AR242x 802.11abg) into our laptop. Since our Testbed 3 emulates the MN-assisted localization scenario, we actually collected the signal strength samples at the MN rather than at the APs like

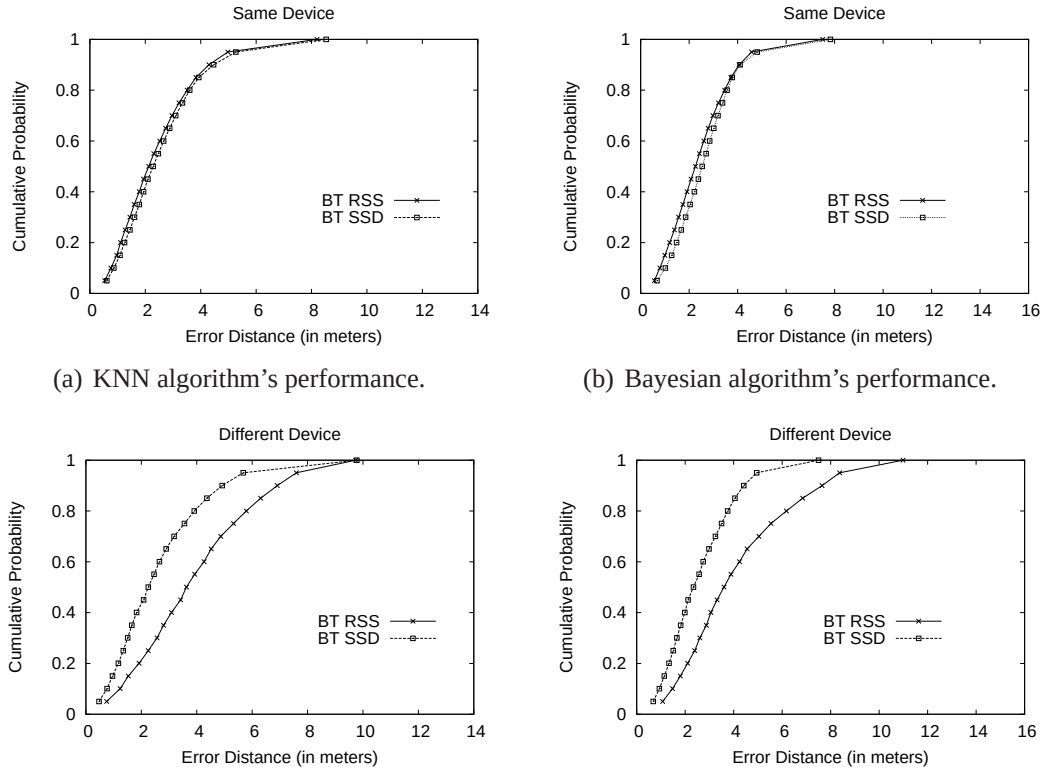


(a) RSS perceived at MN from a Wi-Fi AP. (b) SSD between 2 Wi-Fi APs perceived at MN.

Figure 4.4: RSS and SSD considering 2 different Wi-Fi devices (Testbed 3).

our previous two testbeds. Fig. 4.4(a) shows the RSSs perceived at the two different NICs from an AP (i.e., Linksys WRT54G router) whereas Fig. 4.4(b) depicts the SSDs between two different APs perceived at them.

From Fig. 4.2(a), 4.2(c), 4.3(a) and 4.4(a), it is apparent that, the absolute signal strength perceived at/from a certain AP varies quite significantly between the two devices at each training location. This has repercussion in their use as fingerprints because they are quite different when different mobile devices are used during training. Most works perform their training and testing phase with the same device, thereby, shielding the adverse effect of this phenomenon. On the contrary, the SSD does not suffer much from this effect, thereby, providing a more robust fingerprint as seen in Fig. 4.2(b), 4.2(d), 4.3(b) and 4.4(b). This readily complies with our analysis in Section 4.1. Note that the fluctuations of SSD could be as bad as RSS at some locations. However, SSD is observed to be more stable in our experiments overall. Only the findings at 20 randomly selected locations are presented here. Furthermore, although the SSDs between only one pair of APs are shown, choosing any pair of APs to calculate the SSDs yields improvements over RSS in our experiments. We also notice from Fig. 4.2(b) and 4.3(b) that, the SSD readings obtained for Bluetooth tend to be more robust compared to the Wi-Fi SSDs (Fig. 4.2(d) and 4.4(b)), which will be further



(a) KNN algorithm's performance.

(b) Bayesian algorithm's performance.

(c) KNN algorithm's performance.

(d) Bayesian algorithm's performance.

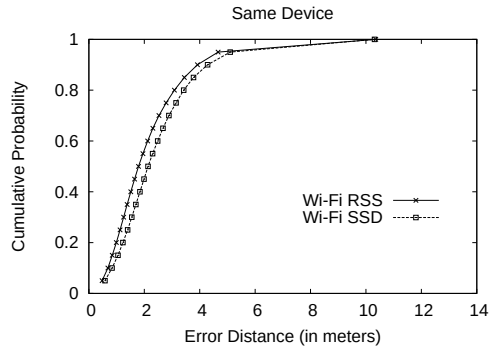
Figure 4.5: Comparison of error performance using RSS vs. SSD as location fingerprint for Bluetooth when the testing phase is conducted with the same training device or a different device.

verified by our later results.

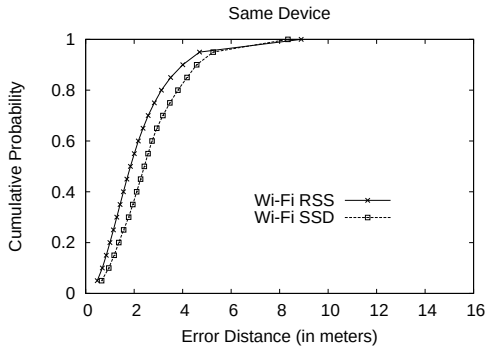
4.3.3 Comparison of SSD and RSS as Location Fingerprint

As pointed out in the previous section, the usage of the same MN for both training and testing phase may have biased the reported results of the existing fingerprinting techniques. To investigate further, we conducted experiments in both our AP-based Bluetooth (Testbed 2) and MN-assisted Wi-Fi (Testbed 3) testbeds to visualize the effect of MN's hardware variations.

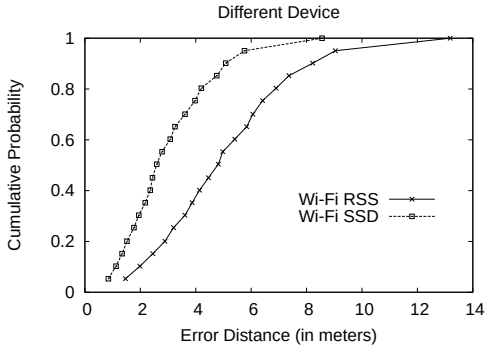
In Testbed 2, we have chosen Bluetooth solutions from four different manufacturers and types as discussed in the previous section for training of the 337 data points



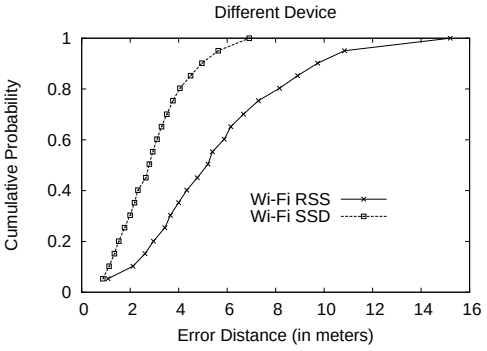
(a) KNN algorithm's performance.



(b) Bayesian algorithm's performance.



(c) KNN algorithm's performance.

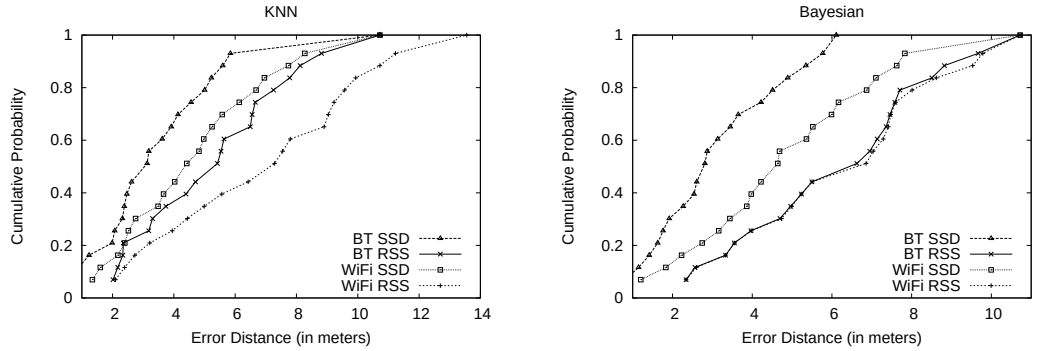


(d) Bayesian algorithm's performance.

Figure 4.6: Comparison of error performance using RSS vs. SSD as location fingerprint for Wi-Fi when the testing phase is conducted with the same training device or a different device.

shown in Fig. 3.2. We separate Ranger's BT-2100 Class 1 adapter's data set as our training samples, while the rest, $(3 \times 337) = 1011$ samples from the other three Class 2 devices are used for testing. In Testbed 3, we have Wi-Fi NICs from two different manufacturers and types as discussed in the previous section. The Intel NIC's collected data at 466 grids shown in Fig. 3.3 are kept as training while the Atheros NIC's collected data at 244 locations are utilized for testing purpose.

To inspect the "same device" effect, we choose Ranger's BT-2100 Class 1 adapter in Testbed 2 to perform both the training and testing phase. Among the 337 training grids, 200 of them are selected randomly as training points while the rest 137 are kept for testing purpose. We then run our algorithms (i.e., $\mathcal{K}$-NN and Bayesian which are discussed in Section 2.2.5 and 2.2.6, respectively) to obtain the localization errors. We



(a) KNN algorithm's performance w.r.t. RSS and SSD as location fingerprint.

(b) Bayesian algorithm's performance w.r.t. RSS and SSD as location fingerprint.

Figure 4.7: Comparison of error performance when using RSS vs. SSD as location fingerprint for both Bluetooth and Wi-Fi (Testbed 1).

repeat this procedure for 101 times to obtain all the errors for different combinations of training and testing samples, and finally come up with the cumulative probability graph. In Testbed 3, Intel PRO/Wireless 2200BG Wi-Fi NIC has been utilized for both the training and testing phase. In this particular testbed, 200 of the 466 training grids shown in Fig. 3.3 are selected randomly as training points, while the rest 266 are kept for testing purpose. We follow similar approach as the one described for Testbed 2 in order to obtain the cumulative probability graph of errors.

From Fig. 4.5(c), 4.5(d), 4.6(c), and 4.6(d), it is apparent that hardware variations of the MN during the testing phase have adverse effect on the RSS based localization performance for both Bluetooth and Wi-Fi. We further notice that, this issue is prevalent regardless of whether the RSS is measured at the APs for AP-based localization (our Bluetooth Testbed 2) or at the MN for MN-assisted localization (our Wi-Fi Testbed 3), This is a severe shortcoming of the fingerprinting techniques since one cannot assume the users to carry the same device with which the training of the system has been performed. On the contrary, SSD based localization performs quite well under hardware variations for both Wi-Fi and Bluetooth and is superior to RSS based localization in all cases (see Fig. 4.5(c), 4.5(d), 4.6(c), and 4.6(d)). The error performance

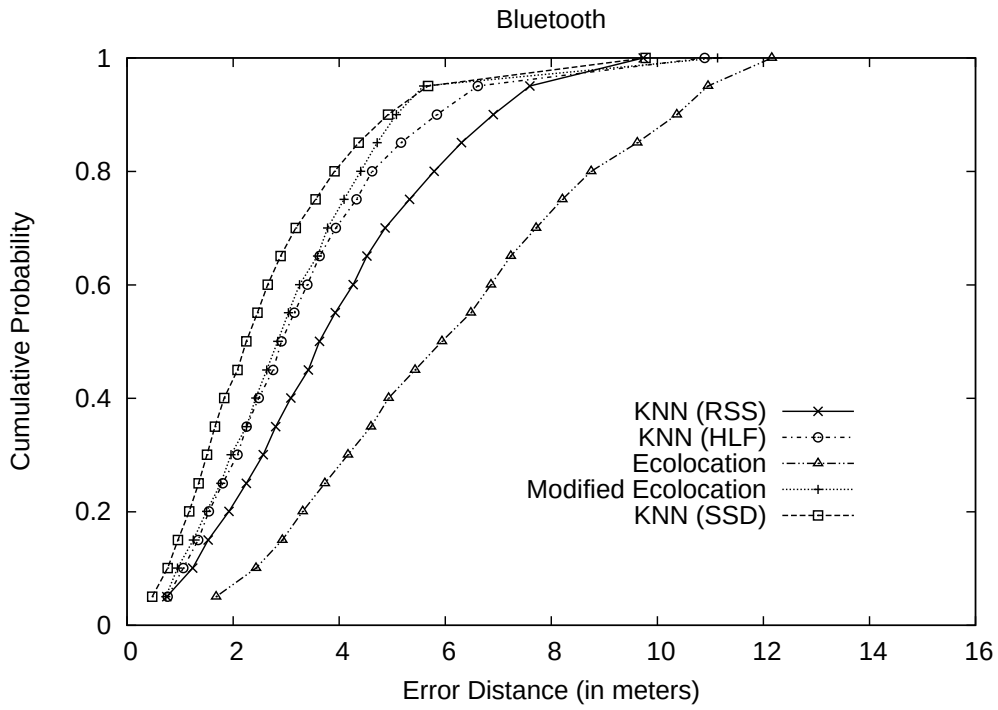


Figure 4.8: Comparison of localization error performance when using various location fingerprints in KNN localization algorithm for Bluetooth.

when using the same device for training and testing can be visualized in Fig. 4.5(a) and 4.5(b) for Bluetooth, and in Fig. 4.6(a) and 4.6(b) for Wi-Fi. The better performance of RSS based algorithms compared to our SSD based algorithms is a pitfall since in real practical scenarios, all the users would hardly carry the same device as the training device.

We conduct another experiment in Testbed 1 which is equipped with both Wi-Fi and Bluetooth capabilities in order to compare Bluetooth and Wi-Fi's performance regarding SSD. For this experiment, we have chosen the laptop's data at the 62 training points (shown in Fig. 3.1) and the PDA's data at the 44 testing points. The Bluetooth and Wi-Fi device details on the laptop and PDA can be found in the previous section and also in Table 3.1. As illustrated in Fig. 4.7(a) and 4.7(b), it can be seen that, the positioning system built upon SSD again outperforms its RSS counterpart for both Bluetooth and Wi-Fi. Furthermore, we also see that, the Bluetooth SSD based systems

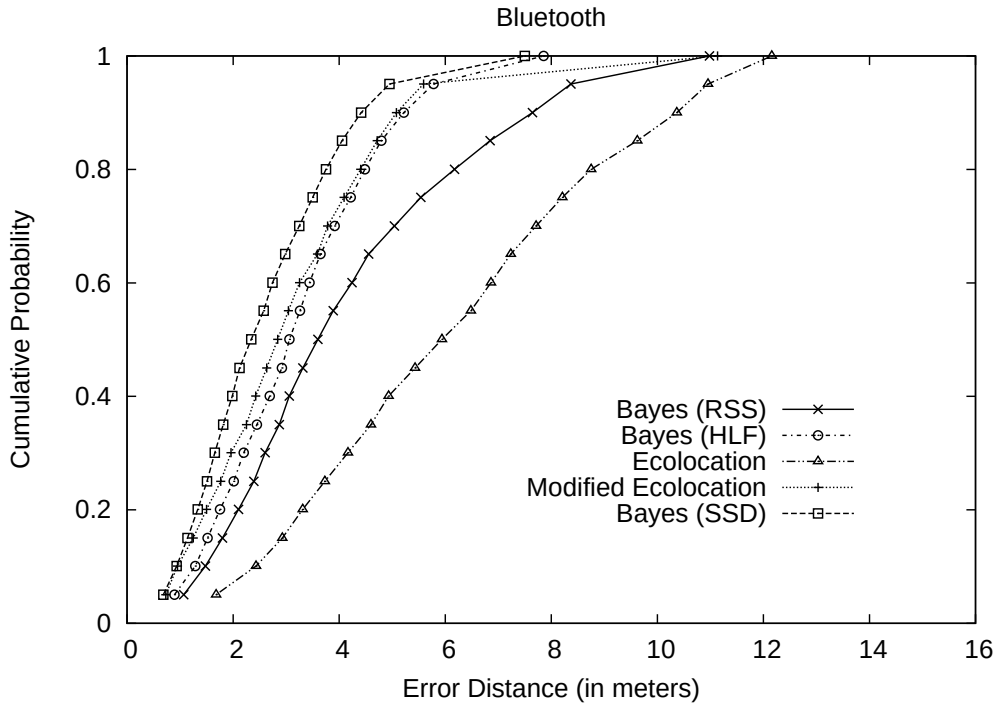


Figure 4.9: Comparison of localization error performance when using various location fingerprints in Bayes localization algorithm for Bluetooth.

perform better than positioning systems utilizing Wi-Fi SSD, as anticipated in the previous section. The average errors of our Bluetooth SSD based localization algorithms (2.58 m for $\mathcal{K}$-NN and 2.55 m for Bayesian) in Testbed 2 are also smaller than Wi-Fi SSD based localization algorithms (2.94 m for $\mathcal{K}$-NN and 2.89 m for Bayesian) in Testbed 3.

4.3.4 Comparison of SSD with Other Robust Location Fingerprints

The results presented in this particular section are obtained from the experiments conducted in our Bluetooth Testbed 2 and Wi-Fi Testbed 3. As discussed in the previous section, for Testbed 2, we separate BT-2100 Class 1 adapter's data set as our training samples, while the rest, $(3 \times 337) = 1011$ samples from the other three Class 2 devices are used for testing. Similarly, for Testbed 3, the Intel NIC's collected data at 466 lo-

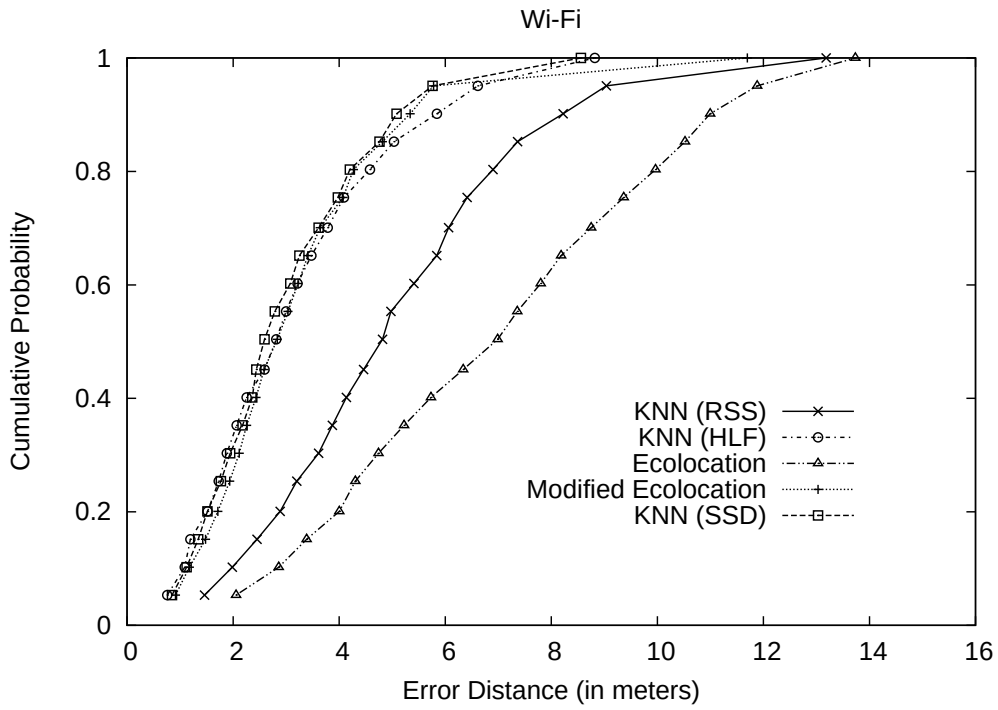


Figure 4.10: Comparison of localization error performance when using various location fingerprints in KNN localization algorithm for Wi-Fi.

cations are kept as training while the Atheros NIC's collected data at 244 locations are utilized for testing purpose.

In case of Bluetooth, it is evident from Fig. 4.8 and 4.9 that, SSD based techniques are better than the other two schemes (HLF and Ecolocation) described in Section 4.2 that could also mitigate the MNs' hardware variation effects. The numerical values of these two figures can be found in Table 4.1. Similar conclusions could be drawn for Wi-Fi SSD based techniques as well from Fig. 4.10 and 4.11. The numerical values of these figures are listed in Table 4.2.

For both Wi-Fi and Bluetooth, we see that, Ecolocation performs even worse than the RSS based algorithms. This can be attributed to the following reasons: i) Ecolocation is mainly targeted at localizing inexpensive sensors and is shown to perform better than other localization algorithms found in wireless sensor networks [69]. Its main advantage lies in the fact that it requires no time-consuming signal strength collection

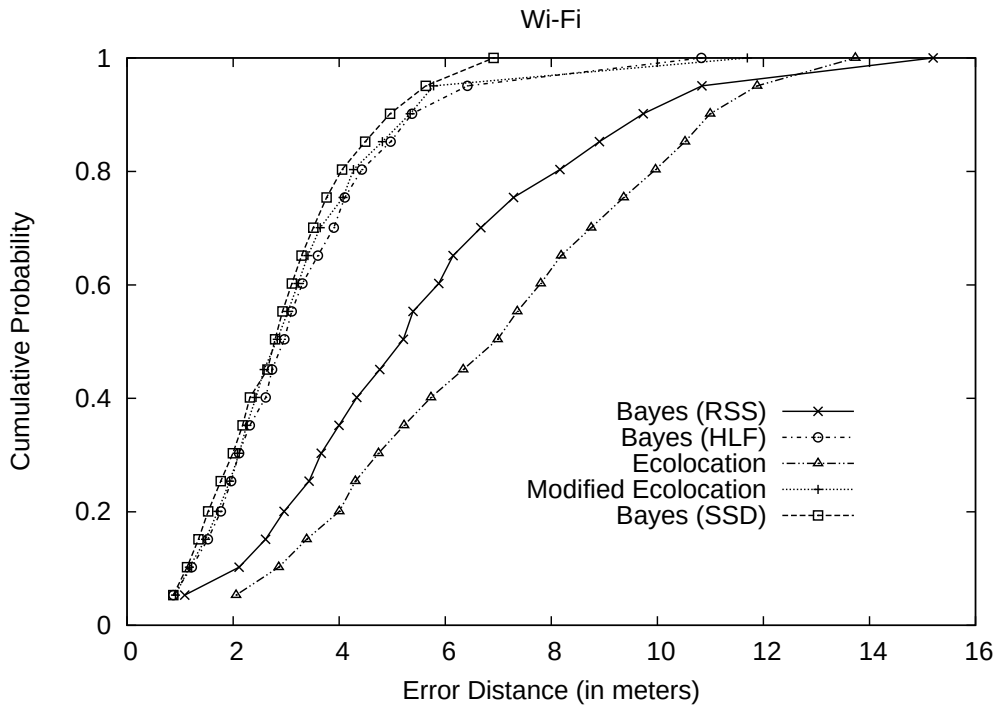


Figure 4.11: Comparison of localization error performance when using various location fingerprints in Bayes localization algorithm for Wi-Fi.

surveys in the location space, whereas all the other algorithms considered here make use of the offline training phase data. ii) RSS measurements do not represent distances accurately in the real world. Therefore, uncertainties could arise while using (4.12) as discussed in [69]. Moreover, since we only have four APs in each testbed, the number of constraints (i.e., $\binom{4}{2}$) at each grid point is also quite limited.

