WI-FI-BASED INDOOR POSITIONING USING HUMAN-CENTRIC COLLABORATIVE FEEDBACK

YAN LUO

Wi-Fi-Based Indoor Positioning Using Human-Centric Collaborative Feedback

by

© Yan Luo

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Master of Sciences

Department of Computer Science

Memorial University of Newfoundland

May 2011

St. John's Newfoundland

Abstract

Position information is an important aspect of a mobile device's context. While GPS is widely used to provide location information, it does not work well indoors. Wi-Fi network infrastructure is found in many public facilities and can be used for indoor positioning. In addition, the ubiquity of Wi-Fi-capable devices makes this approach especially cost-effective.

In recent years, "folksonomy"-like systems such as Wikipedia or Delicious Social Bookmarking have achieved huge successes. User collaboration is the defining characteristic of such systems. For indoor positioning mechanisms, it is also possible to incorporate collaboration in order to improve system performance, especially for fingerprinting-based approaches.

In this thesis, a robust and efficient model is devised for integrating human-centric collaborative feedback within a baseline Wi-Fi fingerprinting-based indoor positioning system. Experiments show that the baseline system performance (i.e., positioning accuracy and precision) is improved by collecting both positive and negative feedback from users. Moreover, the feedback model is robust with respect to malicious feedback, quickly self-correcting based on subsequent helpful feedback from users.

Acknowledgements

Foremost, I would like to thank my supervisors sincerely, Dr. Yuanzhu Chen and Dr. Orland Hoeber, for their continuous support, guidance, and insightful advice. This thesis would not have been possible without the generous resources provided by the Department of Computer Science and Memorial University. My research is supported by NSERC Discovery Grants and scholarship from the University's the School of Graduate Studies.

Last but not least, my acknowledgement and love also go to my parents, Yang Luo and Huiyan Zhu, who have brought me to this wonderful world, taught me to appreciate everything I have, directed me through ups and downs, and stood by me at all time. I must ensure all these parties that I have had a tremendously rewarding experience and unforgettable time at Memorial University.

(Part of this research will appear in IEEE International Conference on Communications 2011 [28].)

Contents

List of Figures

Chapter 1

Introduction

Mobile devices have a unique attribution when compared to other fixed computing devices, which is their mobility. Thus, the position information can be a very important aspect of a mobile device's context. Based on this extra attribute, we are able to provide mobile device users with a special type of intelligent services, called location-aware services.

Traditionally, location-aware services have been confined to outdoor environments. Relatively less research has explored the potential applicability of similar services for indoor settings. However, the indoor location-aware services could also have a very promising application prospect. In this chapter, we will introduce the motivation for conducting research on improving indoor positioning.

1.1 Pervasive and Mobile Computing

A mobile device is typically a pocket-sized yet powerful computing platform. While there are a number of different aspects between using a mobile device and using a desktop/laptop computer, mobility is the most essential characteristic of mobile devices. With such a unique feature, mobile device users have the opportunity to access intelligent services (e.g., the Internet or cellular networks) ubiquitously. This has been a goal of the industry and academia and also a desire of the users for many years.

In order to offer flexible and adaptive services and improve the quality of lives, researchers have recently started to focus on location-aware intelligent services, which provides personalized services based on users' current or past locations. After over a decade of research and development, location-aware services (e.g., navigation, location-based weather reporting, and advertisements) have gradually penetrated into real life. Now, the location-aware services are expected to be one of the most promising technologies in the next few years because it assists human activities in a wide range of applications, from productivity and goal fulfillment to social networking and entertainment. It is also predicted that the location-aware service user base will grow globally from 96 million in 2009 to more than 526 million in 2012 [37].

Traditionally, location-aware applications have been confined to outdoor environments, mostly using the Global Positioning System (GPS). Relatively less research has explored the potential applicability of similar services for indoor settings. However, in large indoor environments such as airports, libraries, or shopping centres, location-awareness can improve the user experiences with these facilities.

For example, suppose some tourists wish to visit a large museum, they can prepare their visitation plan by first selecting the most interesting exhibits. Such a visitation plan can be manipulated in their mobile devices. When in the museum, the device can be connected to an indoor location system. Thanks to its location-awareness, the devices can provide them with personalized guided tours. Thus, the navigation through the museum will be an enhanced experience since the multimedia description and comments can be delivered to them automatically based on their positions and personal preferences. If one tourist stays in front of a painting, the intelligent guide can provide additional information about this piece such as the artist's biography, style, cultural context, etc. Furthermore, by monitoring visitors' navigation patterns and by instantly becoming aware of congestion