For fair comparison, we modify Ecolocation by making use of the offline training phase data. The constraint set for each grid point of the modified algorithm consists of the ordered sequence of RSS values collected during the training phase instead of the distance constraints as discussed in Section 4.2. The ordered sequence of RSSs collected during the location determination phase is now directly compared with each grid point's constraint set without the need for translation into distance constraints using (4.12). As evident from Fig. 4.8, 4.9, 4.10 & 4.11, and Table 4.1 & 4.2, the performance of Ecolocation is enhanced significantly, and the modified algorithm com-

Table 4.1: Percentile values and averages of errors (in meter) when various fingerprints are considered for Bluetooth.

Algorithm (Fingerprint)	25^{th} Percentile	Median	90^{th} Percentile	Average
KNN (RSS)	2.25	3.63	6.91	3.87
KNN (HLF)	1.81	2.91	5.84	3.21
KNN (SSD)	1.35	2.25	4.92	2.58
Bayes (RSS)	2.39	3.60	7.65	4.09
Bayes (HLF)	2.01	3.06	5.22	3.16
Bayes (SSD)	1.51	2.34	4.41	2.55
Ecolocation	3.73	5.94	10.37	6.08
Modified Ecolocation	1.77	2.84	5.08	3.00

fortably outperforms the RSS based algorithms. Although its performance is inferior to our SSD based algorithms, it performs slightly better than the HLF-based algorithms.

4.4 Summary and Conclusions

In this chapter, we introduced the use of *"Signal Strength Difference (SSD)"* as a location fingerprint, and analyzed in detail why it can serve as robust location fingerprint that is irrespective of the hardware used at the mobile device. From the analysis carried out in Section 4.1, we can draw the following conclusions:

- SSD is a more robust location fingerprint compared to traditional RSS regardless of whether the samples are collected at the APs or at the MN. This has been verified with our experimental results as illustrated thoroughly in Section 4.3.2 and 4.3.3.

- Collecting samples at the APs should provide more stable SSD readings compared to measuring them at the MN, because in the latter case, a greater number of assumptions were involved in inferring the SSD's expression. In addition, an

Table 4.2: Percentile values and averages of errors (in meter) when various fingerprints are considered for Wi-Fi.

Algorithm (Fingerprint)	25^{th} Percentile	Median	90^{th} Percentile	Average
KNN (RSS)	3.20	4.82	8.22	4.95
KNN (HLF)	1.73	2.81	5.84	3.08
KNN (SSD)	1.77	2.59	5.08	2.94
Bayes (RSS)	3.43	5.21	9.74	5.49
Bayes (HLF)	1.96	2.97	5.38	3.17
Bayes (SSD)	1.77	2.79	4.96	2.89
Ecolocation	4.30	6.99	11.00	6.88
Modified Ecolocation	1.94	2.83	5.34	3.06

AP-based approach tends to be free from vendor-specific shortcomings since the APs of a particular indoor environment are usually of the same type.

The shortcomings of the popular RSS location fingerprint with MN's hardware variations have been addressed in a few works [25,26,29] as discussed previously. Most fingerprinting solutions use the same mobile device for both training and testing, thereby, shielding the adverse effect of this phenomenon as revealed in our experimental results of Section 4.3.3. We also compare our SSD with two other robust location fingerprints in Section 4.3.4, and found the SSD based algorithms' performance to be superior in case of both Wi-Fi and Bluetooth. No work in the literature has been successful thus far in designing a reasonable Bluetooth based positioning system. We have shown that if inquiry based RSS is available and is used to generate SSD for use as a location fingerprint, a Bluetooth based positioning system with reasonable accuracy can still be achieved as we have demonstrated in our experimental results.

Chapter 5

Analysis of SSD

As emphasized in the previous section, we believe that the signal strength difference (SSD) is the ideal choice for location fingerprint if the positioning system administrator intends to accommodate heterogeneous devices. Therefore, the error bound on localization using SSD needs to be investigated. We feel that the properties of this bound could provide valuable insights to improving the localization accuracy or to the overall design of a positioning system based on SSD.

In this chapter, we analyze the Cramér-Rao Lower Bound (CRLB) [70] of location estimation error given the SSD measurements. A novel characterization of the properties of this bound is presented that allows us to individually assess the impact of different parameters (e.g., number of APs, geometry of the APs, distance of the APs from the MN, etc.) on the accuracy of location estimates. For example, utilizing the effect of distances of the APs from the MN, we have devised a way to define weights for a weighted $\mathcal{K}$-NN scheme that is shown to perform better than the $\mathcal{K}$-NN algorithm. Moreover, the properties also provide valuable design phase suggestions by revealing error trends associated with the system deployment. We also investigate these deployment issues which may give fruitful insights into the design of a positioning system.

The study of estimation bounds on localization using time-of-arrival [71], time-

difference-of-arrival [72], angle-of-arrival [72] techniques, or the RSS location fingerprint [73–75] have been investigated in the literature before. The findings subsequently opened the door for further analysis and design of various efficient localization algorithms which improve the accuracy [69, 72, 74]. We expect similar trend to follow for the SSD location fingerprint as well.

The rest of the chapter is organized as follows. In Section 5.1, we provide a brief review of our SSD location fingerprint, and discuss our localization algorithm in Section 5.2. The CRLB analysis and the impacts of various properties of this bound are presented in detail in Section 5.3. Finally, we present in Section 5.4 the summary and the conclusions drawn.

5.1 Review of SSD Location Fingerprint

The SSD fingerprint is shown to be robust across different mobile devices compared to the traditional RSS both analytically [27] and experimentally [25, 27]. Using the shadowing model, the SSD's expression can be obtained as in (4.6),

$$\left[\frac{p_k}{p_r}\right]_{\text{dB}} = -10\beta \log\left(\frac{d_k}{d_r}\right) + [X_k - X_r]_{\text{dB}}, \tag{5.1}$$

where β is the path-loss exponent, $X_k \sim N(0, \sigma_k^2)$ and $X_r \sim N(0, \sigma_r^2)$ are the shadowing variations, p_k and p_r denote the RSSs at the MN from the k^{th} and r^{th} APs, which are at distances d_k and d_r from the MN, respectively. Note that, p_k has been substituted for the notation $P(d_k)$ in this chapter to express the formulae in a simplistic way.

If a positioning system has K APs, there can be $\binom{K}{2}$ possible SSDs among which only $(K-1)$ values are independent. We obtain these SSD values as, $[\frac{p_k}{p_r}]_{\text{dB}}$, $k = \{1, 2, \ldots, K\} - \{r\}$, where the r^{th} AP is considered as the reference AP. The RSS of the r^{th} AP is subtracted from the other $(K-1)$ APs' RSS values to produce the desired

SSDs. An intuition about how we select the reference r^{th} AP is given at the end of next section. However, for ease of our mathematical calculations in this thesis, we assume the K^{th} AP to be the reference AP, i.e., $r = K$.

5.2 Localization Algorithm

We discussed the traditional RSS based Bayesian inference scheme to locate a user in Section 2.2.6. Our SSD-based localization algorithm is just a slight modification on that scheme. We consider our indoor environment to be modeled as a finite position space $\{c_1, c_2, \ldots, c_L\}$ with a finite observation space $\{o_1, o_2, \ldots, o_M\}$. We define an observation as a vector of signal strength readings over K APs, i.e., $o_j = \{p_1, p_2, \ldots, p_K\}$, where p_k denotes the received signal strength from the k^{th} AP.

Our SSD-based localization scheme is quite similar to the RSS-based algorithm. However, the observation vector is now a $(K-1)$-dimensional signal strength difference readings of the form, $o_j = \{p_1 - p_r, p_2 - p_r, \ldots, p_K - p_r\}$ where p_r denotes the reference r^{th} AP's RSS. Subsequently, the conditional probability of the observation becomes, $Pr(o_j|c_i) = \prod_{k=1}^{K-1} Pr(p_k - p_r|c_i)$. Unless mentioned otherwise, we adopt the $\mathcal{K}$-Nearest Neighbors as our algorithm in this chapter where the average of $\mathcal{K}$ locations having the largest $Pr(o_j|c_i)$'s gives the location estimate. Note that, this $\mathcal{K}$-NN algorithm is different from the $\mathcal{K}$-NN in signal space algorithm which was discussed in Section 2.2.5 and applied in experiments of Chapter 4.

Now, let us discuss how we have modeled the conditional probability $Pr(p_k - p_r|c_i)$. We assume the RSS from an AP at a particular location to be normally distributed. Though some works defy this phenomenon, others lend support to it [27, 30]. Similar to other works [53], we also have not observed any significant improvement when we consider the histogram representation of RSS compared to its Gaussian counterpart.

We post-process our training data to be fitted into the Gaussian distribution, $Z_{p_k|c_i} \sim N(\mu_{ik}, \sigma_{ik}^2)$, where μ_{ik} and σ_{ik} being the average and standard deviation of the signal strength samples collected from the k^{th} AP at training location c_i. Consequently, we obtain,

$$Z_{p_k - p_r|c_i} \sim N(\mu_{ik} - \mu_{ir}, \sigma_{ik}^2 + \sigma_{ir}^2). \tag{5.2}$$

We select the reference r^{th} AP as the one which shows the least average deviation of RSSs over the whole localization area, i.e., $r = \min_k \left\{ \frac{1}{L} \sum_{i=1}^{L} \sigma_{ik}^2 \right\}, \forall k \in \{1, 2, \ldots, K\}$.

5.3 CRLB for Localization using SSD

It is well-known that the Cramér-Rao Lower Bound (CRLB) sets a lower limit for the variance (or covariance matrix) of any unbiased estimates of an unknown parameter (or unknown parameters) [70]. If $\widehat{\theta} = (\widehat{x}\ \widehat{y})^T$ is the estimate of the MN's location $\theta = (x\ y)^T$, then its covariance matrix is,

$$\begin{aligned} \text{Cov}_\theta(\widehat{\theta}) &= E_\theta\{(\widehat{\theta} - \theta)(\widehat{\theta} - \theta)^T\} \\ &= \begin{bmatrix} \sigma_{\widehat{x}}^2 & \sigma_{\widehat{x}\widehat{y}} \\ \sigma_{\widehat{y}\widehat{x}} & \sigma_{\widehat{y}}^2 \end{bmatrix} \end{aligned} \tag{5.3}$$

where $E_\theta\{.\}$ is the expectation operator conditioned on θ. The diagonal elements of (5.3) represent the mean squared errors and the off-diagonal elements are the covariances between different parameters.

The lower bound is given in terms of the Fisher Information Matrix (FIM) [70]. If $f_\theta(P)$ denotes the probability density function (p.d.f.) of observations P conditioned

on θ, then the score function [70] is defined as the gradient of its log-likelihood, i.e.,

$$U(\theta) = \nabla \ln f_\theta(P) = \frac{\partial}{\partial \theta} \ln f_\theta(P). \tag{5.4}$$

The FIM, $J(\theta)$ is the variance of this score function,

$$J(\theta) = E\left\{ \left[\frac{\partial \ln f_\theta(P)}{\partial \theta} \right]^2 \right\}$$

If $f_\theta(P)$ belongs to some exponential family, then, with some regularity conditions imposed [70], we have,

$$J(\theta) = -E\left\{ \frac{\partial U(\theta)}{\partial \theta} \right\}. \tag{5.5}$$

The CRLB is just the inverse of FIM and from its property,

$$\text{Cov}_\theta(\widehat{\theta}) \geq \{J(\theta)\}^{-1} \tag{5.6}$$

Utilizing SSD's expression (5.1), the joint p.d.f. of the $(K-1)$ independent SSD measurements can be written as,

$$f_\theta(P) = \prod_{k=1}^{K-1} \frac{1}{\sqrt{2\pi}\widetilde{\sigma}} \frac{10}{\ln 10} \frac{p_r}{p_k} \times \exp\left\{ -\frac{\left[10 \log \frac{p_k}{p_r} + 10\beta \log\left(\frac{d_k}{d_r}\right) \right]^2}{2\widetilde{\sigma}^2} \right\}, \tag{5.7}$$

where $\tilde{\sigma}^2 = \sigma_k^2 + \sigma_r^2$, p_k and p_r are measured in mW scale, i.e., p_k(dBm) $= 10 \log p_k$ and $d\left[\frac{p_k}{p_r}\right]_{\text{dB}} = \frac{10}{\ln 10} \frac{p_r}{p_k} d\left[\frac{p_k}{p_r}\right]$. Let the FIM be denoted as,

$$J(\theta) = \begin{bmatrix} J_{xx}(\theta) & J_{xy}(\theta) \\ J_{yx}(\theta) & J_{yy}(\theta) \end{bmatrix}. \tag{5.8}$$

Using (5.7) into (5.4) and (5.5), the entries of (5.8) are obtained as,

$$\begin{aligned} J_{xx}(\theta) &= \rho \sum_{k=1}^{K-1} \left[\frac{\cos\phi_k}{d_k} - \frac{\cos\phi_r}{d_r} \right]^2. \\ J_{xy}(\theta) &= J_{yx}(\theta) \\ &= \rho \sum_{k=1}^{K-1} \left[\frac{\cos\phi_k}{d_k} - \frac{\cos\phi_r}{d_r} \right] \left[\frac{\sin\phi_k}{d_k} - \frac{\sin\phi_r}{d_r} \right]. \\ J_{yy}(\theta) &= \rho \sum_{k=1}^{K-1} \left[\frac{\sin\phi_k}{d_k} - \frac{\sin\phi_r}{d_r} \right]^2. \end{aligned}$$

Here, $\phi_k \in [0, 2\pi)$ is the angle the MN makes with respect to the k^{th} AP as illustrated in Fig. 5.1, and $\rho = (\frac{10\beta}{\widetilde{\sigma}\ln 10})^2$. If $\text{var}(\widehat{\theta})_K$ denotes the variance of our location estimate, then from the CRLB property (5.6), we have,

$$\text{var}(\widehat{\theta})_K \geq \frac{\lambda_K}{\rho \cdot \eta_K}, \tag{5.9}$$

where $\lambda_K = \sum_{k=1}^{K-1}(u_k^2 + v_k^2)$, $\eta_K = \sum_{k=1}^{K-1} u_k^2 \sum_{k=1}^{K-1} v_k^2 - \left\{\sum_{k=1}^{K-1} u_k v_k\right\}^2$, $u_k = [\frac{\cos\phi_k}{d_k} - \frac{\cos\phi_r}{d_r}]$ and $v_k = [\frac{\sin\phi_k}{d_k} - \frac{\sin\phi_r}{d_r}]$. The RHS of (5.9) specifies the CRLB of the MN's location estimate using SSD for a system having K APs, i.e.,

$$\mathcal{C}_K = \frac{\lambda_K}{\rho \cdot \eta_K} \tag{5.10}$$

The detailed calculation is presented in Appendix A.1. From (5.10), it is evident that the CRLB depends on i) the number of APs, K, ii) the geometry of the AP, ϕ_k, iii) propagation model parameters, $\widetilde{\sigma}$ and β, and iv) the distance of the AP from the MN, d_k. Next, we elaborately discuss the properties of these parameters on location estimation error bound. Using the findings, we provide insights into positioning system deployment issues, and also propose a modified $\mathcal{K}$-NN scheme that shows improvement over the $\mathcal{K}$-NN algorithm.

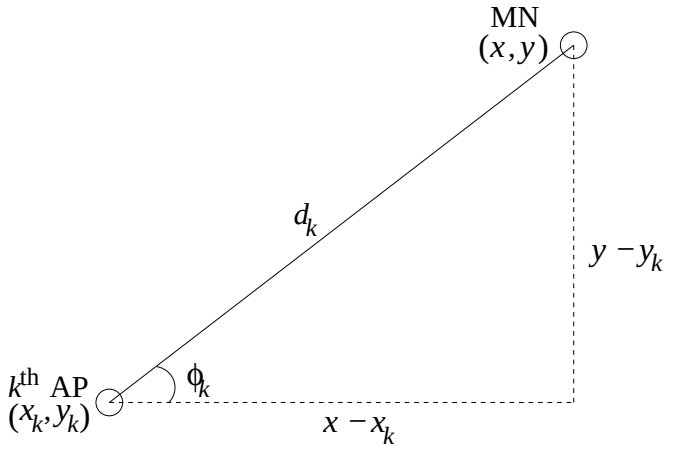


Figure 5.1: Definition of angle ϕ_k.

5.3.1 Impact of the Number of APs

Theorem 5.1. *The introduction of an additional AP with parameters* (d_{K+1}, ϕ_{K+1}) *results in the reduction of the CRLB except when* $\phi_{K+1} = \phi_r = \phi_k, \forall k \in \{1, 2, \ldots, K\} - \{r\}$.

Proof. When an additional $(K+1)^{\text{th}}$ AP is added to the system of K APs, the CRLB can be represented as,

$$\mathcal{C}_{K+1} = \frac{\lambda_{K+1}}{\rho \cdot \eta_{K+1}}, \tag{5.11}$$

where $\lambda_{K+1} = \lambda_K + (u_{K+1}^2 + v_{K+1}^2)$ and $\eta_{K+1} = \eta_K + (v_{K+1}^2 \sum_{k=1}^{K-1} u_k^2 - 2 \cdot u_{K+1} \cdot v_{K+1} \cdot \sum_{k=1}^{K-1} u_k v_k + u_{K+1}^2 \sum_{k=1}^{K-1} v_k^2)$. Our goal is to prove $\mathcal{C}_K - \mathcal{C}_{K+1} \geq 0$. Subtracting (5.11) from (5.10), we obtain,

$$\begin{aligned} \mathcal{C}_K - \mathcal{C}_{K+1} \;=\; & \left\{ v_{K+1}^2 \left(\lambda_K \sum_{k=1}^{K-1} u_k^2 - \eta_K \right) + u_{K+1}^2 \cdot \right. \\ & \left(\lambda_K \sum_{k=1}^{K-1} v_k^2 - \eta_K \right) - 2 u_{K+1} \cdot v_{K+1} \cdot \\ & \left. \lambda_K \sum_{k=1}^{K-1} u_k v_k \right\} / (\rho \cdot \eta_K \cdot \eta_{K+1}). \end{aligned} \tag{5.12}$$

The denominator of (5.12) is non-negative since $\rho > 0$ (see Eq. (A.2)) and $\eta_K, \eta_{K+1} \geq 0$ because of the following inequality,

$$\sum_{k=1}^{K} u_k^2 \sum_{k=1}^{K} v_k^2 - \left\{ \sum_{k=1}^{K} u_k v_k \right\}^2 \geq 0 \tag{5.13}$$

The induction proof of the above inequality is shown in Appendix A.2. To prove the non-negativity of the numerator of (5.12), we use the following inequality into (5.12) (see Appendix A.3 for proof),

$$\sqrt{\left(\lambda_K \sum_{k=1}^{K-1} u_k^2 - \eta_K\right)\left(\lambda_K \sum_{k=1}^{K-1} v_k^2 - \eta_K\right)} \geq \lambda_K \sum_{k=1}^{K-1} u_k v_k. \tag{5.14}$$

Subsequently, we obtain,

$$\begin{aligned}
\mathcal{C}_K - \mathcal{C}_{K+1} \;\geq\; & \left\{ v_{K+1}^2 \left(\lambda_K \sum_{k=1}^{K-1} u_k^2 - \eta_K\right) + u_{K+1}^2 \cdot \right. \\
& \left(\lambda_K \sum_{k=1}^{K-1} v_k^2 - \eta_K\right) - 2 \cdot u_{K+1} \cdot v_{K+1} \cdot \\
& \left. \sqrt{\left(\lambda_K \sum_{k=1}^{K-1} u_k^2 - \eta_K\right)\left(\lambda_K \sum_{k=1}^{K-1} v_k^2 - \eta_K\right)} \right\} \\
& / \left(\rho \cdot \eta_K \cdot \eta_{K+1}\right) \\
= \; & \left\{ v_{K+1} \sqrt{\left(\lambda_K \sum_{k=1}^{K-1} u_k^2 - \eta_K\right)} - u_{K+1} \cdot \right. \\
& \left. \sqrt{\left(\lambda_K \sum_{k=1}^{K-1} v_k^2 - \eta_K\right)} \right\}^2 / (\rho \cdot \eta_K \cdot \eta_{K+1}) \\
\geq \; & 0.
\end{aligned} \tag{5.15}$$

The case $\mathcal{C}_K - \mathcal{C}_{K+1} = 0$ arises only when both the equality conditions of Eq. (A.6) and (5.15) hold. These two equalities are satisfied only for the scenario, $\phi_{K+1} = \phi_r =$

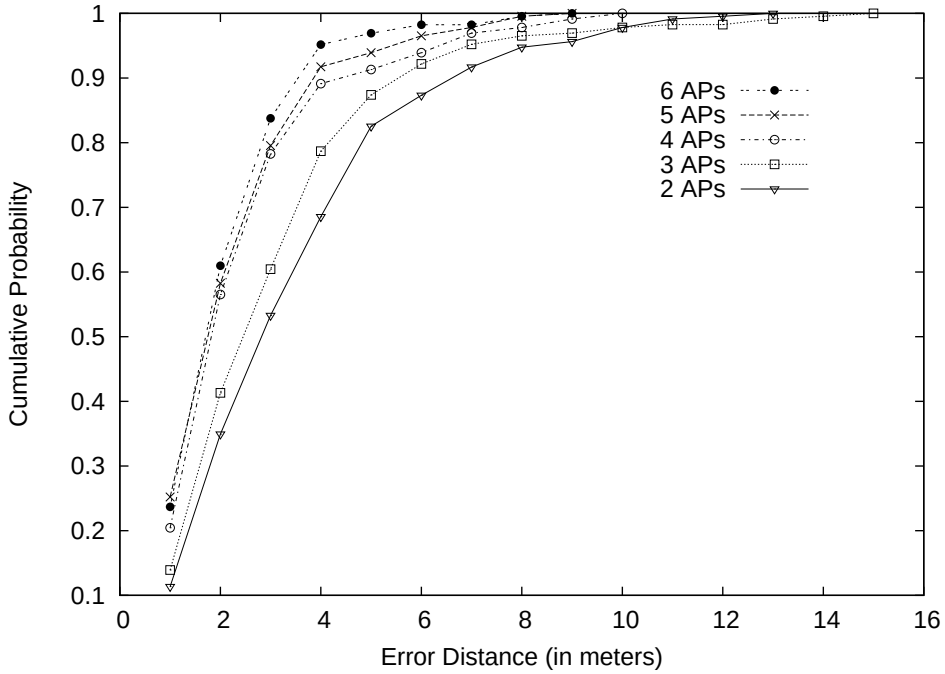


Figure 5.2: Localization accuracy improves with increasing number of APs.

ϕ_k, $\forall k \in \{1, 2, \ldots, K\} - \{r\}$, i.e., when all the APs are collinear (see Appendix A.4 for detailed calculation). Therefore, except for this situation, the introduction of an additional AP indeed lowers the CRLB of the location estimate. □

Fig. 5.2 shows the experimental results (conducted in our testbed of Fig. 5.3) of localization accuracy as we vary the number of APs. It shows monotonic increase in localization accuracy as the number of APs increases. This experimental result is completely in sync with the findings of our CRLB analysis. We have used $\mathcal{K}$-NN algorithm discussed in Section 5.2 as our location classifier. However, from the pattern recognition theory [57], it is well-known that the dimension of the feature vector (i.e., number of APs in our case) cannot be arbitrarily increased to achieve better accuracy. As a matter of fact, we have not seen any improvement in localization accuracy when the number of APs is greater than ten.

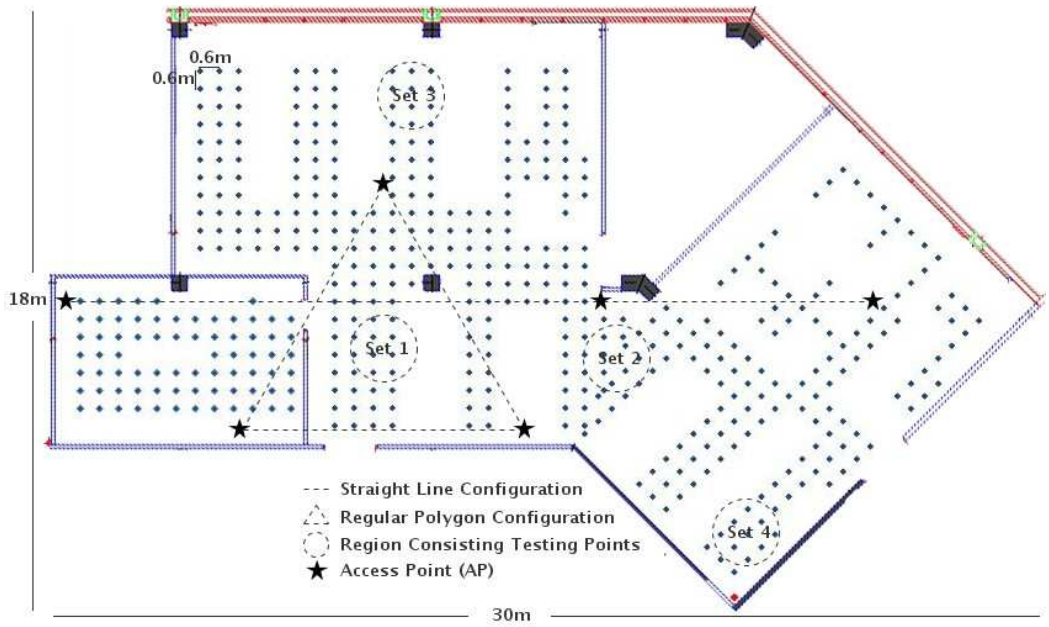


Figure 5.3: Two different configurations of three APs: i) Regular Polygon and ii) Straight Line. The four testing sets are indicated by the circular regions.

5.3.2 Impact of the Geometry of APs

When $\phi_k = \phi, \forall k \in \{1, 2, \ldots, K\}$, we see that the FIM (5.8) is singular. In other words, if all the APs lie on a straight line w.r.t. the MN, then the location estimation error bound (5.10) is the largest.

Optimal geometry occurs when the MN is situated at the center of a K-sided regular polygon where the vertices of the polygon indicate the positions of the K APs. In this setting, we have, $d_k = d, \forall k = \{1, 2, \ldots, K\}$, and $\sum_{k=1}^{K} \sin(h\phi_k) = \sum_{k=1}^{K} \cos(h\phi_k) = 0$, for any integer, $h \geq 1$. Let us assume $\phi_r = 0°$, then the FIM (5.8) takes the following form,

$$J(\theta) = \frac{\rho}{d^2} \begin{bmatrix} \frac{3K}{2} & 0 \\ 0 & \frac{K}{2} \end{bmatrix}. \tag{5.16}$$

Using the terms of (5.16), we obtain the lower bound of $\text{var}(\widehat{\theta})_K$ as $\frac{8d^2}{3\rho K}$, which suggests

that it decreases with increasing K. In other words, the lower bound for a square configuration of the APs ($K = 4$) would be smaller than that of an equilateral triangle configuration ($K = 3$). Moreover, the lower bound may also decrease for the same number of APs' setup where they are closer to the MN (i.e., d decreases) compared to the scenario when they are farther.

In Table 5.1, we list the average localization errors when the equilateral triangle configuration of the three APs is considered for four different testing sets (the circular regions in Fig. 5.3). The testing set near the centroid of the equilateral triangle (Set 1 in Fig. 5.3) performs the best, which complies with the analysis here. Therefore, the system administrator may place the APs in such a way that a subset of them creates regular polygon with "popular area" being the centroid of it. Here, we have investigated the optimal APs' geometry from the CRLB perspective using SSD as location fingerprint. Experimental finding on optimal placement of APs for localization with traditional RSS fingerprint can be found in [76, 77] which shows similar trend.

Table 5.2 shows the average localization errors for the optimal and worst-case (collinear) configurations of the APs for a particular testing set (Set 1 of Fig. 5.3), which also supports our analysis. However, the collinear configuration of the APs is found to be better for signal coverage [77]. Since both data communication and providing location service would be the responsibilities of a Wi-Fi infrastructure, the system administrator needs to take these conflicting requirements into account during deployment.

5.3.3 Impact of the Propagation Model Parameters

From (5.10) and (A.2), it is evident that, if the shadowing variance of SSD, $\widetilde{\sigma}$ increases, then CRLB also increases. A lower value of path-loss exponent β increases the CRLB as well. As a result, the location estimation error bound (5.10) would be larger in both

Table 5.1: Average localization errors when different testing sets are used for optimal configuration of the three APs (the equilateral triangle in Fig. 5.3)

Testing Points' Set	Set 1	Set 2	Set 3	Set 4
Average Error (in meter)	2.95	3.31	3.33	3.71

Table 5.2: Average localization errors when the optimal (regular polygon) and worst-case (straight line) configurations of the three APs are used

Configuration	Average Error (in meter)
Regular Polygon	2.95
Straight Line	4.02

cases. On the contrary, when the RSS shows less fluctuations (i.e., $\widetilde{\sigma} \downarrow$), the location fingerprints tend to be more consistent and are likely to produce better localization accuracy. Also, the shadowing model of RF propagation performs better indoors (larger β) compared to outdoor environments. Similar effects of these parameters are observed on accuracy for RSS based localization as well [19, 69].

5.3.4 Impact of the Distance of an AP from the MN

In this section, we first analyze the impact of APs' distances on localization accuracy from the CRLB perspective. Subsequently, we utilize the finding of this analysis to define *weights* for the weighted least squares (WLS) approach which is shown to improve localization accuracy both analytically and experimentally. Finally, we take all these findings into account in order to modify our $\mathcal{K}$-NN algorithm in Section 5.2 that reduces the localization error further.

Under optimal geometric configuration of the APs, it has been seen in Section 5.3.2 that, the setup with APs closer to the MN yields a reduction in the CRLB compared to the setting where the APs are farther. Now, let us investigate the effect of the distance of the MN from an AP in a more generic scenario.

Theorem 5.2. *The decrease in any of the distances d_k's of the APs from the MN results in reduction of the CRLB. Specifically, if $d'_k = \alpha d_k,\ 0 < \alpha \leq 1$, the new CRLB satisfies,*

$$\mathcal{C}_{K'} \leq \mathcal{C}_K \tag{5.17}$$

Proof. Suppose the reference r^{th} AP is the farthest from the MN, and for the ease of our calculation, we assume $d_r \to \infty$. From (5.10), we have, $\mathcal{C}_K = \frac{\lambda_K}{\rho \cdot \eta_K}$, where $\lambda_K = \lambda_{K-1} + \frac{1}{d_K^2}$ and $\eta_K = \eta_{K-1} + v_K^2 \sum_{k=1}^{K-2} u_k^2 + u_K^2 \sum_{k=1}^{K-2} v_k^2 + 2 \cdot u_K \cdot v_K \cdot \{\sum_{k=1}^{K-2} u_k v_k\}$. Because of our assumption, the identities u_k and v_k take the following forms, $u_k = \frac{\cos\phi_k}{d_k}$ and $v_k = \frac{\sin\phi_k}{d_k}$. Similarly, the new CRLB can be written as, $\mathcal{C}_{K'} = \frac{\lambda_{K'}}{\rho \cdot \eta_{K'}}$, where $\lambda_{K'} = \lambda_{K-1} + \frac{1}{\alpha^2 d_K^2}$ and $\eta_{K'} = \eta_{K-1} + v_{K'}^2 \sum_{k=1}^{K-2} u_k^2 + u_{K'}^2 \sum_{k=1}^{K-2} v_k^2 + 2 \cdot u_{K'} \cdot v_{K'} \cdot \{\sum_{k=1}^{K-2} u_k v_k\}$ and the identities, $u_{K'} = \frac{\cos\phi_K}{\alpha \cdot d_K}$ and $v_{K'} = \frac{\sin\phi_K}{\alpha \cdot d_K}$.

Our goal is to prove $\mathcal{C}_K - \mathcal{C}_{K'} \geq 0$. Using the CRLB expressions, we have,

$$\mathcal{C}_K - \mathcal{C}_{K'} = \frac{\lambda_K \eta_{K'} - \lambda_{K'} \eta_K}{\rho \cdot \eta_K \cdot \eta_{K'}}. \tag{5.18}$$

The denominator of (5.18), $\rho \cdot \eta_K \cdot \eta_{K'} \geq 0$. Now, we have to prove that the numerator of (5.18) is non-negative. The numerator can be simplified as,

$$\begin{aligned} & \lambda_K \eta_{K'} - \lambda_{K'} \eta_K \\ = \ & (v_{K'}^2 - v_K^2) \left(\lambda_{K-1} \sum_{k=1}^{K-2} u_k^2 - \eta_{K-1} \right) \\ & + (u_{K'}^2 - u_K^2) \left(\lambda_{K-1} \sum_{k=1}^{K-2} v_k^2 - \eta_{K-1} \right) \\ & - 2\lambda_{K-1} (u_{K'} v_{K'} - u_K v_K) \left\{ \sum_{k=1}^{K-2} u_k v_k \right\}. \end{aligned} \tag{5.19}$$

Using the following inequality,

$$\sqrt{(u_{K'}^2 - u_K^2)(v_{K'}^2 - v_K^2)\left(\lambda_{K-1}\sum_{k=1}^{K-2} v_k^2 - \eta_{K-1}\right)\left(\lambda_{K-1}\sum_{k=1}^{K-2} u_k^2 - \eta_{K-1}\right)} \geq \lambda_{K-1}(u_{K'}v_{K'} - u_K v_K)\left\{\sum_{k=1}^{K-2} u_k v_k\right\} \quad (5.20)$$

in (5.19) (the proof is shown in Appendix A.5), we obtain,

$$\begin{aligned} \mathcal{C}_K - \mathcal{C}_{K'} \;\geq\; & \left\{\sqrt{(v_{K'}^2 - v_K^2)\left(\lambda_{K-1}\sum_{k=1}^{K-2} u_k^2 - \eta_{K-1}\right)}\right. \\ & \left. -\sqrt{(u_{K'}^2 - u_K^2)\left(\lambda_{K-1}\sum_{k=1}^{K-2} v_k^2 - \eta_{K-1}\right)}\right\}^2 \\ & / (\rho \cdot \eta_K \cdot \eta_{K'}) \\ \geq\; & 0. \end{aligned}$$

The case $\mathcal{C}_K - \mathcal{C}_{K'} = 0$ arises when $\alpha = 1$ (i.e., $d_{K'} = d_K$) as the numerator (5.19) becomes zero. □

5.3.4.1 Weighted Least Squares (WLS) Approach

The CRLB given by (5.10) provides a benchmark for comparing the performance of location-estimation, but does not explicitly describe the estimator that achieves it [70]. In this section, using the impact of a closer AP in defining *weights*, we emerge with a WLS estimator that is shown to perform better than LS estimator both analytically and experimentally.

Let the coordinates of the $(K+1)$ APs, $\theta_k = [x_k\ y_k]^T$, $k = 1, 2, \ldots, (K+1)$ be known, and the MN's position is $\theta = [x\ y]^T$. The distances d_k's between the MN and the K APs (excluding the reference r^{th} AP) are calculated by means of linear RF

propagation modeling using (5.1) as, $\left[\frac{p_k}{p_r}\right]_{\text{dB}} = a \log d_k + b$, where $a = -10\beta$ and we assume $b = 10\beta \log d_r + [X_k - X_r]_{\text{dB}}$. Therefore, we have K equations of the form,

$$\| \theta - \theta_k \|^2 = \| \theta \|^2 + \| \theta_k \|^2 - 2\theta_k^T \theta = d_k^2, \ k = 1, 2, \ldots, K.$$

Taking the difference of each of the above K equations, this system transforms into a set of $\binom{K}{2}$ linear equations,

$$2(\theta_k - \theta_l)^T \theta = (\| \theta_k \|^2 - \| \theta_l \|^2) - (d_k^2 - d_l^2).$$

The above set of linear equations is of the form $y = \mathcal{X}\theta$, where y is a column vector of length $\binom{K}{2}$ whose components are from $(\| \theta_k \|^2 - \| \theta_l \|^2) - (d_k^2 - d_l^2)$, and $\mathcal{X}$ is a $\binom{K}{2} \times 2$ matrix whose rows are from the set $\{2(\theta_k - \theta_l)^T\}$, $k = 1, 2 \ldots, K-1$, $l > k$. The observations can be represented as, $y = \mathcal{X}\theta + \mathcal{N}$, where $\mathcal{N}$ is a zero-mean noise column vector of length $\binom{K}{2}$. The general weighted quadratic risk function is of the form,

$$R(\theta)_{\text{WLS}} = \left[(y - \mathcal{X}\theta)^T \mathcal{W}^T (y - \mathcal{X}\theta)\right] \tag{5.21}$$

where $\mathcal{W}$ is a $\binom{K}{2} \times \binom{K}{2}$ symmetric weight matrix. The WLS estimate of the node's location is given by [78],

$$\widehat{\theta} = (\mathcal{X}^T \mathcal{W} \mathcal{X})^{-1} \mathcal{X}^T \mathcal{W} y, \tag{5.22}$$

and the covariance matrix [78],

$$\Psi_{\text{WLS}} = (\mathcal{X}^T \mathcal{W} \mathcal{X})^{-1} \mathcal{X}^T \mathcal{W} \Lambda \mathcal{W} \mathcal{X} (\mathcal{X}^T \mathcal{W} \mathcal{X})^{-1}. \tag{5.23}$$

When the observation errors are uncorrelated, the weight matrix, $\mathcal{W}$, is diagonal. The resulting estimator is the *best linear unbiased estimator* (BLUE) if the weight matrix is equal to the inverse of the covariance matrix of the observation errors [79], i.e., $\mathcal{W} =$

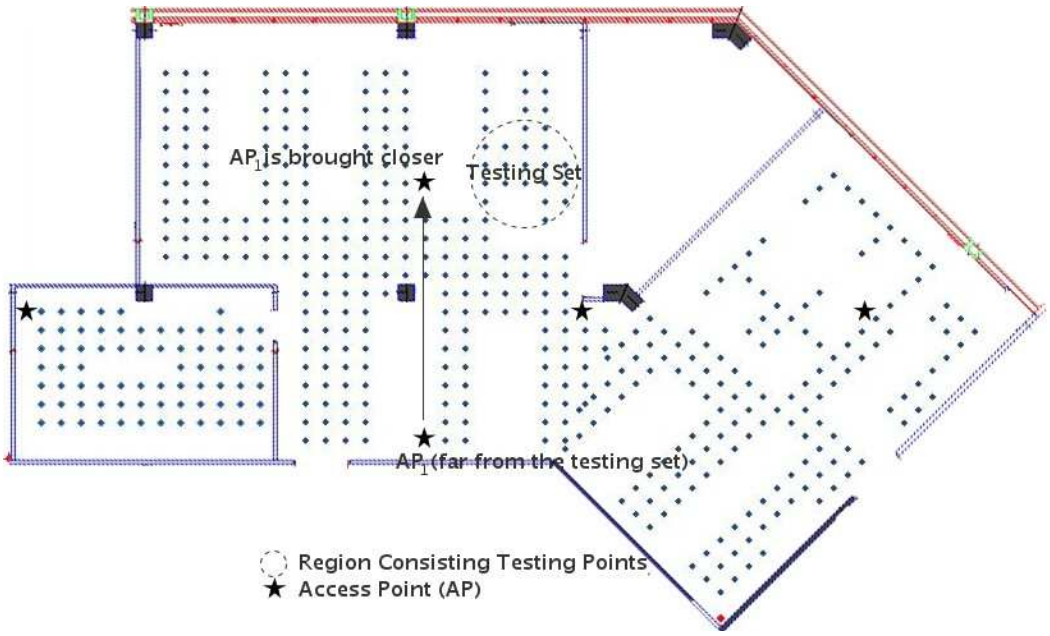


Figure 5.4: From a distant position, AP_1 is brought closer to the testing set which is indicated by the circular region. The other APs' positions are collinear.

Λ^{-1}, where Λ is the covariance matrix of the observation errors. Using this identity in (5.23), the best linear unbiased estimator's covariance matrix can be represented as, $\Psi_{\text{BLUE}} = (\mathcal{X}^T \Lambda^{-1} \mathcal{X})^{-1}$. Using matrix algebra, it can be shown that for any other choice of the weight matrix $\mathcal{W}$ [79], $\Psi_{\text{BLUE}} \leq \Psi_{\text{WLS}}$.

Let us consider the scenario of Fig. 5.4 where AP_1 is first stationed far from the testing set. Suppose $\mathcal{W}$ is defined as the inverse of the covariance matrix of the observation errors:

$$\mathcal{W} = \Lambda^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2+\sigma_2^2} & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & \frac{1}{\sigma_1^2+\sigma_K^2} & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & \frac{1}{\sigma_{K-1}^2+\sigma_K^2} \end{bmatrix} \tag{5.24}$$

Consequently, the matrix $\mathcal{X}^T\Lambda^{-1}\mathcal{X}$ can be evaluated as the following,

$$\mathcal{X}^T\Lambda^{-1}\mathcal{X} = \begin{bmatrix} \sum\limits_{\substack{k=1\\k<l\leq K}}^{K-1} \frac{(x_k-x_l)^2}{\sigma_k^2+\sigma_l^2} & \sum\limits_{\substack{k=1\\k<l\leq K}}^{K-1} \frac{(x_k-x_l)(y_k-y_l)}{\sigma_k^2+\sigma_l^2} \\ \sum\limits_{\substack{k=1\\k<l\leq K}}^{K-1} \frac{(x_k-x_l)(y_k-y_l)}{\sigma_k^2+\sigma_l^2} & \sum\limits_{\substack{k=1\\k<l\leq K}}^{K-1} \frac{(y_k-y_l)^2}{\sigma_k^2+\sigma_l^2} \end{bmatrix} \tag{5.25}$$

Now, suppose AP_1 is brought closer to the testing set (i.e., MN) as shown in Fig. 5.4. Let us denote this new distance as d_1' and the corresponding observation error variance and covariance matrix as ${\sigma_1'}^2$ and Λ', respectively. $\mathcal{W}$ is expected to vary due to the observation in localization literature [17] that, the measurement errors associated with the closer d_1' calculation are generally less error-prone than that of the distant d_1 case, i.e., ${\sigma_1'}^2 \leq {\sigma_1}^2$. For ease of calculation, let us assume that the new (closer) position of AP_1 and its old (farther) position are symmetric to each other with respect to the straight line formed by the other APs. So, the new matrix $\mathcal{X}^T{\Lambda'}^{-1}\mathcal{X}$ takes the form,

$$\mathcal{X}^T{\Lambda'}^{-1}\mathcal{X} = \begin{cases} \left(\mathcal{X}^T{\Lambda'}^{-1}\mathcal{X}\right)_{11} = \sum\limits_{\substack{k=2\\k<l\leq K}}^{K-1} \frac{(x_k-x_l)^2}{\sigma_k^2+\sigma_l^2} + \sum\limits_{l=2}^{K} \frac{(x_1-x_l)^2}{{\sigma_1'}^2+\sigma_l^2} \\ \left(\mathcal{X}^T{\Lambda'}^{-1}\mathcal{X}\right)_{12} = \sum\limits_{\substack{k=2\\k<l\leq K}}^{K-1} \frac{(x_k-x_l)(y_k-y_l)}{\sigma_k^2+\sigma_l^2} + \sum\limits_{l=2}^{K} \frac{(x_1'-x_l)(y_1'-y_l)}{{\sigma_1'}^2+\sigma_l^2} \\ \left(\mathcal{X}^T{\Lambda'}^{-1}\mathcal{X}\right)_{21} = \sum\limits_{\substack{k=2\\k<l\leq K}}^{K-1} \frac{(x_k-x_l)(y_k-y_l)}{\sigma_k^2+\sigma_l^2} + \sum\limits_{l=2}^{K} \frac{(x_1'-x_l)(y_1'-y_l)}{{\sigma_1'}^2+\sigma_l^2} \\ \left(\mathcal{X}^T{\Lambda'}^{-1}\mathcal{X}\right)_{22} = \sum\limits_{\substack{k=2\\k<l\leq K}}^{K-1} \frac{(y_k-y_l)^2}{\sigma_k^2+\sigma_l^2} + \sum\limits_{l=2}^{K} \frac{(y_1-y_l)^2}{{\sigma_1'}^2+\sigma_l^2} \end{cases} \tag{5.26}$$

Using (5.25) and (5.26) and the identity ${\sigma_1'}^2 \leq \sigma_1^2$, it can be easily seen that, the matrix $(\mathcal{X}^T{\Lambda'}^{-1}\mathcal{X} - \mathcal{X}^T\Lambda^{-1}\mathcal{X})$ is positive semidefinite. Therefore, we obtain, $\mathcal{X}^T{\Lambda'}^{-1}\mathcal{X} \geq \mathcal{X}^T\Lambda^{-1}\mathcal{X}$ which in turn yields $\left(\mathcal{X}^T{\Lambda'}^{-1}\mathcal{X}\right)^{-1} \leq \left(\mathcal{X}^T\Lambda^{-1}\mathcal{X}\right)^{-1}$, i.e.,

$\Psi'_{\text{BLUE}} \leq \Psi_{\text{BLUE}}$. In other words, considering an AP closer to the MN rather than a distant one produces a smaller covariance matrix, thereby, resulting in a *better* estimator. Fig. 5.5 shows our experimental results which strengthen this fact. We see that, the localization accuracy improves for both LS and WLS approaches when a near AP is considered in place of a distant one. Moreover, WLS's performance is better than the normal LS approach. We conclude this section with some remarks:

Remark 1. *The least squares' (both LS and WLS) results in Fig. 5.5 are worse than the $\mathcal{K}$-NN. We attribute this inferior performance to the simple linear RF propagation model we have used in LS approaches to infer the distances d_k's.*

Remark 2. *The measurement errors in $\mathcal{W}$ (5.24) are calculated as deviations of the real and predicted distances between the MN and the APs. In doing so, we have assumed the real distances between the MN and the APs to be known which is usually not true in practice. Our modified $\mathcal{K}$-NN algorithm (described in the next section) does not suffer from this issue.*

5.3.4.2 Modification to the $\mathcal{K}$-NN Algorithm

It has been seen in Theorem 5.2 that the CRLB is reduced when we consider a closer AP compared to a distant one. We also applied this property to a well-known estimator algorithm (WLS) in the previous section, and show that its accuracy improves, both analytically and experimentally.

Utilizing the above two observations, we modified our $\mathcal{K}$-NN algorithm presented in Section 5.2 into a weighted $\mathcal{K}$-NN scheme where the conditional probability is now changed to $Pr(o_j|c_i) = \prod_{k=1}^{K-1} \left\{ \left[\frac{1}{d_{ki}^{\gamma}} \cdot Pr(p_k - p_r|c_i) \right] / \left(\sum_{i=1}^{L} \frac{1}{d_{ki}^{\gamma}} \right) \right\}$. Here, L is the number of training locations and γ ($\gamma \geq 0$) is the weight exponent. The weight is chosen in such a way that the closer APs are given more importance. We have chosen γ empirically as 2. Note that, $\gamma = 0$ transforms our weighted $\mathcal{K}$-NN into the normal

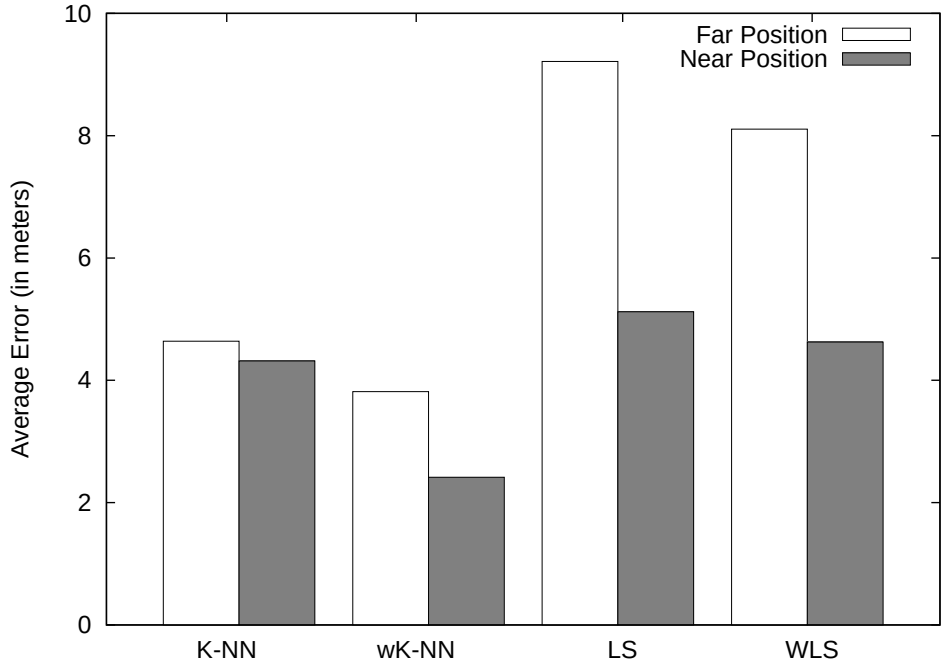


Figure 5.5: Average localization errors of four different algorithms for two different placements of AP_1 (near vs. far as shown in Fig. 5.4). The testing set is indicated by the circular region in Fig. 5.4.

$\mathcal{K}$-NN algorithm. The rest of the algorithm is exactly similar to the one described in Section 5.2.

Fig. 5.5 shows that the weighted $\mathcal{K}$-NN (w$\mathcal{K}$-NN) algorithm's performance is better than the $\mathcal{K}$-NN (see the first two pairs of bars from the left). Moreover, the localization accuracy of the w$\mathcal{K}$-NN is further improved when one of the APs (AP_1) can be placed closer to the testing set.

5.4 Summary and Conclusions

In this chapter, we studied the CRLB of localization using SSD as location fingerprint. We investigated the effects of various parameters (e.g., quantity and geometry of the APs, the distances of the APs from the MN, etc.) of this bound on localization error. Subsequently, we utilize one of the findings of our analysis, which reveals that a closer

AP to the MN is more *important* for localization purpose compared to a distant one, in order to define weights for a w$\mathcal{K}$-NN algorithm. Moreover, we also provide design phase suggestions for various geometric configurations and quantities of the APs by revealing error trends associated with them. In short, our analysis and experimental results provide valuable insights into the localization performance and deployment issues of a positioning system based on SSD. We expect SSD to be the preferred choice of location fingerprint over RSS if a positioning system aims to provide services to devices with heterogeneous hardware solutions. Our analysis and findings in this chapter should open the door for further analysis and designing of various efficient localization algorithms where SSD is chosen as the location fingerprint.

Chapter 6

Shorter Training Phase

As discussed previously, majority of indoor localization techniques that rely on in-building communications infrastructure (e.g., Wi-Fi, Bluetooth, etc.) utilize location fingerprinting techniques [9,17,18,20,25–27,30], where some location-dependent signal parameters are collected at a number of locations as location fingerprints in an offline training phase. During the online location estimation phase, the signal parameter obtained is *compared* with those training data to estimate the user location. The procedure of creating the training database of signal parameters entails a laborious offline phase because the location system administrator needs to take readings at every selected location of interest. Moreover, if for any unforeseeable reason, the setup changes (e.g., due to renovation, rearrangement of furniture, etc.), the whole training phase needs to be repeated again in the changed environment. The need for an exhaustive training phase can certainly limit the mass deployment of a fingerprint based positioning system. Moreover, the accuracy offered by a system which is under-trained may not satisfy the requirement of various location dependent applications.

In this chapter, we propose two ideas that try to relieve/shorten the exhaustive training phase typically seen in the fingerprinting techniques. In Section 6.1, a simple linear regression interpolation technique is applied to facilitate under-trained posi-

tioning systems. We investigate the possibility of building a positioning system from scratch or fine-tuning an under-trained positioning system through user participation in Section 6.2.

6.1 Interpolation Technique

As mentioned earlier, one of the key objectives of fingerprint-based positioning systems is to shorten their training phase. In this section, we show that samples collected at the APs considering only a few training locations could be sufficient for achieving reasonable localization accuracy with the help of proper interpolation techniques.

6.1.1 Fictitious Training Points

A few prior works in the literature [26,80] have attempted to shorten the training phase of a positioning system. They contend that, rather than performing an exhaustive survey to create a location fingerprint database that requires substantial cost and labor, one could simply collect a limited number of readings. Haebarlen *et al.* [26] achieve this goal by dividing the whole area into rooms/cells, thereby limiting the location estimates to room-level granularity. On the contrary, Li *et al.* [80] try to complete the database using interpolation of readings taken at other training points. Our work has adopted the latter approach. We hold the view that an interpolation-based training approach may stand out when the environment or setup changes. Normally, in such scenarios, the location services may have to be suspended, while waiting for the creation of an appropriate location fingerprint database that models the change. This procedure is both time and labor intensive, and the service downtime might be long. On the contrary, the positioning system administrator may choose to continue location service provisioning by performing a rough survey (i.e., taking a few samples) in the changed environment or setup, and fill up the voids in the training database with the

help of interpolation-based techniques. The database may then be augmented incrementally by taking more samples until the positioning system achieves a reasonable accuracy. Li *et al.* [80] have only used some intuitive guidelines to generate these *fictitious* training points. In this thesis, we have used weighted linear regression in order to obtain a better fit for those *fictitious* training points, by exploiting the spatial similarity [30] of signal strength distribution.

Using the path-loss model without the shadowing variation of (4.1), we can write,

$$\overline{P(d)}|_{\text{dBm}} = -10\beta \log(d) + P(d_0)|_{\text{dBm}} + 10\beta \log(d_0), \tag{6.1}$$

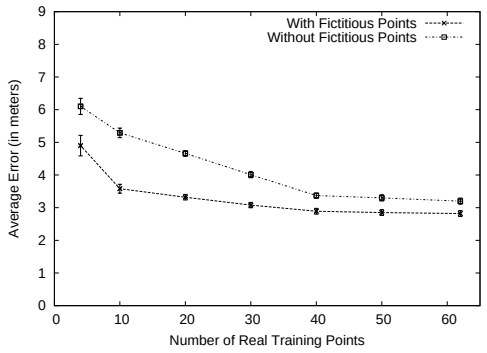
where $\overline{P(d)}|_{\text{dBm}}$ denotes the average RSS perceived at a distance d from the transmitter and the other symbols have usual meanings as discussed in Section 4.1. Based on (6.1), we fit a linear regression model of the type $y = ax + b$, to predict the RSS perceived at an AP, where $y = \overline{P(d)}|_{\text{dBm}},\ a = -10\beta,\ x = \log(d)$, and $b = P(d_0)|_{\text{dBm}} + 10\beta \log(d_0)$.

We denote *fictitious training points* as those training points in the database that are generated using interpolation from the actual training sample sets. In order to deduce a fictitious training point j, each AP's RSS is formulated according to (6.1), exploiting the signal strength values collected at the APs during the training phase. For example, if there are K APs, K different regression equations will be formed in order to deduce a single fictitious training point's fingerprint. The unknown parameters, i.e., a and b for each AP, are approximated using *weighted least mean square* method applied on the training data. Our target is to minimize $\sum_i w_{ji}(y_i - \hat{y}_i)^2$ where y_i and $\hat{y}_i$ represent the actual and predicted signal strengths, respectively, at a particular AP for the i^{th} real training point. The weights, w_{ji}'s, are assigned utilizing the spatial similarity of signal strength distribution reported in localization literature [30] which basically states that the RSSs observed at neighboring locations tend to exhibit similar properties. In our experiments, we have chosen the weight to be inversely proportional

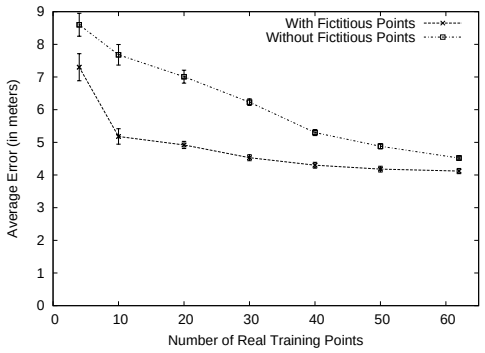
to the distance between a certain fictitious point j and the actual training point i (i.e., $\frac{1}{d_{ji}}$). Consequently, for each fictitious point, the closer real training points contribute more heavily in formulating the APs' regression equations, complying with the spatial similarity property mentioned earlier. Once we have approximated the signal patterns of the APs regarding a certain fictitious point (i.e., evaluate the values of a's and b's of their linear regression formulae), we would just plug in the distance of that particular fictitious point from the corresponding APs in order to obtain its fingerprint. Note that, each fictitious point has a different set of regression coefficients for the K APs' signal patterns. This is due to the fact that the weights, w_{ji}'s, that are associated with the actual training samples in the minimization formula above are different for each fictitious point j. The detailed calculation of a fictitious training point's fingerprint is presented in Appendix B.

6.1.2 Experimental Results

In the previous section, it was argued that the application of proper interpolation techniques could enhance an under-trained positioning system's accuracy immensely. In that regard, our simple linear regression based method performs very well as manifested in Fig. 6.1(a) and 6.1(b). This experiment is conducted inside our Testbed 1 which offers both Wi-Fi and Bluetooth functionality. As can be seen, when fictitious training points are utilized, the use of merely 20 real training points are sufficient to achieve the same level of accuracy as a positioning system with as many as 62 training points. For producing Fig. 6.1(a) and 6.1(b), a fixed number of fictitious points (≈ 60) scattered over the testbed uniformly, are introduced regardless of the number of real training points. The real training points are selected randomly from the $(62+44) = 106$ data points, and then the 44 testing samples are chosen randomly from the rest. For each combination (i.e., 4, 10, 20, etc.) of real training points, we repeat this proce-



(a) Bayesian algorithm's performance with and without fictitious points (Bluetooth).



(b) Bayesian algorithm's performance with and without fictitious points (Wi-Fi).

Figure 6.1: Bayesian algorithms' performance corresponding to varying number of real training locations for Wi-Fi and Bluetooth.

dure for 101 times to obtain the averages with 95% confidence interval as shown in Fig. 6.1(a) and 6.1(b). The inferior performance of Wi-Fi compared to Bluetooth can be attributed to the fact that the number of samples collected at each location for Wi-Fi using our own program was quite insignificant for Testbed 1. Therefore, we might not have been able to capture the Wi-Fi signal characteristics accurately.

6.2 User Feedback based Positioning System

In this section, we propose an idea where the end-users can actually contribute to the construction of a positioning system incrementally, as well as the fine-tuning of an under-trained system. We define *user feedback* as the information about a user's actual position as indicated by the user to the system, either explicitly or implicitly.

We claim that the contribution of users' feedback to any positioning system is two-fold. Firstly, user feedbacks greatly help in fine-tuning an under-trained positioning system with proper filtering of the malicious feedbacks. Secondly, if users are *well-behaved*, our experimental results show that the participation of end-users can actually assist in the construction of a positioning system incrementally from scratch. UCSD's

ActiveCampus project also tries to solve the indoor localization problem with the help of user feedbacks [81]. They utilize the corrections made by users on their estimated positions similar to us. However, their interpretation of the user's correction is simply a location and its received signal strength (RSS) signature pair, similar to a traditional training sample, which is completely different from how we interpret user feedback.

We contend that the combination of user feedback together with interpolation methods could eliminate the need for an exhaustive training phase, as the need for signal strength survey by administrators has been the key obstacle for the mass deployment of fingerprint based indoor positioning system. Our system can be particularly beneficial for large area deployment where it is quite demanding on the system administrator's part to visit all the possible areas and tirelessly perform the training phase. A user's feedback may not always truly reflect his/her actual location either due to the user's carelessness while giving feedback or deliberate ill intentions. Therefore, we define a Region of Confidence (RoC) with each estimated position to provide a measure of likelihood of a user's position, which is not just useful to the user when they give feedback; but also helps to assign *credibility* to each individual feedback in order to aid its incorporation into our system.

The important issue of adapting the positioning system seamlessly when its surroundings change (without performing the entire training phase all over again), has been overlooked in most fingerprint-based localization research. In our work, we emphasize that, a positioning system that exploits user feedbacks would guarantee reasonable performance over a longer period even if its surroundings change. This is crucial as the environments in a real system could constantly change, and it will be very difficult and demanding if system administrators need to monitor such changes and having to perform the signal strength survey all over again every time it changes. Apart from the above novelties, we have also denoted the signal strength signature of a user feedback in an efficient way and proved it analytically. In the following, we

summarize our objectives:

- We try to relieve the exhaustive training phase of a traditional fingerprint-based positioning system through user participation in both explicit and implicit way.
- We show that user feedback can greatly help in fine-tuning an under-trained positioning system which is already in operation. Moreover, under certain assumptions on user behavior and with the help of our interpolation method, we show that a positioning system solely based on user feedback could be built from scratch.
- We also show that, with the help of user feedback, changes in surroundings could be detected, allowing the system to adapt to the new environment in a seamless manner.

We hold the view that, there are many factors that could motivate a user to input feedbacks to the system. There can also be some indirect ways to obtain user feedbacks too. We list a few in the following:

- In an indoor scenario, a user may know where he/she is at present, but he/she may wish to obtain the route to another place within the same building from there. By inputting a more accurate starting point than what the system suggests, he/she can obtain a more refined route from the system.
- In a commercial system, a user who volunteers to provide feedback in an area he/she is familiar with, may earn credits for using the positioning service in an unfamiliar area later on.
- In a “location-based” social networking environment, giving feedback may facilitate the system to reveal nearby friends and places of interest more accurately.

- User trails as in [82] could be utilized to provide implicit feedback. In [82], the user trail is recorded as an ordered sequence of landmarks (e.g., access points (APs), card readers, etc.) where he/she has visited. To formulate a user feedback from trails, we could collect the signal strength samples of a user device between the user's visit to two successive landmarks. Since the start and end positions of the user are known (i.e., the two landmarks' positions), the intermediate locations could be interpolated by applying some assumptions on the user movement (e.g., constant speed). Subsequently, these interpolated locations could be correlated with the signal strength samples collected and treated as user feedback.

- Various landmarks (e.g., APs, tags, card readers, etc.) installed at several fixed positions in the building could act as continuous sources of implicit feedback as well.

6.2.1 User Feedback Model

As previously mentioned, user feedback is the information about a user's actual position as indicated by the user to the system either explicitly or implicitly. In this section, we discuss how the user feedback is visualized from a positioning system's point of view. Whenever a user inputs feedback to the system, it is interpreted as, $\mathcal{F} = (\mathcal{L}, \mathcal{S}, w)$, where

$\mathcal{L}$ = $[x\ y]^T$ = the position indicated by a user,

$\mathcal{S}$ = $[\bar{S}_1\ \bar{S}_2\ \ldots\ \bar{S}_K]^T$ = the RSS signature of the feedback captured at the K APs,

w = the degree at which a system believes the feedback, i.e., the *credibility* or *weight* of each individual feedback.

Next, we elaborately discuss all three components of a user feedback in Section 6.2.1.1, 6.2.1.2 and 6.2.1.3, respectively.

6.2.1.1 Location Indicated by User, $\mathcal{L}$

While giving a feedback, the user indicates his/her actual position. This location information is interpreted by the system as Cartesian coordinates (i.e., $[x\ y]^T$) in an indoor environment. In practice, there might be some uncertainties involved when a user tries to indicate his/her actual position at the time of providing feedback to the positioning system. These uncertainties might arise owing to the carelessness on the user's part while pinpointing his/her location on the map, or he/she may deliberately provide inaccurate location information. We will discuss two different user models in Section 6.2.2.3 which try to broadly emulate these two types of user behaviors while providing feedback.

6.2.1.2 Signature of a User Feedback, $\mathcal{S}$

We first discuss our choice of a user feedback's signature, and then prove that it is an efficient one. During the offline training phase of a fingerprint-based positioning system, we know that the system administrator positions himself/herself at a particular location of interest for the RSSs to be measured at the APs. The RSSs perceived at the APs actually denotes the signature of that particular location. We also utilized the RSSs measured at the APs during a user feedback to denote its signature in a similar way taking into account some additional details. For example, in order to denote the signature of a user feedback, we sample the signal strengths perceived at APs over a 5-second window, and instead of using a single sample from each AP, the *mean* of all the samples over the 5-second window has been used. Furthermore, the time when a user clicks his actual position in the map is treated as the median of that window. Our approach is taken in view with the following facts:

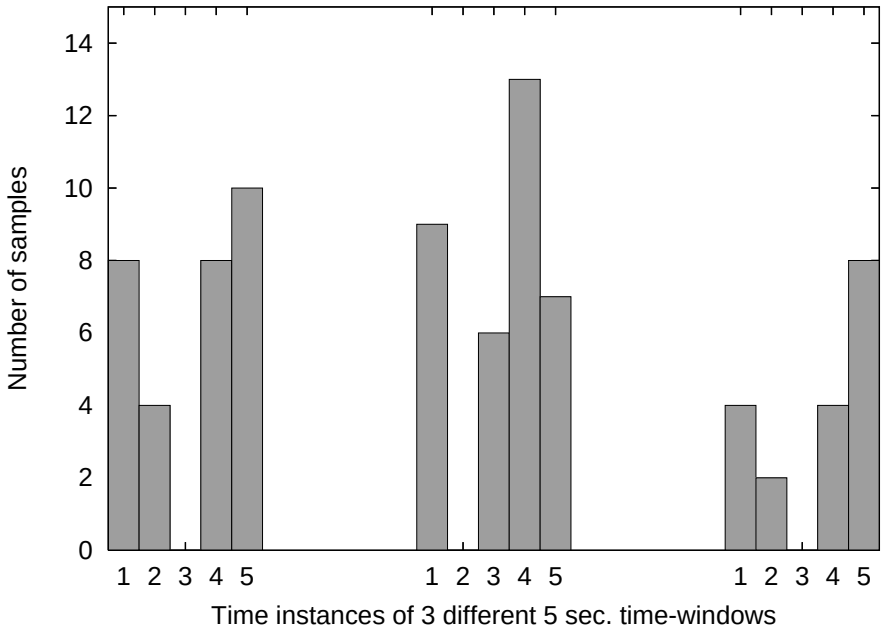


Figure 6.2: Number of samples collected per second over three 5-sec windows. It can be seen that, some slots are empty.

- Whenever a user clicks to input feedback, it is reasonable to assume that he/she has been at that particular location for a while. Hence, we have chosen the clicking instant of the user as the median of the 5-second window, rather than the beginning of the window.

- The probability that an AP fails to collect any sample from the mobile node (MN) during a user feedback is greatly reduced as well. Fig. 6.2 shows some cases when our AP failed to receive any sample from the MN within certain slots of a user feedback's time-window. If the probability that an AP receives a sample from an MN is q, then the probability that an AP receives at least one sample within the 5-second window can be expressed as, $1-(1-q)^{5m}$, where m is the number of packets sent by the MN within a 1-second slot and each 1-second slot is assumed to be independent. For example, if $q = 0.5$ and $m = 2$, the probability of getting a sample at the AP during a user feedback increases

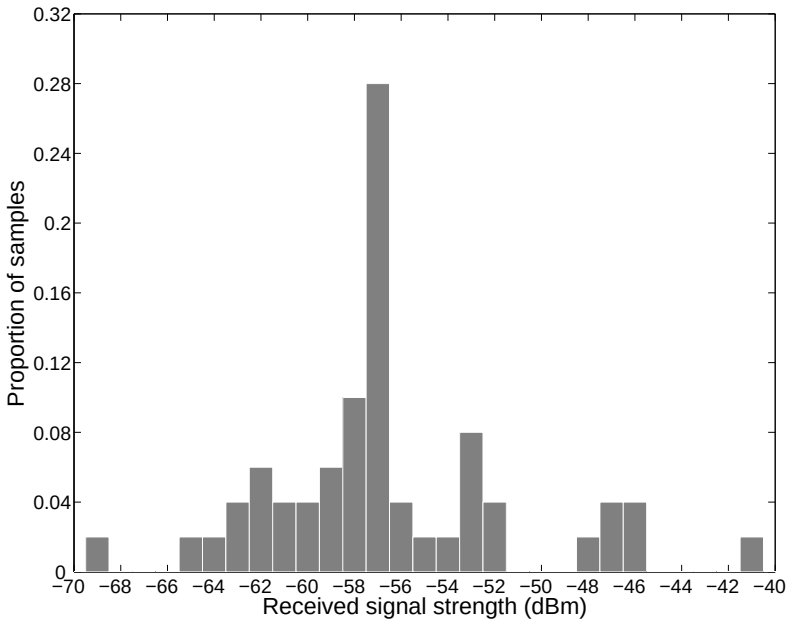


Figure 6.3: Histogram of signal strength samples received at an AP when the mobile device used in training is stationed at a particular location.

from 0.75 to 0.999 when a 5-second window is considered compared to 1 second.

- Capturing more samples should provide more information about the signal strength distribution at a particular location, which generally has a tail (see Fig. 6.3). The use of just a single sample would be unlikely to work well.

- The *mean* of all the collected samples' signal strengths inside the time-window is an efficient unbiased estimate of a user feedback's signature compared to any other linear combination of the samples' RSSs. This can be realized with the help of Theorem 6.1.

Theorem 6.1. *Suppose S_k denotes the signal strength distribution of the samples collected at the k^{th} AP during a user feedback. If S_{ki} specifies the i^{th} sample's RSS of the n samples observed inside the time-window at that AP, then the linear combination of observations $\sum_{i=1}^{n} a_{ki} S_{ki}$ is an unbiased estimate of $E(S_k)$ given $\sum_{i=1}^{n} a_{ki} = 1$. It is also the most efficient one when $a_{ki} = \frac{1}{n}, \forall i \in \{1, 2, \ldots, n\}$.*

Proof. The linear combination $\sum_{i=1}^{n} a_{ki} S_{ki}$ is an unbiased estimate of $E(S_k)$ because, $E(\sum_{i=1}^{n} a_{ki} S_{ki}) = \sum_{i=1}^{n} a_{ki} E(S_{ki}) = E(S_k) \sum_{i=1}^{n} a_{ki} = E(S_k)$. Since the estimate is unbiased, then the particular combination that is most efficient is the one which minimizes the variance, i.e., var($\sum_{i=1}^{n} a_{ki} S_{ki}$) = $\sum_{i=1}^{n} {a_{ki}}^2$ var(S_{ki}) = var(S_k) $\sum_{i=1}^{n} {a_{ki}}^2$. Consequently, the problem can be reformulated as, minimize $\sum_{i=1}^{n} {a_{ki}}^2$ subject to $\sum_{i=1}^{n} a_{ki} = 1$. Now, using basic optimization theory, it directly follows that the particular linear combination $\frac{1}{n} \sum_{i=1}^{n} S_{ki}$, or the sample *mean*, $\bar{S}_k$, is the most efficient unbiased estimator of $E(S_k)$. □

Corollary 6.2. *If $\bar{S}_k$ is an efficient unbiased estimate of the signal strength samples' signature collected at the k^{th} AP inside a time-window, then for a positioning system with K APs, $\mathcal{S} = [\bar{S}_1\ \bar{S}_2\ \ldots\ \bar{S}_K]^T$ is indeed an efficient unbiased estimate of a user feedback's signature.*

Proof. Corollary 6.2 can be realized by extending Theorem 6.1 for all the K APs, together with the assumption that the APs are independent of each other [30]. □

6.2.1.3 Credibility or Weight of a User Feedback, w

Without the credibility factor, w, a user feedback is typically a traditional training sample of location and RSS signature pair $(\mathcal{L}, \mathcal{S})$ from a positioning system's perspective. The traditional training samples are generally collected by a positioning system's administrative people. Therefore, all the samples are treated with equal importance. On the other hand, the sources of user feedbacks can be different entities (e.g., system administrators, normal users, intruders etc.). Consequently, there should be certain credibility factor associated with each feedback given, i.e., a measure for the system to believe that the user is actually at his/her claimed position. In many ways, this approach is similar to a *location verification* technique which ensures that the claimed source location is associated with a high level of trust. Existing location verification tech-

niques [83–86] either accept/reject a source's location claim. They generally require specialized hardware (incorporated with non-RF technologies) to verify a source's location claim more precisely [83, 84] or the accuracy level within which the location claim is verified, is set to be quite large [85, 86]. However, our positioning system has certain implications which makes the use of these location verification techniques infeasible:

- Our positioning system is built upon RF technology (Bluetooth) preferably using off-the-shelf hardware in order to provide location service in a cost-effective way. Consequently, the more precise solution to verify a location claim with the help of specialized hardware is not applicable.

- The accept/reject policy of the existing location verification techniques would restrict the user feedback to have only one of the two extreme values, i.e., $w \in \{0, 1\}$. If a strict margin is set for incorporating the user feedback, then many useful feedback might be filtered out. On the contrary, if a large tolerance level is set, many malicious user feedback might be incorporated which may ultimately cause the actual accuracy offered by the system to deteriorate.

Therefore, instead of an accept/reject policy of the existing location verification techniques, we come up with a strategy which assigns *relative* weights to the user feedbacks utilizing their credibility. Later on, it will be shown that, this approach actually helps in fine-tuning an existing positioning system to achieve better accuracy. Next, we elaborately describe how the user feedbacks are assigned relative weights based on their credibility while being incorporated into the system. In order to realize this, we first describe the "Region of Confidence (RoC)" concept, which subsequently helps to derive our weight assignment policy for each individual feedback.

Region of Confidence: We define a system parameter, RoC, which gives a measure of the system's overall accuracy and precision[1]. We express RoC as a two-parameter entity, i.e., (e, p), where the parameters e, and p, denote the accuracy grain size and the expected precision of the system, respectively. In localization literature, the term "accuracy" generally indicates the grain size of the position information provided (in some works, the accuracy grain size is referred as "localization error distance" as well), while the term "precision" specifies how often we could attain that accuracy [6]. For example, if a positioning system can determine positions within 3 meters for about 90 percent of the measurements, that particular system qualifies to be 90% precise in providing 3-meter accuracy. Intuitively, a higher precision would compel the system to provide a coarser accuracy, and similarly, in order to achieve finer accuracy, the system may turn out to be not so precise. We define RoC in a way that considers both requirements, in order to facilitate our feedback-based positioning system. In general, RoC provides a measure of likelihood of a user's estimated position and also influences the weights that would be associated with the feedbacks which we describe later.

In order to create the "Precision vs. Accuracy" graph of Fig. 6.4(a), which we term as "RoC profile graph", first we assume that our positioning system is already in an initial state with some training samples. Now, we inspect its performance when *well-behaved* users' (whose claimed locations do not deviate from their actual locations by a large margin) feedbacks are incorporated into the system in order to obtain the "RoC profile graph". It can be seen that, the shape of our "RoC profile graph" has a similar trend as those "Precision vs. Accuracy" curves found in existing localization literature [32, 87]; it shows that the precision, p, increases with larger accuracy grain size or localization error distance, e. Intuitively, the "RoC profile" may not be fully reflective of the system's actual state with only a limited number of user feedbacks. As

[1]Note that, our definition and purpose of RoC is quite different from an earlier work. In [21], RoC was formed utilizing simple geometry in order to fight *aliasing*, i.e., to eliminate physically different locations which have similar signatures in signal space.

we gather more and more user feedbacks, we can approximate the "RoC profile" more accurately (using the feedbacks as both training and testing samples). In the following section, we depict how the "RoC profile" has been utilized to derive the trend of the credibility to be assigned to a user feedback.

Feedback Weight Assignment Policy: Since user feedbacks may contain dubious information, we should not treat all feedbacks with equal importance. Whenever a user claims to be at a particular location via feedback, that information is associated with a certain degree of credibility. In order to calculate this credibility factor, consider a positioning system where n user feedbacks have been utilized as test samples to obtain the "RoC profile". Subsequently, for any point (e, p) of Fig. 6.4(a), it is obvious from the definition of RoC that, $p \times n$ user feedbacks' estimated positions do not deviate from its actual one by more than e. In other words, if we think of a circle with the accuracy grain size or localization error distance, e, as radius, then $p \times n$ user feedbacks can be thought to be inside it. Now, suppose if we increase the radius e by a small amount Δe (i.e., p also increases in Fig. 6.4(a)), then Δn new user feedbacks fall inside the new area. So, the proportion of user feedbacks falling inside the area $[\pi(e + \Delta e)^2 - \pi e^2]$ is $\frac{\Delta n}{n}$. Consequently, we denote the probability of occurrence of a user feedback inside this unit area as,

$$\kappa = \frac{\frac{\Delta n}{n}}{[\pi(e + \Delta e)^2 - \pi e^2]} \approx \frac{\Delta n}{2\pi n e \Delta e}. \tag{6.2}$$

Subsequently, we define the *weight* or *credibility* of the i^{th} user feedback utilizing (6.2) as follows:

$$\omega_i = \frac{\kappa_i}{\max\{\kappa_1, \kappa_2, \ldots, \kappa_n\}}. \tag{6.3}$$

Note that, ω_i is just the normalized form of κ_i so that $\omega_i \in [0, 1)$. Now, let us investigate the rationale behind choosing such a weight assignment criteria. Consider two user feedbacks, i and j with RoC (e_i, p_i) and (e_j, p_j), respectively. Their positions in the

"RoC Profile Graph" are shown in Fig. 6.4(a) where $e_i < e_j$. Following similar steps which were involved in obtaining (6.2), we have,

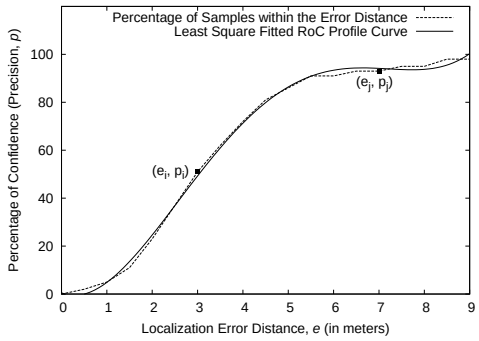
$$\kappa_i \approx \frac{\Delta n_i}{2\pi n e_i \Delta e} \quad \text{and} \quad \kappa_j \approx \frac{\Delta n_j}{2\pi n e_j \Delta e}.$$

The parameters e's and Δn's have certain effects in the above expressions:

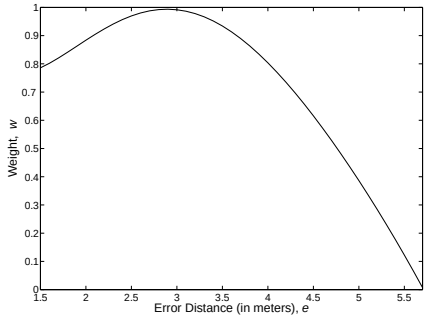
- $e_i < e_j$ implies the accuracy of the i^{th} feedback's estimated position by the system is higher than that of the j^{th} user feedback. Therefore, from the system's perspective, it is natural to believe the i^{th} user feedback more than the j^{th} one.

- Consider the number of user feedbacks, Δn_i and Δn_j which fall into the two new areas that have been formed by extending the radius e_i and e_j by the same amount, Δe, respectively. If $\Delta n_i > \Delta n_j$, then a greater number of user feedbacks which are used to create the "RoC Profile Graph", falls into the i^{th} feedback's new area than that of the j^{th} feedback's area. Consequently, it is natural for the system to believe the i^{th} user to be more *well-behaved* since the system's "RoC Profile Graph" had been created utilizing the *well-behaved* users' feedbacks as mentioned in the previous section. Therefore, it is only fitting to assign more weight to the i^{th} user feedback than the j^{th} one.

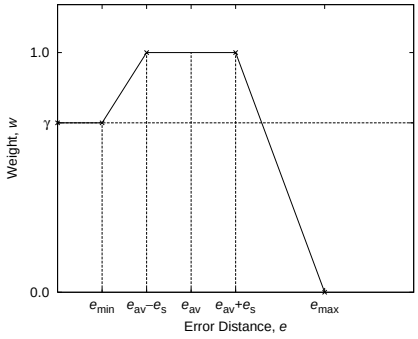
From Fig. 6.4(a), using the numerical values of the parameters, $n = 44$, $\Delta e = 0.5\text{m}$, $e_i = 3\text{m} < e_j = 7\text{m}$, and $\Delta n_i = 5 > \Delta n_j = 1$, we find, $\kappa_i > \kappa_j$. In other words, the i^{th} user feedback is more believable than the j^{th} user feedback from our positioning system's perspective. Next, we describe our ultimate simplified weight assignment policy for each individual feedback taking into account the aforementioned facts.



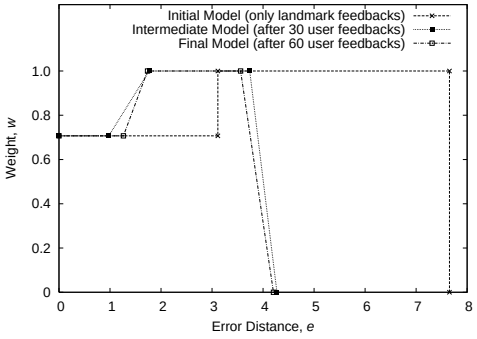
(a) RoC profile graph showing that precision increases with larger accuracy grain size or localization error distance.



(b) Feedback weights' profile generated from the RoC profile shown in Fig. 6.4(a) using Eqn. (6.3)



(c) Our simplified feedback-weight assigning model pertaining to Fig. 6.4(b).



(d) Various stages of our feedback-weight assigning model as the number of user feedbacks increases.

Figure 6.4: Illustration of how we approximate the feedback-weight assigning model from the RoC profile graph, as well as its variation when different number of feedbacks are incorporated.

By utilizing the RoC profile together with (6.3), we obtain the trend for weights to be associated with user feedbacks as shown in Fig. 6.4(b). We observe that, the weight's maximum occurs when the accuracy grain size or the localization error (e) of the user feedback's estimated position is close to our system's average localization error (≈ 3m), and decreases as the estimation error becomes larger. Since it is desirable to have a weighting scheme that is simple and yet capable of evolving with time as more user feedbacks become available, we define a feedback-weight assigning model as follows. A maximum weight of 1 shall be assigned when the localization error (e) of a user feedback's position is within one standard deviation (e_{s}) from the average error (e_{av}), as shown in Fig. 6.4(c). This is in accordance with the view that our system is fairly accurate and therefore, we expect the system's estimated positions' errors to be around this average quantity. Assigning maximum weight around one standard deviation of this average helps to build, and subsequently, fine-tune the system gracefully. From $e_{\text{av}} + e_{\text{s}}$ to e_{max} (maximum error), the weight follows a similar trend as in Fig. 6.4(b). The horizontal dotted line (i.e., $w = \gamma$) of Fig. 6.4(c) indicates the filter of our weighting scheme. We associate a constant weight, γ (which is 3 dB lower than w_{max}), to the user feedbacks when the estimation error is less than e_{min}, in the view that our system's predictions of these positions are already quite good. The weight assignment policy for the i^{th} user feedback of our model as shown in Fig. 6.4(c) can be summarized as,

$$w_i = \begin{cases} \gamma & e_i \leq e_{\text{min}} \\ 1 + (1-\gamma)\left(\frac{e_i - e_{\text{av}} + e_{\text{s}}}{e_{\text{av}} - e_{\text{s}} - e_{\text{min}}}\right) & e_{\text{min}} < e_i < e_{\text{av}} - e_{\text{s}} \\ 1 & e_{\text{av}} - e_{\text{s}} \leq e_i \leq e_{\text{av}} + e_{\text{s}} \\ 1 + \left(\frac{e_i - e_{\text{av}} - e_{\text{s}}}{e_{\text{av}} + e_{\text{s}} - e_{\text{max}}}\right) & e_{\text{av}} + e_{\text{s}} < e_i < e_{\text{max}} \\ 0 & e_i \geq e_{\text{max}} \end{cases}$$

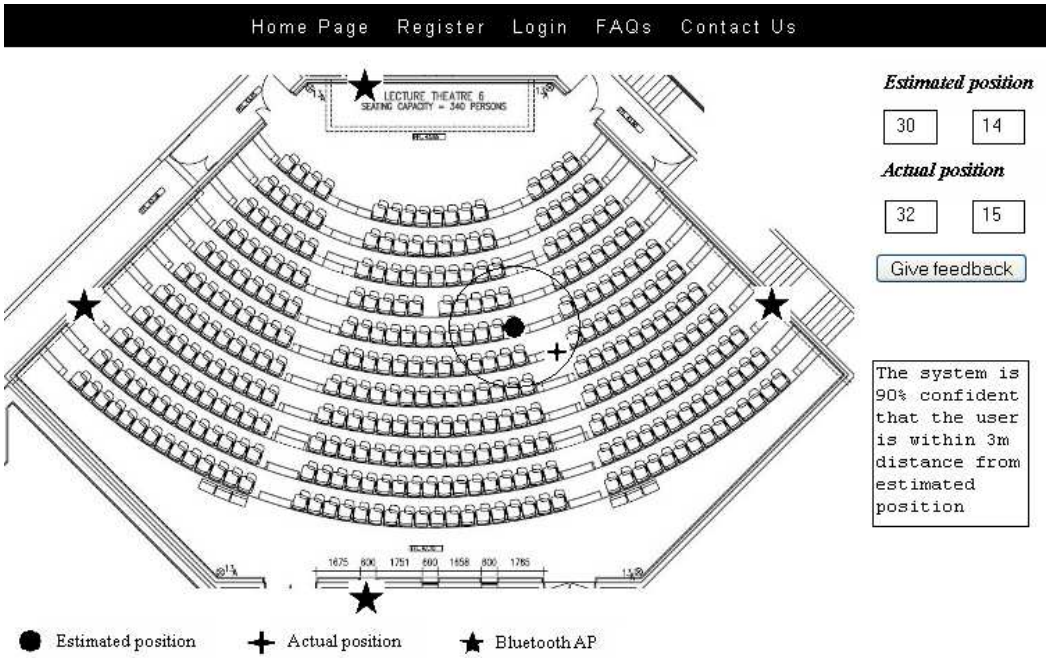


Figure 6.5: Interface for user feedback input – the experimental testbed is a lecture theater in campus (Testbed 1).

where $e_i = \sqrt{(x_i - x_{\text{est}})^2 + (y_i - y_{\text{est}})^2}$, $[x_i \; y_i]^T$ is the i^{th} user's claimed location, and $[x_{\text{est}} \; y_{\text{est}}]^T$ is the system's estimate of that user's position.

Fig. 6.4(d) shows the evolvement of our feedback-weight assigning model as user feedbacks are increasingly incorporated. Our initial system only consists of *landmark* feedbacks (e.g., the feedbacks from the 4 APs). Two other stages of our system are shown in Fig. 6.4(d) where 30 and 60 *well-behaved* user feedbacks are subsequently incorporated. The definition of various user feedbacks (e.g., landmark, well-behaved etc.) can be found in Section 6.2.2.4. For each stage of the system, 44 testing samples which are completely different from the incorporated user feedbacks are utilized to obtain the error model. As can be seen, this model helps to improve the accuracy of our system, since both the average error and its standard deviation decreases with increasing number of user feedbacks.

6.2.2 System Description

6.2.2.1 User Interface and Experimental Testbed

We start by providing a brief description of our user interface used to input feedbacks into our system. Fig. 6.5 shows the interface for a user to input feedbacks that are to be incorporated into our positioning system. We can observe from the interface that, a user is always provided with the system's estimation of his/her position (i.e., the shaded circle on the map) together with the RoC. Subsequently, the user can choose to inform the system about his/her actual location by clicking on the corresponding position within the map, and pressing the "Give feedback" button. The experimental results in this chapter are based on Bluetooth data samples collected from both Testbed 1 and 2.

6.2.2.2 Usage of User Feedback in Positioning Algorithms

Depending on the positioning algorithm used, there are various ways how a user feedback can be utilized. In the following, we briefly describe the two approaches we have undertaken in order to make use of the user feedback in our positioning algorithm.

- As discussed in Section 6.1.1, we utilize interpolation technique to create the RSS signature of a *fictitious* training point where no training sample has been taken. Unlike a typical fingerprint-based positioning system that requires an exhaustive sample collection phase, interpolation helps to achieve the same goal with much fewer training samples. In addition, it is advantageous in our case since the user feedback locations may not be uniform over the entire localization area. An interpolation technique can help to fill up the voids in the training database where no user feedback has been obtained. In order to deduce a fictitious training point j, each AP's RSS is formulated according to weighted linear regression formula exploiting the signal strength values collected at the APs for

user feedbacks. If there are K APs, K different regression equations will be formed in order to deduce a single fictitious training point's fingerprint. However, the difference from our interpolation technique explained in Appendix B is – whereas the weight in (B.2) corresponds only to the spatial similarity factor; here, the user feedback's credibility factor is also taken into consideration regarding the weight calculation in (C.1). In Appendix C, we provide the details about how the interpolation technique predicts the RSS of a *fictitious* training point where real training samples are not collected or obtained through user feedbacks.

- We have used two well-known localization algorithms (i.e., weighted $\mathcal{K}$-Nearest Neighbors ($\mathcal{K}$-NN) and Bayesian) [17,20] where the user feedbacks' weights are utilized to denote the weights of the algorithms.

6.2.2.3 User Models

In this section, we describe our two user models which try to emulate the two broad categories of the user behavior while giving feedbacks. These "user feedback behavior" models are utilized in the experiments to emulate the *real* user feedbacks from our collected data.

- *User Model 1:* The user may be unfamiliar with the surroundings, and subsequently fails to pinpoint his/her actual position on the map. We model this phenomenon as, $[x\ y]^T = [x_a + N(0,\sigma^2) \quad y_a + N(0,\sigma^2)]^T$, where x_a and y_a denote the actual location coordinates when no uncertainty is involved and $N(0,\sigma^2)$ is a normal distribution with zero mean and variance σ^2. We assume that this is the most common model of a user's feedback and it is also capable of modeling many different user feedbacks (by varying σ). For example, we know that a *well-behaved* user is the one whose claimed location does not deviate from

his/her actual location by a large margin. For experimental purposes, we model a *well-behaved* user as one where the uncertainty parameter of the feedback position (i.e., σ^2) does not exceed the system's ultimate achievable average accuracy. Since our positioning system can offer 3m average accuracy, we assume that the feedback position of a *well-behaved* user regarding our system conforms to the equality, $\sigma = \sqrt{3}$m.

- *User Model 2:* There may be some feedbacks where the user feels totally unsure about his actual position corresponding to the map. We model this phenomenon as, $[x\ y]^T = [U(0, x_{\text{max}}) \quad U(0, y_{\text{max}})]^T$, where x_{max} and y_{max} depict the maximum possible location coordinates of the testbed and $U(\cdot)$ denotes a uniform distribution over the range. The feedbacks given by those who try to sabotage the positioning system intentionally, also fall into this category.

6.2.2.4 Classification of User Feedback

Based on our user models of the previous section and the weight assignment policy for each individual feedback discussed in Section 6.2.1.3, we classify user feedbacks into four categories:

- *Super-user feedback:* These are the feedbacks provided by system administrators and alike, and they are expected to be included into the system with 100% belief (i.e., $w = 1$).

- *Regular-user feedback:* We consider the feedbacks from ordinary users who use the positioning system's services to be the mainstay in the fine-tuning of our system. These are the most common type of feedbacks which are amalgamated with some uncertainties. Our *User Model 1* discussed in the previous section tries to emulate this particular type of feedback.

- *Landmark feedback:* The APs can be regarded as sources of feedbacks as well, since they also transmit radio signals, and their locations are known and fixed. We have four such APs in each of our experimental testbeds as shown in Fig. 6.5 and Fig. 6.6, respectively. Note that, the RSS signature vector of this type of feedback comprises of $K-1$ components instead of K, because one of the K APs is actually considered as an MN here. We fill this void with the maximum RSS rating corresponding to our Bluetooth adapter. Apart from that, various other devices (e.g., beacons, card-readers, tags etc.) installed at several fixed positions in the building could act as continuous sources of landmark feedback too. Landmark feedback is a form of super-user feedback (just that the sources are static fixed points) since it is always believed with $w = 1$. Therefore, the inclusion of such static fixed points as a source of feedback will increase the number of super-user feedbacks, and subsequently, will have positive impact on localization accuracy.

- *Spurious-user feedback:* The feedbacks given by those users who are oblivious about their surroundings, and also those who aim to sabotage the positioning system, are harmful. Instead of fine-tuning the system to achieve better accuracy, these spurious-user feedbacks could make the positioning error larger if incorporated. Our weight-assignment policy of Section 6.2.1.3 ensures that these types of feedbacks are filtered out.

6.2.3 Results and Findings

The results of Section 6.2.3.1 are based on the experimental data of our lecture theater testbed (Fig. 6.5) while the results presented in Sections 6.2.3.3 and 6.2.3.4 are obtained from our research laboratory testbed's data (Fig. 6.6).

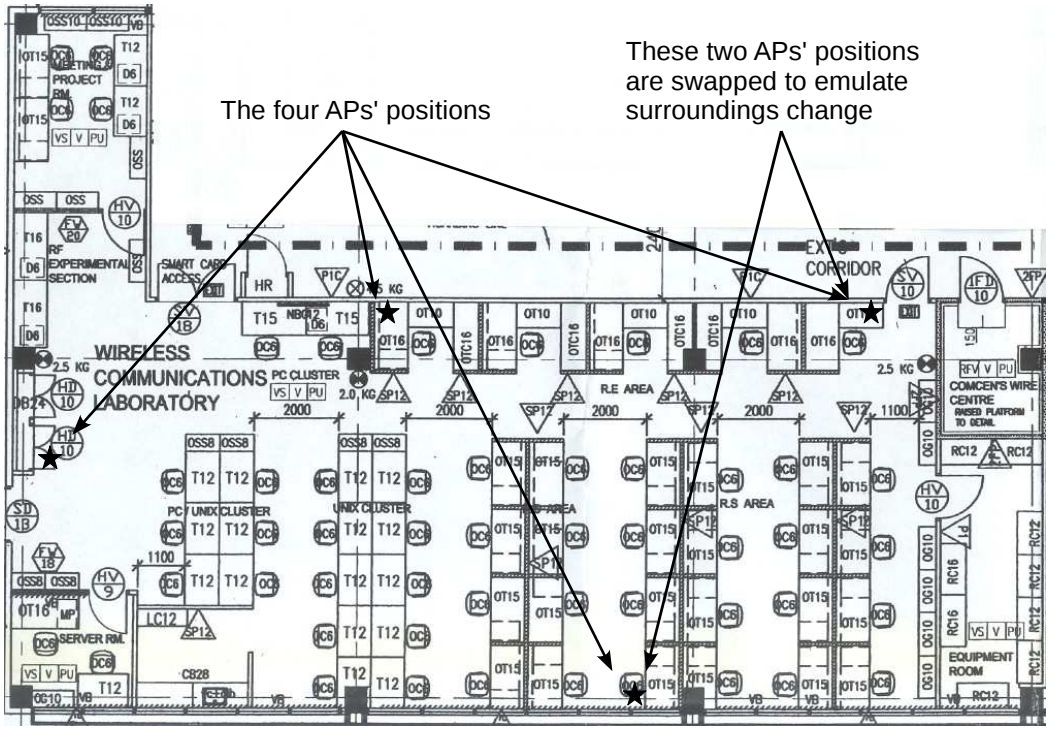
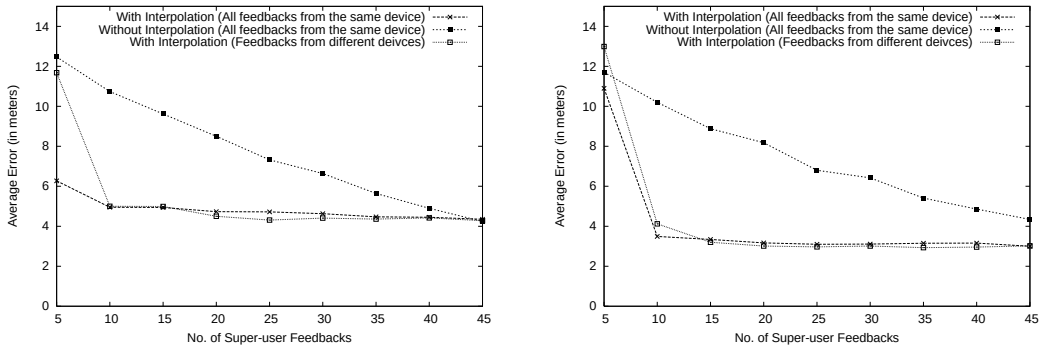


Figure 6.6: Emulating surroundings change in Testbed 2.

6.2.3.1 Interpolation aids our user feedback based positioning system

In order to demonstrate the usefulness of our interpolation based approach described in Section 6.2.2.2, we have carried out an experiment that only considers super-user feedbacks, where all feedbacks are assigned the maximum weight (i.e., $w = 1$). As can be seen from Fig. 6.7(a) and 6.7(b), the system that uses interpolation easily outsmarts the one that does not.

Since different users are expected to carry devices with heterogeneous hardware, selecting RSS as a location fingerprint could easily hamper a user feedback based positioning system. RSS is known to vary quite significantly at a particular location for different device hardware even under the same wireless conditions [25–27, 29]. As a result, we have chosen a robust location fingerprint, namely, *Signal Strength Difference (SSD)*, since it is argued to be able to accommodate devices with heterogeneous hardware solutions unlike the RSS [27]. We also verified our system's robustness when



(a) When KNN localization algorithm is used.

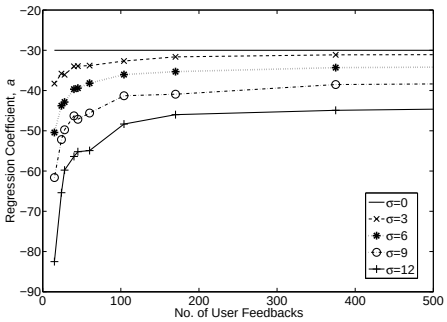
(b) When Bayesian localization algorithm is used.

Figure 6.7: Demonstration of how interpolation helps to improve our positioning system's accuracy – only super-user feedbacks are considered here.

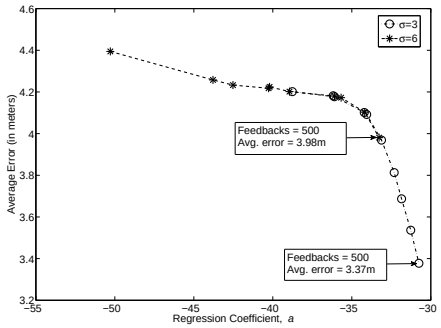
the users input their feedbacks using different types of devices (e.g., Bluetooth Class 1 or 2 devices), which could easily occur in a real deployment scenario. Fig. 6.7(a) and 6.7(b) show similar performance for both cases, regardless of whether the user feedbacks are given using only one type of device or not. SSD has been used as a location fingerprint in the subsequent experiments as well. Even though a user feedback's signature is a K-dimension RSS vector, while applying it into a localization algorithm, a $(K-1)$-dimension SSD vector is always calculated from it.

6.2.3.2 Evolvement of user feedback based positioning system

In this experiment, we investigate the prospect of creating a positioning system utilizing only regular-users' feedbacks from scratch. We try to estimate the linear regression coefficients for the equation in (B.1) (given in Appendix C) which are necessary for generating the interpolated training points from user feedbacks. Here, we emulate different types of users by changing the value of σ of "*User Model 1*" which we have defined in Section 6.2.2.3. We contend that if the two linear regression coefficients (i.e., a and b) computed from regular-user feedbacks can somehow match the coefficients computed from super-user feedbacks, then our interpolation-based approach



(a) The values of regression coefficient, a, required for predicting the RSS at an AP when various user types of "*User Model 1*" (i.e., different σ's) are considered.



(b) Two cases of Fig. 6.8(a), $\sigma = 3$m and $\sigma = 6$m, are picked to show the corresponding average errors of the system with the calculated a's.

Figure 6.8: Simulation results of how different user behaviors affect the regression coefficient a values and correspondingly, influence the system's achievable average accuracy.

should perform equally well even though these feedbacks have uncertainties. We see from Fig. 6.8(a) that using feedbacks from users exhibiting lower uncertainty (e.g., $\sigma = 3$) can almost achieve the same a as the case when no uncertainty is involved ($\sigma = 0$). Furthermore, it can be noted from Fig. 6.8(a) that increasing the uncertainty in user feedbacks have the effect of swaying the estimated a values away from the $\sigma = 0$ case. Similar observations have been made with the other coefficient, b.

In our interpolation-based approach, we first calculate the regression coefficients (i.e., a and b) for all the APs at an interpolated point making use of the user feedbacks as training samples. Subsequently, the RSS signatures of the APs at every interpolated point are calculated, and all of them are then treated as normal training samples together with the user feedbacks in our localization algorithm. Table 6.1 lists the average localization errors when a significant number ($= 500$) of user feedbacks with different values of uncertainty parameter, σ, are being considered. We see that the average accuracy (3.37m) achieved for $\sigma = 3$m case is very close to the accuracy when there is no uncertainty (3.1m). This is expected since the calculated a value for $\sigma = 3$m case after 500 feedbacks is very close to the a value obtained for $\sigma = 0$ (see Fig. 6.8(a)).

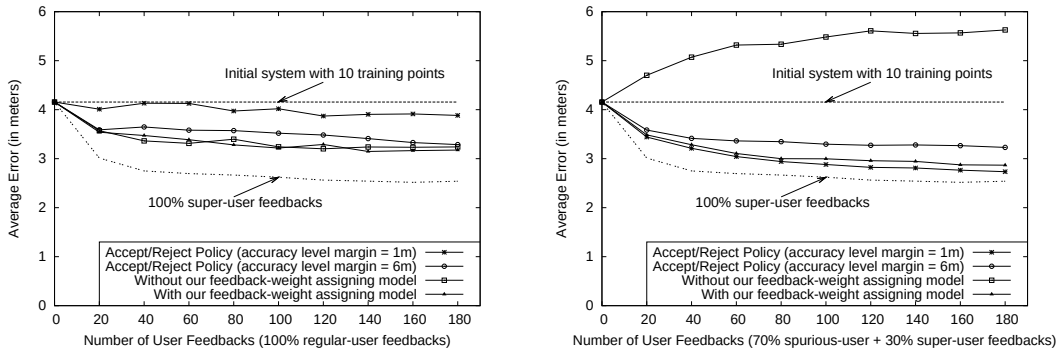
Table 6.1: Relationship between the uncertainty parameter, σ, and average localization error for our experiment conducted.

Number of User Feedbacks	Uncertainty Parameter, σ of *"User Model 1"*	Average Localization Error (in meter)
500	$\sigma = 0$	3.1
	$\sigma = 3$m	3.37
	$\sigma = 6$m	3.98
	$\sigma = 9$m	4.71
	$\sigma = 12$m	6.18

The higher uncertainty cases (e.g., $\sigma = 6\text{m}, \sigma = 9\text{m}$, etc.) report coarser accuracy as can be seen from Table 6.1, which is also justified according to their curves shown in Fig. 6.8(a). Therefore, we can approximate the regression coefficients of our interpolated points more accurately for user feedbacks with lower uncertainty which in turn yields better localization accuracy. In a nutshell, we argue that if we decide to build our system with user feedbacks from scratch, our interpolation-based approach may still enable us to achieve reasonable accuracy, provided that the user behavior does not stray too drastically. Note that the results for this particular experiment are obtained through simulation, unlike the others in this paper where real experimental data are used.

6.2.3.3 Fine-tuning of an existing positioning system utilizing user feedbacks

In this section, we wish to show that we could fine-tune a positioning system in order to achieve finer accuracy by exploiting our feedback-weight assigning model, irrespective of any assumption on user behavior. For this experiment, we choose two different combinations of user feedbacks where one consists of only *well-behaved* regular-user feedbacks while the other comprises of 70% spurious-user and 30% super-user feedbacks. In both cases, we assume that the positioning system is already running with some feedbacks (4 landmark feedbacks + 6 super-user feedbacks) so that we can ap-



(a) 100% *well-behaved* regular-user feedbacks.

(b) 70% spurious-user feedbacks + 30% super-user feedbacks.

Figure 6.9: Performance comparison of our feedback-weight assigning model with other options in the fine-tuning of an under-trained positioning system.

proximate the initial "RoC Profile". Consequently, we can come up with the feedback weight-assigning model of Section 6.2.1.3 from this initial state of our system. We consider 137 testing points to evaluate the localization errors which are completely different from the user feedback points. As more user feedbacks become available, the weight-assigning model continuously updates itself in a similar manner as previously shown in Fig. 6.4(d), which helps to fine-tune the system.

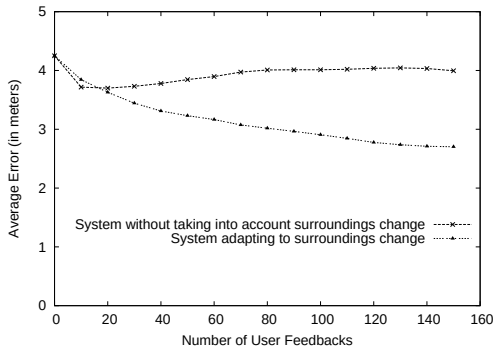
The two horizontal lines of Fig. 6.9(a) and 6.9(b) at 4.16m represent the initial system's performance with only 10 training points. Our feedback-weight assigning model shows that the system's performance improves when more feedbacks are incorporated. Without our feedback-weight assigning model, the system's performance deteriorates when spurious-user feedbacks dominate as can be seen from Fig. 6.9(b). For 100% spurious-user feedbacks scenario (the results are omitted for brevity), our system's performance remains relatively unchanged from the initial system's performance. This means that our feedback-weight assigning model could shield the system from the adverse effect of this type of feedbacks. For the well-behaved user case, the feedback positions may turn out to be very close to the actual positions which will eventually make them a bit similar to super-user feedbacks. The inclusion of super-

user feedbacks into the system always helps regardless of whether we are using our model or not. Therefore, the "without feedback-weight assigning model" might have been seen to perform almost similar to (or even slightly better than) our model in Fig. 6.9(a). Our model's effectiveness over the "without feedback-weight assigning model" can be realized when different types of feedbacks are mixed (e.g., one instance can be seen in Fig. 6.9(b)).

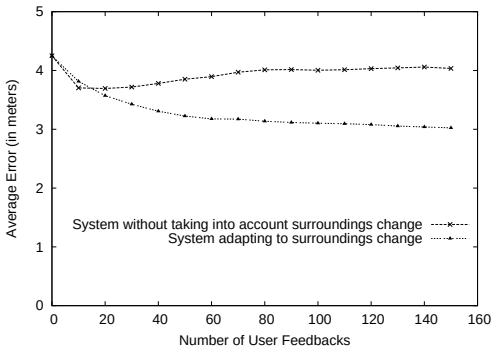
We also compare the accept/reject policy of location verification techniques discussed in Section 6.2.1.3 to incorporate a user feedback with varying accuracy level margins. If the accuracy level margin is set too large (≈ 6m), a number of spurious-user feedbacks may get through to the system, thereby causing it to perform worse. Setting a strict margin (e.g., 1m) may overcome this issue as can be seen from Fig. 6.9(b). However, if the accuracy level margin is set too strict, many of the well-behaved regular-user feedbacks are rejected. Consequently, the system's performance does not improve much over the initial system when this type of feedback dominates as revealed in Fig. 6.9(a). On the contrary, our feedback-weight assigning model is quite automated (no need for manual setting of accuracy level margin) and is shown to perform reasonably well in the presence of different types of feedbacks.

The 100% super-user feedback curves in both Fig. 6.9(a) and 6.9(b) show the performance when the feedbacks are given by super-users only (i.e., $w = 1$). This performance is comparable to the traditional fingerprint-based system where all the samples are collected exhaustively by administrators. This provides a performance benchmark for the user feedback based positioning system.

Note that we have only provided Bayesian algorithm's results; the KNN algorithm's results show similar trends, and are not included here.



(a) 30% regular-user + 70% super-user feedbacks.



(b) 70% regular-user + 30% super-user feedbacks.

Figure 6.10: Adaptation of our system when it perceives that the surroundings have changed.

6.2.3.4 Effect of change of surroundings on our user feedback based positioning system

One of the major drawbacks of existing fingerprint-based positioning systems is that it is not adaptable to environmental changes, i.e., the training phase has to be repeated all over again for the changed surroundings. Our system does not suffer from such shortcomings since user feedbacks are continuously employed to fine-tune it. Furthermore, our system's whole process of adapting to the changed environment is automated, and does not require any outside intervention. In order to help perceive that there is a change in the surrounding, we exploit landmark feedbacks. Since the landmark feedbacks from the APs are continuous, the system can approximate the APs' positions all the time. We infer that there is a change in surrounding when the estimated positions of all the APs deviate quite significantly from their actual positions. Algorithm 6.1 (in Page 39) describes the adaptation process of our positioning system. From algorithm 6.1, we see that, when the system perceives its surroundings to have changed, it enters into the adaptation mode. In this mode, all the previously incorporated user feedbacks are associated with an exponential outdate-factor together with their assigned weights. As a result, new user feedbacks are given more importance.

In order to emulate a change in the surrounding in our experiments, we swapped the positions of two of our APs as shown in Fig. 3.2. This serves our purpose of creating a changed environment since the two APs' signal strength signatures change quite significantly. Our initial system consists of 50 super-user feedbacks from the *old* setting and we utilize 137 testing points from the *new* setting to evaluate the localization errors. The two curves of Fig. 6.10(a) and 6.10(b) depict the performances of two systems where one system is incorporated with our surroundings change algorithm and the other one is not. As can be seen from the figures, the system which could realize the change in surroundings, performs significantly better in the *new* setting as more user feedbacks are incorporated into the system. For this experiment, we choose two different combinations of user feedbacks where in one scenario, the super-user feedbacks dominate (Fig. 6.10(a)), while in the other, the regular-user feedbacks dominate (Fig. 6.10(b)). In both scenarios, our system could adapt seamlessly with the surroundings change. Note that, the super-user dominating scenario demonstrates lower localization error for the same number of user feedbacks compared to the regular-user dominating scenario which is justifiable. The presence of spurious-user feedbacks also does not affect the adaptation process (the results are omitted for brevity). This is due to our feedback-weight assigning model which is found to be successful in dealing with them in the previous section. We have also observed in our experiments that around $20 \sim 30$ user feedbacks are required for the system to return to its normal mode (i.e., to leave its adaptation mode). This state transformation occurs when the landmark feedbacks start to give better estimations of the APs' positions again.

6.3 Summary and Conclusions

In this chapter, we proposed two approaches to relieve/shorten the exhaustive training phase typically seen in the fingerprint based positioning systems. Firstly, a simple

Algorithm 6.1 Adaptation of our user feedback based positioning system

System State: A positioning engine with n samples or user feedbacks. Let $(e_{\text{ex}}, p_{\text{ex}})$ denote the system's expected RoC. It is a tunable parameter for the administrator within which he/she expects the K landmarks' (e.g., APs') estimated positions to be verified. If all the K landmarks' estimated positions deviate from $(e_{\text{ex}}, p_{\text{ex}})$, the system infers surroundings change, and enters the adaptation mode. The system returns to normal mode again when all the K landmarks' estimated positions are within the system's expected RoC $(e_{\text{ex}}, p_{\text{ex}})$. The landmarks' positions are estimated continuously by the system from the landmark feedbacks.

1: **for** every *new* batch of N feedbacks collected **do**
2: **if** all K APs' estimated positions deviate from $(e_{\text{ex}}, p_{\text{ex}})$ **then**
3: $h \leftarrow \alpha$ {outdate factor: α small constant – adaptation mode}
4: **else**
5: $h \leftarrow 0$ {no outdate factor – normal mode}
6: **end if**
7: **for** $i = 1$ to n **do**
8: $w_i \leftarrow \exp(-h) \times w_i$ {outdating older samples' weights if $h \neq 0$}
9: **end for**
10: $n \leftarrow n + N$
11: calculate the interpolated RSS signatures as discussed in Appendix C
12: run localization algorithm (e.g., Bayesian or KNN) with only the feedbacks having $w_i \geq \gamma$ as test samples among the *new* feedbacks {w_i and γ are defined in 6.2.1.3}
13: update feedback-weight assigning model's parameters (i.e., $e_{\text{min}}, e_{\text{av}}, e_{\text{s}}$ and e_{max}) of Section 6.2.1.3.
14: **end for**
15: **goto** 1

weighted linear regression technique is applied to generate fictitious training points when the system contains very few training samples. Adopting an appropriate interpolation technique can go a long way in solving the drawbacks suffered by an under-trained positioning system, as vindicated by our analysis and results. Secondly, we propose a novel idea where users can take part in fine-tuning an under-trained positioning system. Our feedback-weight assigning model which assigns *relative* weights to user feedbacks, fine-tunes an under-trained positioning system, thereby, helps it to achieve finer accuracy. We also show that, if users are *well-behaved*, we can actually construct a positioning system incrementally from scratch exploiting our interpolation-based

techniques with the user feedbacks. We contend that the exhaustive training phase seen in the traditional location fingerprinting techniques might be relieved through it. Through the use of landmark feedbacks, we could successfully infer changes in the environment, and switch our system's mode to be more adaptable. The whole procedure is quite dynamic, and requires no intervention from the positioning system administrator's part. In summary, we conclude that our user feedback based positioning system is fairly accurate, cost-effective, robust and requires no or very little training phase. We have implemented our system in two testbeds – one is placed inside an amphitheater (Testbed 1) while the other is within a research laboratory (Testbed 2). Our system performed quite well in both scenarios as can be seen from the results.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this dissertation, we addressed some interesting issues arising from fingerprint-based positioning systems, and obtained favorable results. We provided an elaborate discussion on the indoor location fingerprints that are commonly used with two well-known RF technologies, namely, Wi-Fi and Bluetooth, and pointed out their pitfalls when they are used for localization purpose. Our analysis in Chapter 3 reveals that, although the RSS turns out to be the most viable option as a location fingerprint among all the signal parameters that are available from both Wi-Fi and Bluetooth, it still suffers from many pitfalls arising from device hardware variations. Hence, we define a robust location fingerprint, the SSD, which provides a more *robust* signature than the traditional RSS. We analyzed in detail why it can serve as a robust location fingerprint that is irrespective of the hardware used at the mobile device. Our analysis, as well as experimental results in Chapter 4, have both verified this claim. We compared SSD with two other robust location fingerprints, and found the SSD based algorithms' performance to be superior. In Chapter 5, we analyze the Cramér-Rao Lower Bound of localization using

SSD as location fingerprint. A novel characterization of the properties of this bound is presented that allows us to individually assess the impact of different parameters (e.g., number of APs, geometry of the APs, distance of the APs from the MN, etc.) on the accuracy of location estimates. This analysis should provide important guidelines for the system designer if SSD is the chosen location fingerprint.

In Chapter 6, firstly, we have elaborated on our idea of using weighted linear regression techniques to improve localization models when the system contains very few training samples. Adopting appropriate interpolation technique can go a long way in solving the drawbacks suffered by an under-trained positioning system, as vindicated by our analysis and results. As we have seen, without the aid of interpolation, our testbed of moderate size already requires around 60 real training points in order to attain reasonable accuracy. A larger deployment area would benefit greatly from our technique, in terms of reducing labor and cost requirements. Nevertheless, prior works [80] have indicated that, when sufficient samples have already been collected over the whole localization area, the gain arising from the use of interpolation techniques is not significant. Our results have shown a similar trend, as revealed in Fig. 6.1(a) and 6.1(b). Secondly, we propose an interpolation-based fingerprinting technique utilizing user feedback which does not require an exhaustive training phase typically seen in the indoor localization solutions. We argue that the contribution of users' feedback to any positioning system is two-fold. On one hand, users' feedback greatly help in fine-tuning an under-trained positioning system with proper filtering. On the other hand, if users are *well-behaved*, our experimental results show that the participation of end-users can actually assist in the construction of a positioning system incrementally from scratch. We also show that user feedback-based positioning system adapts quite well when surroundings change.

Although no work in the literature has been successful thus far in designing a reasonable Bluetooth-based positioning system, we have shown in our analysis that

previous works have used signal parameters that are inappropriate for localization purpose. However, if inquiry-based RSS is available and is used to generate SSD for use as a location fingerprint, a Bluetooth-based positioning system with reasonable accuracy can still be built as we have demonstrated in our experimental results.

In summary, we have derived analytically a robust location fingerprint definition, and verified it experimentally as well. We devised ways through weighted linear regression techniques and utilizing user feedback to facilitate under-trained positioning systems.

7.2 Future Work

In this dissertation, we have shown our new robust location fingerprint, SSD to perform well as a location fingerprint over MNs' hardware variations. From our research, we realized that there are generally two main characteristics of a *good* location fingerprint:

- It should provide a *stable* signature (e.g., signal strength) at a particular location even if the MN's hardware changes.
- From one location to another, this signature should also be distinct.

We only concentrated to explore a location fingerprint which produces *stable* signature (i.e., SSD) at a particular location even if the MN's hardware changes. However, an interesting research topic would be to incorporate both the requirements and come up with a location fingerprint that meets them well.

The design guidelines provided by this dissertation in Chapter 5 when SSD is chosen as the location fingerprint, may create a conflict to the design of WLAN or Bluetooth infrastructure for optimal signal coverage. A possible research topic based on the placement of access points to provide both communications and location service should be explored in the future. A comparison of the Cramér-Rao Lower Bounds

between SSD and RSS is also required in order to investigate both fingerprints' performances in more detail.

While our use of weighted linear regression model for generating fictitious points in Chapter 6 has achieved reasonable results, it may be worthwhile to pursue whether the use of more complex propagation models could further improve the performance of under-trained positioning systems.

The use of Bluetooth technology for localization certainly requires more investigation. Although previous works have largely provided discouraging results [62], or required the aid of additional wireless technologies (e.g., Wi-Fi) [21, 52], our experience with Bluetooth shows that it is a promising technology as well that should not be overlooked.

We had three experimental testbeds that can be categorized as testbeds of only moderate size. Additional experiments could be conducted on testbeds with different setup and larger size to explore its viability across different settings. Our user feedback based positioning system is currently based on Bluetooth wireless technology, but it can easily be extended to accept feedbacks from devices using other technologies as well (e.g., Wi-Fi). A hybrid system where the user feedback from devices with multiple wireless technologies is incorporated certainly poses an interesting research problem.

Bibliography

[1] T. Kindberg and A. Fox, "System software for ubiquitous computing," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 70–81, Jan. 2002.

[2] D. Saha and A. Mukherjee, "Pervasive computing: A paradigm for the 21st century," *IEEE Computer*, vol. 36, no. 3, pp. 25–31, Mar. 2003.

[3] J. J. Caffrey and G. L. st$\ddot{u}$ber, "Overview of radiolocation in CDMA cellular systems," *IEEE Communications Magazine*, vol. 36, no. 4, pp. 38–45, Apr. 1998.

[4] J. J. Caffrey and G. L. St$\ddot{u}$ber, "Subscriber location in CDMA cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 2, pp. 406–416, 1998.

[5] K. Pahlavan, X. Li, and J. Maleka, "Indoor geolocation science and technology," *IEEE Communications Mag.*, vol. 40, no. 2, pp. 112–118, 2002.

[6] J. Hightower and G. Borriella, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[7] J. Hightower and G. Borriello, "Location sensing techniques," *IEEE Computer*, 2001.

[8] T. S. Rappaport, J. H. Reed, and B. D. Woerner, "Position location using wireless communications on highways of the future," *IEEE Commun. Mag.*, vol. 34, no. 10, pp. 33–41, Oct. 1996.

[9] A. Ladd, K. Bekris, G. Marceau, A. Rudys, L. Kavraki, and D. Wallach, "Robotics-based location sensing using wireless Ethernet," Department of Computer Science, Rice University, Tech. Rep. TR02-393, 2002.

[10] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*. 4th Ed., Springer Verlag, 1997.

[11] P. Enge and P. Misra, "Special issue on GPS: The global positioning system," *Proc. of the IEEE*, vol. 87, no. 1, pp. 3–172, 1999.

[12] R. Want, A. Hopper, V. Falc$\tilde{a}$o, and J. Gibbons, "The active badge location system," *ACM Trans. on Information Systems*, vol. 10, no. 1, pp. 91–102, Jan. 1992.

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

[14] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," in *Proc. ACM MobiCom'00*, Boston, MA, Aug. 2000, pp. 32–43.

[15] Ubisense, "Precise real-time location systems and GIS consulting by Ubisense." [Online]. Available: http://www.ubisense.net

[16] T. S. Rappaport, *Wireless Communications – Principles and Practice*. Prentice Hall, 1996.

[17] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 775–784.

[18] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz, "A probabilistic room location service for wireless networked environments," in *In Proc. UbiComp'01*, Atlanta, GA, Sept. 2001, pp. 18–34.

[19] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," in *Proc. IEEE INFOCOM'04*, Mar. 2004, pp. 1012–1022.

[20] M. A. Youssef, A. Agrawala, and A. U. Shankar, "WLAN location determination via clustering and probability distributions," in *Proc. IEEE PERCOM'03*, Mar. 2003.

[21] Y. Gwon, R. Jain, and T. Kawahara, "Robust indoor location estimation of stationary and mobile users," in *Proc. IEEE INFOCOM'04*, Mar. 2004, pp. 1032–1043.

[22] Ekahau, "Ekahau – Wi-Fi RTLS and site survey solutions." [Online]. Available: http://www.ekahau.com/

[23] R. Battiti, M. Brunato, and A. Villani, "Statistical learning theory for location fingerprinting in wireless LANs," Universita di Trento, Dipartimento di Informatica e Telecomunicazioni, Tech. Rep. DIT-02-0086, Oct. 2002.

[24] R. Battiti, T. L. Nhat, and A. Villani, "Location-aware computing: a neural network model for determining location in wireless LANs," Universita di Trento, Dipartimento di Informatica e Telecomunicazioni, Tech. Rep. DIT-02-0083, Feb. 2002.

[25] P. Tao, A. Rudys, A. M. Ladd, and D. S. Wallach, "Wireless LAN location-sensing for security applications," in *Proc. ACM WiSe'03*, San Diego, CA, Sept. 2003, pp. 11–20.

[26] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proc. ACM MobiCom'04*, Philadelphia, PA, 2004, pp. 70–84.

[27] M. Hossain, H. Nguyen Van, Y. Jin and W.-S. Soh, "Indoor localization using multiple wireless technologies," in *Proc. IEEE MASS*, Pisa, Italy, Oct. 2007. [Online]. Available: http://www.ece.nus.edu.sg/stfpage/elesohws/mass07.pdf

[28] M. B. Kjærgaard, "Automatic mitigation of sensor variations for signal strength based location systems," in *Proc. Second Int. Workshop on Location and Context Awareness*, 2006.

[29] M. B. Kjærgaard and C. V. Munk, "A calibration-free solution for handling differences in signal strength," in *Proc. IEEE PerCom'08*, Mar. 2008.

[30] K. Kaemarungsi and P. Krishnamurthy, "Properties of indoor received signal strength for WLAN location fingerprinting," in *Proc. MobiQuitous'04*, San Diego, CA, 2004, pp. 14–23.

[31] C. Wang, H. Wu, and N.-F. Tzeng, "RFID-based 3-D positioning schemes," in *Proc. IEEE INFOCOM'07*, Anchorage, Alaska, May 2007, pp. 1235–1243.

[32] E. Elnahrawy, X. Li and and R. P. Martin, "Using area-based presentations and metrics for localization systems in wireless LANs," in *Proc. IEEE WLN*, Nov. 2004.

[33] J. Webr and C. Lanzl, "Designing a positioning systems for finding things and people indoors," *IEEE Spectrum*, vol. 35, no. 9, pp. 71–78, Sept. 1998.

[34] PlaceLab. [Online]. Available: http://www.placelab.org/

[35] A. LaMarca, Y. Chawathe, S. Consolvo1, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place lab: Device positioning using radio beacons in the wild," in *Proc. of Pervasive*, 2005, pp. 116–133.

[36] WiGLE, "Wigle - Wireless Geographic Logging Engine - plotting Wi-Fi on maps." [Online]. Available: http://www.wigle.net

[37] S. Saha, K. Chaudhuri, D. Sanghi, and P. Bhagwat, "Location determination of a mobile device using IEEE 802.11 access point signals," in *Proc. WCNC'03*, New Orleans, LA, 2003, pp. 1987–1992.

[38] SKYHOOK, "Skyhook wireless." [Online]. Available: http://www.skyhookwireless.com/

[39] Z. Weissman, "Indoor location white paper," Apr. 2004. [Online]. Available: www.tadlys.com/media/downloads/Indoor_location_Systems.pdf

[40] K. Pahlavan, X. Li, and J. Makela, "Indoor geolocation science and technology," *IEEE Comm Soc. Mag.*, Feb. 2002.

[41] J. Hightower, R. Want, and G.Borriello, "SpotON: An indoor 3D location sensing technology based on RF signal strength," University of Washington, Tech. Rep. UW CSE 00-02-02, Feb. 2000.

[42] D. Nicolescu and B. Nath, "DV based positioning in ad hoc networks," *Journal of Telecommunications Systems*, 2003.

[43] R. Nagpal, "Organizing a global coordinate system from local information on an amorphous computer," MIT A.I. Laboratory, Tech. Rep. A.I. Memo 1666, Aug. 1999.

[44] B. Fang, "Simple solutions for hyperbolic and related position fixes," *IEEE Trans. Aerospace and Elect. Sys.*, pp. 734–753, Sep. 1990.

[45] J. B. Marion and W. F. Hornyak, *Physics for Scientists & Engineers: Combined Volume*. Saunders College Publishing, 1982.

[46] P. Biswas, H. Aghajan, and Y. Ye, "Integration of angle of arrival information for multimodal sensor network localization using semi-definite programming," in *Proc. of* 39^{th} *Asilomar Conference on Signals, Systems and Computers*, Oct. 2005.

[47] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proc. IEEE INFOCOM'03*, San Francisco, CA, 2003.

[48] H. Wu, C. Wang, and N.-F. Tzeng, "Novel self-configurable positioning technique for multi-hop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 609–621, 2005.

[49] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, 2000.

[50] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes in large scale sensor networks," in *Proc. ACM/IEEE Mobicom'03*, 2003, pp. 81–95.

[51] N. Priyantha, A. Miu, H. Balakrishnan, and S. Teller, "The Cricket compass for context-aware mobile applications," in *Proc. ACM MobiCom'01*, Rome, Italy, July 2001.

[52] D. Pandya, R. Jain, and E. Lupu, "Indoor location using multiple wireless technologies," in *Proc. IEEE PIMRC*, Beijing, China, Sept. 2003, pp. 2208–2212.

[53] T. Roos, P. Myllymaki, H. Tirri, P. Misikangas, and J. Sievanen, "A probabilistic approach to WLAN user location estimation," *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, July 2002.

[54] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, 1994.

[55] V. N. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, NY, 1998.

[56] B. Schölkopf, "Statistical learning and kernel methods," Microsoft Research, Tech. Rep. MSR-TR-2000-23, 2000.

[57] A. K. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4–37, 2000.

[58] J. Bardwell, "Converting signal strength percentage to dBm values." [Online]. Available: http://www.wildpackets.com/elements/whitepapers/Converting_Signal_Strength.pdf

[59] ——, "A discussion clarifying often-misused 802.11 WLAN terminologies." [Online]. Available: http://madwifi-project.org/attachment/wiki/UserDocs/RSSI/you_believe_D100201.pdf

[60] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks: A Unified Approach*. Upper Saddle River, NJ: Prentice Hall PTR, 2001.

[61] A. Kotanen, M. Hännikäinen, H. Leppäkoski, and T. D. Hämäläinen, "Experiments on local positioning with Bluetooth," in *Proc. ITCC '03*, April 2003.

[62] A. Madhavapeddy and A. Tse, "A study of Bluetooth propagation using accurate indoor location mapping," in *Proc. UbiComp '05*, Tokyo, Japan, Sept. 2005, pp. 105–122.

[63] T. libpcap Project. [Online]. Available: http://sourceforge.net/projects/libpcap/

[64] BlueZ, "Official Linux Bluetooth protocol stack." [Online]. Available: http://www.bluez.org

[65] B. SIG, "Bluetooth core specification version 2.0 + enhanced data rate."

[66] T. public repository. [Online]. Available: http://www.tcpdump.org/

[67] C. S. Radio, "Bluecore4-ROM Bluetooth chipset." [Online]. Available: http://www.csr.com/products/bc4range.htm

[68] M. S. Gast, *802.11 Wireless Networks: The Definitive Guide*. O'Reilly & Associates Inc., 2002.

[69] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, "Ecolocation: a sequence based technique for RF localization in wireless sensor networks," in *Proc. ISPN'05*, Apr. 2005.

[70] H. L. V. Trees, *Optimum array processing: Part IV of detection, estimation and modulation theory*. John Wiley & Sons, Inc., 2002.

[71] Y. Qi, H. Suda, and H. Kobayashi, "On time-of-arrival positioning in a multipath environment," in *Proc. IEEE VTC'04*, Sept. 2004.

[72] A. Dersan and Y. Tanik, "Passive radar localization by time difference of arrival," in *Proc. MILCOM*, Oct. 2002, pp. 1251–1257.

[73] C. Chang and A. Sahai, "Estimation bounds for localization," in *Proc. IEEE SECON'04*, Oct. 2004, pp. 415–424.

[74] S. Venkatesh and R. M. Buehrer, "Multiple-access insights from bounds on sensor localization," *Pervasive and Mobile Computing*, vol. 4, no. 1, pp. 33–61, 2008.

[75] N. Patwari and A. O. Hero, "Location estimation accuracy in wireless sensor networks," in *Proc. IEEE Asilomar Conference on Signals and Systems*, Pacific Grove, CA, Nov. 2002, pp. 1523–1527.

[76] R. Battiti, M. Brunato, and A. Delai, "Optimal wireless access point placement for location-dependent services," Universita di Trento, Dipartimento di Informatica e Telecomunicazioni, Tech. Rep. DIT-03-052, 2003.

[77] Y. Chen, J.-A. Francisco, W. Trappe, and R. Martin, "A practical approach to landmark deployment for indoor localization," in *Proc. IEEE SECON'06*, 2006, pp. 365–373.

[78] T. T. Soong, *Fundamentals of Probability and Statistics for Engineers*. Wiley, 2004.

[79] S. M. Kay, *Fundamentals of Signal Processing, Volume I: Estimation Theory*, 2nd ed., 1993.

[80] B. Li, J. Salter, A. G. Dempster, and C. Rizos, "Indoor positioning techniques based on wireless LAN," in *1st IEEE Int. Conf. on Wireless Broadband & Ultra Wideband Communications*, Sydney, AUS, Mar. 2006.

[81] E. S. Bhasker, S. W. Brown, and W. G. Griswold, "Employing user feedback for fast, accurate, low-maintenance geolocationing," in *Proc. PerCom*, Mar. 2004, pp. 111–120.

[82] D. Papadogkonas, G. Roussos, and M. Levene, "Analysis, ranking and prediction in pervasive computing trails," in *Proc. IET Intelligent Environments (IE08)*, Seattle, July 2008.

[83] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proc. ACM WiSe '03*, NY, USA, 2003, pp. 1–10.

[84] D. Chaum, "Distance-bounding protocols (extended abstract)," in *In Theory and Application of Cryptographic Techniques*, 1993, pp. 344–359.

[85] D. Singelee and B. Preneel, "Location verification using secure distance bounding protocols," in *Proc. IEEE MASS*, Washington, USA, 2005.

[86] M. D. Corner and B. D. Noble, "Zero-interaction authentication," in *Proc. MobiCom '02*, New York, NY, USA, 2002, pp. 1–11.

[87] E. Elnahrawy, X. Li and and R. P. Martin, "The limits of localization using signal strength: A comparative study," in *IEEE SECON*, Santa Clara, CA, Oct. 2004.

List of Publications

1. A.K.M. Mahtab Hossain, Yunye Jin, Hien N. Van and Wee-Seng Soh, “An Indoor Positioning System Based on Robust Location Fingerprint for Wi-Fi and Bluetooth,” submitted to *IEEE/ACM Transactions on Networking*.

2. A.K.M. Mahtab Hossain, Hien N. Van and Wee-Seng Soh, “Utilization of User Feedback in Indoor Positioning System,” to appear, *Pervasive and Mobile Computing*, Elsevier.

3. A.K.M. Mahtab Hossain and Wee-Seng Soh, “Cramer-Rao Bound Analysis of Localization Using Signal Strength Difference as Location Fingerprint,” to appear in *IEEE INFOCOM*, San Diego, California, USA, Mar. 2010.

4. A.K.M. Mahtab Hossain, Hien N. Van and Wee-Seng Soh, “Fingerprint-based Location Estimation with Virtual Access Points,” *IEEE ICCCN*, St. Thomas, Aug. 2008.

5. A.K.M. Mahtab Hossain, Hien N. Van, Yunye Jin and Wee-Seng Soh, “Indoor Localization using Multiple Wireless Technologies,” *IEEE MASS*, Pisa, Italy, Oct. 2007.

6. A.K.M. Mahtab Hossain and Wee-Seng Soh, “A Comprehensive Study of Bluetooth Signal Parameters for Localization,” *IEEE PIMRC*, Athens, Greece, Sep. 2007.

Appendix A

Some Proofs for Chapter 5

A.1 Detailed Calculation of CRLB for Localization using SSD as Location Fingerprint

The joint p.d.f. (5.7) of the $(K-1)$ independent SSD measurements can be simplified as,

$$f_\theta(P) = \prod_{k=1}^{K-1} \frac{1}{\sqrt{2\pi}\widetilde{\sigma}} \frac{10}{\ln 10} \frac{p_r}{p_k} \exp\left\{ -\frac{\rho}{8} \left(\ln \frac{d_k^2}{\tilde{d}_{kr}^2} \right)^2 \right\}, \tag{A.1}$$

$$\begin{aligned} \text{where } \rho &= \left(\frac{10\beta}{\widetilde{\sigma} \ln 10} \right)^2 \\ \text{and } \quad \tilde{d}_{kr} &= d_r \left(\frac{p_r}{p_k} \right)^{\frac{1}{\beta}}. \end{aligned}$$

Consequently, the log-likelihood of (A.1) takes the form,

$$\begin{aligned}
\ln f_\theta(P) &= \sum_{k=1}^{K-1}\left[C_{kr} - \frac{\rho}{8}\left(\ln\frac{d_k^2}{\tilde{d}_{kr}^2}\right)^2\right], \\
\text{where } C_{kr} &= \frac{1}{\sqrt{2\pi}\widetilde{\sigma}}\frac{10}{\ln 10}\frac{p_r}{p_k}, \quad k = 1, 2, \ldots, (K-1) \\
&= \text{Constant w.r.t. } \theta.
\end{aligned}$$

To derive the entries of (5.8), we calculate the score function,

$$\begin{aligned}
U(\theta) &= \left\{\frac{\partial}{\partial x}\ln f_\theta(P), \frac{\partial}{\partial y}\ln f_\theta(P)\right\} \\
&= \left\{-\frac{\rho}{2}\sum_{k=1}^{K-1}\left(\ln\frac{d_k^2}{\tilde{d}_{kr}^2}\right)\left[\frac{(x-x_k)}{d_k^2} - \frac{(x-x_r)}{d_r^2}\right],\right. \\
&\quad \left.-\frac{\rho}{2}\sum_{k=1}^{K-1}\left(\ln\frac{d_k^2}{\tilde{d}_{kr}^2}\right)\left[\frac{(y-y_k)}{d_k^2} - \frac{(y-y_r)}{d_r^2}\right]\right\}.
\end{aligned}$$

Taking the derivative of $U(\theta)$ w.r.t. θ, we obtain,

$$\begin{aligned}
&\frac{\partial}{\partial x^2}\ln f_\theta(S) = -\frac{\rho}{2}\sum_{k=1}^{K-1}\left\{\left(\ln\frac{d_k^2}{\tilde{d}_{kr}^2}\right)\times\right. \\
&\left.\frac{\partial}{\partial x}\left[\frac{(x-x_k)}{d_k^2} - \frac{(x-x_r)}{d_r^2}\right] + 2\left[\frac{(x-x_k)}{d_k^2} - \frac{(x-x_r)}{d_r^2}\right]^2\right\}. \\
&\frac{\partial}{\partial y\partial x}\ln f_\theta(S) = \frac{\partial}{\partial x\partial y}\ln f_\theta(S) \\
&= -\frac{\rho}{2}\sum_{k=1}^{K-1}\left\{\left(\ln\frac{d_k^2}{\tilde{d}_{kr}^2}\right)\frac{\partial}{\partial y}\left[\frac{(x-x_k)}{d_k^2} - \frac{(x-x_r)}{d_r^2}\right]\right. \\
&\left.+ 2\left[\frac{(x-x_k)}{d_k^2} - \frac{(x-x_r)}{d_r^2}\right]\left[\frac{(y-y_k)}{d_k^2} - \frac{(y-y_r)}{d_r^2}\right]\right\}. \\
&\frac{\partial}{\partial y^2}\ln f_\theta(S) = -\frac{\rho}{2}\sum_{k=1}^{K-1}\left\{\left(\ln\frac{d_k^2}{\tilde{d}_{kr}^2}\right)\times\right. \\
&\left.\frac{\partial}{\partial y}\left[\frac{(y-y_k)}{d_k^2} - \frac{(y-y_r)}{d_r^2}\right] + 2\left[\frac{(y-y_k)}{d_k^2} - \frac{(y-y_r)}{d_r^2}\right]^2\right\}.
\end{aligned}$$

We see that, all the elements of derivatives of the score function depend on a term, $\left(\ln \frac{d_k^2}{d_{kr}^2}\right)$, which has an expected value of zero. Therefore, the entries of (5.8) becomes,

$$\begin{aligned}
J_{xx}(\theta) &= \rho \sum_{k=1}^{K-1} \left[\frac{(x-x_k)}{d_k^2} - \frac{(x-x_r)}{d_r^2} \right]^2 \\
&= \rho \sum_{k=1}^{K-1} \left[\frac{\cos\phi_k}{d_k} - \frac{\cos\phi_r}{d_r} \right]^2 . \\
J_{xy}(\theta) &= J_{yx}(\theta) \\
&= \rho \sum_{k=1}^{K-1} \left[\frac{(x-x_k)}{d_k^2} - \frac{(x-x_r)}{d_r^2} \right] \left[\frac{(y-y_k)}{d_k^2} - \frac{(y-y_r)}{d_r^2} \right] \\
&= \rho \sum_{k=1}^{K-1} \left[\frac{\cos\phi_k}{d_k} - \frac{\cos\phi_r}{d_r} \right] \left[\frac{\sin\phi_k}{d_k} - \frac{\sin\phi_r}{d_r} \right] . \\
J_{yy}(\theta) &= \rho \sum_{k=1}^{K-1} \left[\frac{(x-x_k)}{d_k^2} - \frac{(x-x_r)}{d_r^2} \right]^2 \\
&= \rho \sum_{k=1}^{K-1} \left[\frac{\sin\phi_k}{d_k} - \frac{\sin\phi_r}{d_r} \right]^2 .
\end{aligned}$$

Here, $\phi_k \in [0, 2\pi)$ is the angle the MN makes w.r.t. the k^{th} AP as illustrated in Fig. 5.1. Subsequently, the CRLB can be expressed as,

$$\{J(\theta)\}^{-1} = \frac{1}{|J(\theta)|} \begin{bmatrix} J_{yy}(\theta) & -J_{xy}(\theta) \\ -J_{yx}(\theta) & J_{xx}(\theta) \end{bmatrix}, \tag{A.2}$$
$$\text{where } |J(\theta)| = J_{xx}(\theta) \cdot J_{yy}(\theta) - J_{xy}(\theta) \cdot J_{yx}(\theta).$$

Suppose the variance of the location estimate of SSD-based localization with K APs is denoted as $\text{var}(\widehat{\theta})_K$. From the CRLB property (5.6), we know that, $\text{Cov}(\widehat{\theta}, \theta) \geq \{J(\theta)\}^{-1}$, i.e., the matrix $\text{Cov}(\widehat{\theta}, \theta) - \{J(\theta)\}^{-1}$ is positive semidefinite [70]. Since the diagonal elements of positive semidefinite matrices are larger or equal to zero, we obtain the following inequalities for any unbiased estimator using the identities of (5.3)

and (A.2),

$$\sigma^2_{\widehat{x}} \geq \frac{J_{yy}(\theta)}{|J(\theta)|} \quad \text{and} \quad \sigma^2_{\widehat{y}} \geq \frac{J_{xx}(\theta)}{|J(\theta)|}.$$

Consequently, we have,

$$\begin{aligned} \text{var}(\widehat{\theta})_K = \sigma^2_{\widehat{x}} + \sigma^2_{\widehat{y}} &\geq \frac{J_{xx}(\theta) + J_{yy}(\theta)}{|J(\theta)|} \\ &= \frac{\lambda_K}{\rho \cdot \eta_K}, \end{aligned} \tag{A.3}$$

where we define, $\lambda_K = \sum_{k=1}^{K-1}(u_k^2 + v_k^2)$, $\eta_K = \sum_{k=1}^{K-1} u_k^2 \sum_{k=1}^{K-1} v_k^2 - \left\{\sum_{k=1}^{K-1} u_k v_k\right\}^2$, $u_k = [\frac{\cos\phi_k}{d_k} - \frac{\cos\phi_r}{d_r}]$ and $v_k = [\frac{\sin\phi_k}{d_k} - \frac{\sin\phi_r}{d_r}]$.

A.2 Induction Proof of Inequality (5.13)

Suppose the inequality statement to be proven is denoted by $S(K)$.

Basis: It can be easily seen that $S(1)$ holds. Let us show that $S(2)$ holds too. For $K = 2$, the LHS of the inequality (5.13) can be simplified as,

$$\begin{aligned} (u_1^2 + u_2^2)(v_1^2 + v_2^2) - (u_1 v_1 + u_2 v_2)^2 &= (u_2 v_1 - v_2 u_1)^2 \\ &\geq 0. \end{aligned}$$

Therefore, $S(2)$ holds as well.

Inductive Step: Suppose $S(K)$ holds, i.e.,

$$\sum_{k=1}^{K} u_k^2 \sum_{k=1}^{K} v_k^2 - \left(\sum_{k=1}^{K} u_k v_k\right)^2 \geq 0. \tag{A.4}$$

Now, it must be shown that $S(K+1)$ holds, i.e.,

$$\sum_{k=1}^{K+1} u_k^2 \sum_{k=1}^{K+1} v_k^2 - \left(\sum_{k=1}^{K+1} u_k v_k \right)^2 \geq 0.$$

The LHS of the above inequality can be rewritten as,

$$\begin{aligned}
& \sum_{k=1}^{K+1} u_k^2 \sum_{k=1}^{K+1} v_k^2 - \left(\sum_{k=1}^{K+1} u_k v_k \right)^2 \\
= \; & (\sum_{k=1}^{K} u_k^2 + u_{K+1}^2)(\sum_{k=1}^{K} v_k^2 + v_{K+1}^2) - \left(\sum_{k=1}^{K} u_k v_k \right)^2 \\
& - 2 \left(\sum_{k=1}^{K} u_k v_k \right) u_{K+1} v_{K+1} - (u_{K+1} v_{K+1})^2 \\
= \; & \sum_{k=1}^{K} u_k^2 \sum_{k=1}^{K} v_k^2 - \left(\sum_{k=1}^{K} u_k v_k \right)^2 + u_{K+1}^2 \sum_{k=1}^{K} v_k^2 \\
& + v_{K+1}^2 \sum_{k=1}^{K} u_k^2 - 2 \left(\sum_{k=1}^{K} u_k v_k \right) u_{K+1} v_{K+1}.
\end{aligned}$$

Using the induction hypothesis (A.4) and the identity

$$u_{K+1}^2 \sum_{k=1}^{K} v_k^2 + v_{K+1}^2 \sum_{k=1}^{K} u_k^2 - 2 \left(\sum_{k=1}^{K} u_k v_k \right) u_{K+1} v_{K+1} \geq 0, \tag{A.5}$$

we can show that $S(K+1)$ holds indeed. The inequality (A.5) follows from:

$$\begin{aligned}
& (u_{K+1} v_k - v_{K+1} u_k)^2 \geq 0 \\
\Rightarrow \; & u_{K+1}^2 v_k^2 + v_{K+1}^2 u_k^2 - 2 u_{K+1} v_{K+1} u_k v_k \geq 0 \\
\Rightarrow \; & u_{K+1}^2 \sum_{k=1}^{K} v_k^2 + v_{K+1}^2 \sum_{k=1}^{K} u_k^2 \\
& - 2 \left(\sum_{k=1}^{K} u_k v_k \right) u_{K+1} v_{K+1} \geq 0.
\end{aligned}$$

A.3 Proof of Inequality (5.14)

Simplifying the LHS of the inequality (5.14), we obtain,

$$\begin{aligned}
& \sqrt{\left(\lambda_K \sum_{k=1}^{K-1} u_k^2 - \eta_K\right)\left(\lambda_K \sum_{k=1}^{K-1} v_k^2 - \eta_K\right)} \\
= & \sqrt{\lambda_K^2 \sum_{k=1}^{K-1} u_k^2 \sum_{k=1}^{K-1} v_k^2 - \eta_K \cdot \lambda_K \sum_{k=1}^{K-1} (u_k^2 + v_k^2) + \eta_K^2} \\
= & \sqrt{\lambda_K^2 \cdot \eta_K + \lambda_K^2 \left\{\sum_{k=1}^{K-1} u_k v_k\right\}^2 - \eta_K \cdot \lambda_K^2 + \eta_K^2} \\
= & \sqrt{\lambda_K^2 \left\{\sum_{k=1}^{K-1} u_k v_k\right\}^2 + \eta_K^2} \\
\geq & \lambda_K \sum_{k=1}^{K-1} u_k v_k, \quad \text{since } \eta_K^2 \geq 0. \qquad \text{(A.6)}
\end{aligned}$$

A.4 Proof of $\phi_{K+1} = \phi_r = \phi_k, \forall k \in \{1, 2, \ldots, K\} - \{r\}$ when $\mathcal{C}_K - \mathcal{C}_{K+1} = 0$

Here, we give the proof of the claim that, the equality conditions of both (A.6) and (5.15) result in the following, $\phi_{K+1} = \phi_r = \phi_k, \forall k \in \{1, 2, \ldots, K\} - \{r\}$.

The equality condition of (A.6) requires, $\eta_K^2 = 0$, i.e., $\eta_K = 0$. Consequently, from the definition of η_K, we can write,

$$\sum_{k=1}^{K-1} u_k^2 \sum_{k=1}^{K-1} v_k^2 = \left\{\sum_{k=1}^{K-1} u_k v_k\right\}^2. \qquad \text{(A.7)}$$

Using the identities of u_k and v_k, it can be deduced that, only when $\phi_k = \phi_r, \forall k \in \{1, 2, \ldots, K\} - \{r\}$, the LHS and RHS of (A.7) become equal.

Now, putting $\eta_K = 0$ into the equality condition of (5.15), we obtain,

$$v_{K+1}^2 \sum_{k=1}^{K-1} u_k^2 = u_{K+1}^2 \sum_{k=1}^{K-1} v_k^2. \tag{A.8}$$

Plugging the values of u_{K+1}, v_{K+1}, u_k and v_k into (A.8), we get,

$$\begin{aligned} &\left[\frac{\sin\phi_{K+1}}{d_{K+1}} - \frac{\sin\phi_r}{d_r}\right]^2 \sum_{k=1}^{K-1} \left[\frac{\cos\phi_k}{d_k} - \frac{\cos\phi_r}{d_r}\right]^2 \\ &= \left[\frac{\cos\phi_{K+1}}{d_{K+1}} - \frac{\cos\phi_r}{d_r}\right]^2 \sum_{k=1}^{K-1} \left[\frac{\sin\phi_k}{d_k} - \frac{\sin\phi_r}{d_r}\right]^2. \end{aligned} \tag{A.9}$$

Putting $\phi_k = \phi_r, \forall k \in \{1, 2, \ldots, K\} - \{r\}$ (derived from the equality condition of (A.6) above) into (A.9), we have,

$$\begin{aligned} &\cos^2\phi_r \left[\frac{\sin\phi_{K+1}}{d_{K+1}} - \frac{\sin\phi_r}{d_r}\right]^2 \\ &= \sin^2\phi_r \left[\frac{\cos\phi_{K+1}}{d_{K+1}} - \frac{\cos\phi_r}{d_r}\right]^2. \end{aligned} \tag{A.10}$$

Simplifying (A.10), it can be easily seen that, $\phi_{K+1} = \phi_r$. Combining this result with $\phi_k = \phi_r, \forall k \in \{1, 2, \ldots, K\} - \{r\}$, we finally obtain, $\phi_{K+1} = \phi_r = \phi_k, \forall k \in \{1, 2, \ldots, K\} - \{r\}$.

A.5 Proof of Inequality (5.20)

Simplifying the LHS of the above inequality we get,

$$(u_{K'}v_{K'} - u_K v_K)\sqrt{\lambda_{K-1}^2 \sum_{k=1}^{K-2} u_k^2 \sum_{k=1}^{K-2} v_k^2 - \eta_{K-1}\cdot\lambda_{K-1}\left\{\sum_{k=1}^{K-2}(u_k^2 + v_k^2)\right\} + \eta_{K-1}^2}$$

$$= (u_{K'}v_{K'} - u_K v_K)\sqrt{\lambda_{K-1}^2\left[\eta_{K-1} + \left\{\sum_{k=1}^{K-2} u_k v_k\right\}^2\right] - \eta_{K-1}\cdot\lambda_{K-1}^2 + \eta_{K-1}^2}$$

$$\geq \lambda_{K-1}(u_{K'}v_{K'} - u_K v_K)\left\{\sum_{k=1}^{K-2} u_k v_k\right\}, \text{ since, } \eta_{K-1}^2 \geq 0.$$

Appendix B

Calculation of RSS at Fictitious Training Points

Suppose, there are n training points for which the real measurements of RSSs have been taken at the K APs. Our goal is to emulate the RSSs of K APs for J possible interpolated training points utilizing those real measurements.

First, we consider calculating the regression coefficients which were introduced in Section 6.1.1 in order to formulate the average RSS of the k^{th} AP for a particular fictitious point j. From Section 6.1.1, we know that, the linear regression RSS prediction formula takes the following form,

$$\hat{y}_{ki} \quad = \quad a_k x_{ki} + b_k, \tag{B.1}$$

where, $\hat{y}_{ki}$ = the predicted RSS of the k^{th} AP when the MN is at i^{th} training point, $a_k = -10\beta,\ x_{ki} = \log(d_{ki})$ and $b_k = P(d_0)|_{\text{dBm}} + 10\beta \log(d_0)$.

Now, we consider calculating the regression coefficients, i.e., a_k and b_k of (B.1) in order to formulate the RSS of the k^{th} AP for a particular fictitious training point j.

Utilizing the spatial similarity of RSS samples which suggests that the closer training points should contribute more in formulating the interpolated RSS, the weighted least mean square minimization function for our linear regression model can be written as,

$$R^2(a_{jk}, b_{jk}) = \sum_{i=1}^{n} w_{ji}[y_{ki} - (a_{jk}x_{ki} + b_{jk})]^2, \tag{B.2}$$

where y_{ki} = real measurement of RSS at the k^{th} AP when the MN is at i^{th} training point, w_{ji} = normalized weight considering spatial similarity of RSS = $\frac{1/d_{ji}}{\sum_{i=1}^{n} 1/d_{ji}}$, d_{ji} = distance of fictitious point j from the i^{th} training point, $x_{ki} = \log(d_{ki})$ = log distance of k^{th} AP from the i^{th} training point, $a_{jk},\ b_{jk}$ = regression coefficients of the linear RSS prediction formula of the k^{th} AP for j, $k \in \{1, 2, \ldots, K\}$, and $j \in \{1, 2, \ldots, J\}$. Note that, depending on the fictitious point j, for which the RSS will be predicted, the associated weight (i.e., w_{ji}) changes for the RSS perceived at an AP corresponding to different real training points. Hence, an additional subscript is used in (B.2) to denote the regression coefficients for an AP w.r.t. different fictitious points compared to (B.1).

Denote,

$$\mathcal{Y}_k = \begin{bmatrix} y_{1k} \\ y_{2k} \\ \vdots \\ y_{nk} \end{bmatrix}, \quad \mathcal{X}_k = \begin{bmatrix} 1 & x_{1k} \\ 1 & x_{2k} \\ \vdots & \vdots \\ 1 & x_{nk} \end{bmatrix}, \quad \mathcal{W}_j = \begin{bmatrix} w_{j1} & 0 & 0 & \ldots & 0 \\ 0 & w_{j2} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & w_{jn} \end{bmatrix} \text{ and } \mathcal{B}_{jk} = \begin{bmatrix} b_{jk} \\ a_{jk} \end{bmatrix}.$$

Using these matrix notations, now we differentiate (B.2) w.r.t. $\mathcal{B}_{jk}$ and set it to zero,

$$\begin{aligned} & \frac{\partial}{\partial \mathcal{B}_{jk}} \left[(\mathcal{Y}_k - \mathcal{X}_k \mathcal{B}_{jk})^T \mathcal{W}_j (\mathcal{Y}_k - \mathcal{X}_k \mathcal{B}_{jk}) \right] = 0 \\ \Rightarrow \quad & \frac{\partial}{\partial \mathcal{B}_{jk}} \left[{\mathcal{Y}_k}^T \mathcal{W}_j \mathcal{Y}_k - {\mathcal{B}_{jk}}^T {\mathcal{X}_k}^T \mathcal{W}_j \mathcal{Y}_k - {\mathcal{Y}_k}^T \mathcal{W}_j \mathcal{X}_k \mathcal{B}_{jk} + {\mathcal{B}_{jk}}^T \mathcal{X}^T \mathcal{W}_j \mathcal{X}_k \mathcal{B}_{jk} \right] = 0 \end{aligned}$$

$$\Rightarrow \quad \frac{\partial}{\partial \mathcal{B}_{jk}} \left[{\mathcal{Y}_k}^T \mathcal{W}_j \mathcal{Y}_k + {\mathcal{B}_{jk}}^T {\mathcal{X}_k}^T \mathcal{W}_j \mathcal{X}_k \mathcal{B}_{jk} - 2{\mathcal{Y}_k}^T \mathcal{W}_j \mathcal{X}_k \mathcal{B}_{jk} \right] = 0$$

$$\Rightarrow \quad {\mathcal{B}_{jk}}^T {\mathcal{X}_k}^T \mathcal{W}_j \mathcal{X}_k = {\mathcal{Y}_k}^T \mathcal{W}_j \mathcal{X}_k$$

$$\Rightarrow \quad \mathcal{X}^T {\mathcal{W}_j}^T \mathcal{X}_k \mathcal{B}_{jk} = {\mathcal{X}_k}^T {\mathcal{W}_j}^T \mathcal{Y}_k.$$

If the matrix $\left(\mathcal{X}^T {\mathcal{W}_j}^T \mathcal{X}_k\right)$ is non-singular, the regression coefficients are given by the formula,

$$\mathcal{B}_{jk} = \left(\mathcal{X}^T {\mathcal{W}_j}^T \mathcal{X}_k\right)^{-1} {\mathcal{X}_k}^T {\mathcal{W}_j}^T \mathcal{Y}_k. \tag{B.3}$$

For a particular fictitious point j, the regression coefficients $\mathcal{B}_{jk}$ of the k^{th} AP's signals can be obtained through (B.3). Consequently, the RSS of the k^{th} AP for a fictitious point j can be emulated as,

$$\text{RSS}_{jk} = a_{jk} \log d_{jk} + b_{jk}. \tag{B.4}$$

Plugging in the values of a_{jk}, b_{jk} and d_{jk} (the distance of the fictitious point j from k^{th} AP) into (B.4), we finally obtain the RSS fingerprint for j considering only AP k. To deduce the RSS vector comprising of all the K APs for a particular fictitious point j,

Appendix C

Utilization of User Feedback

Here, we show how user feedback's credibility (i.e., weight w) is taken into account to generate the fictitious training point's RSS. Suppose there are n user feedbacks for which the real measurements of RSSs have been taken at the K APs. Similar to Appendix B, our goal is to emulate the RSSs of K APs for J possible interpolated training points utilizing those real measurements of user feedbacks.

Utilizing both the spatial similarity and user feedback credibility factors, the weighted least mean square minimization function for our linear regression model can be written as,

$$R^2(a_{jk}, b_{jk}) = \sum_{i=1}^{n} c_{ji}[y_{ki} - (a_{jk}x_{ki} + b_{jk})]^2. \tag{C.1}$$

All the symbols of (C.1) have the usual meaning as in Appendix B apart from the composite weight, c_{ji}, which is defined as, $c_{ji} = \frac{u_{ji} \times v_i}{\sum_{i=1}^{n} u_{ji} \times v_i}$, $u_{ji} =$ normalized weight considering spatial similarity of RSS $= \frac{1/d_{ji}}{\sum_{i=1}^{n} 1/d_{ji}}$, v_i $=$ normalized weight for i^{th} feedback considering its credibility $= \frac{w_i}{\sum_{i=1}^{n} w_i}$, d_{ji} $=$ distance of interpolated point j from the i^{th} training point.

Denote,

$$\mathcal{Y}_k = \begin{bmatrix} y_{1k} \\ y_{2k} \\ \vdots \\ y_{nk} \end{bmatrix}, \ \mathcal{X}_k = \begin{bmatrix} 1 & x_{1k} \\ 1 & x_{2k} \\ \vdots & \vdots \\ 1 & x_{nk} \end{bmatrix}, \ \mathcal{C}_j = \begin{bmatrix} c_{j1} & 0 & 0 & \dots & 0 \\ 0 & c_{j2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & c_{jn} \end{bmatrix} \text{ and } \mathcal{B}_{jk} = \begin{bmatrix} b_{jk} \\ a_{jk} \end{bmatrix}.$$

Carrying out similar calculations as in Appendix B, the regression coefficients are given by the formula,

$$\mathcal{B}_{jk} = \left(\mathcal{X}^T {\mathcal{C}_j}^T \mathcal{X}_k\right)^{-1} {\mathcal{X}_k}^T {\mathcal{C}_j}^T \mathcal{Y}_k. \tag{C.2}$$

For a particular interpolated point j, the regression coefficients $\mathcal{B}_{jk}$ of the k^{th} AP's signals can be obtained through (C.2). Consequently, the RSS of the k^{th} AP for an interpolated point j can be emulated as,

$$\text{RSS}_{jk} = a_{jk} \log d_{jk} + b_{jk}. \tag{C.3}$$

Plugging the values of a_{jk}, b_{jk} and d_{jk} (the distance of the interpolated point j from k^{th} AP) into (C.3), we finally obtain the RSS fingerprint for j considering only AP k. To deduce the RSS vector comprising of all the K APs for a particular interpolated point j, we have to follow the same procedure for all $k \in \{1, 2, \dots, K\}$. Finally, in order to obtain the RSS vector of the K APs for all the J interpolated points over the localization area, we have to repeat the whole calculation of this section for all $j \in \{1, 2, \dots, J\}$. Note that, when all user feedbacks are believed equally, we have, $c_{ji} = \frac{u_{ji} \times \frac{1}{n}}{\sum_{i=1}^{n} u_{ji} \times \frac{1}{n}} = \frac{u_{ji}}{\sum_{i=1}^{n} u_{ji}} = u_{ji}$. In other words, only spatial similarity weight factor would be taken into consideration in calculating the RSS signatures of the interpolated points which yields the exact same scenario as in Appendix B.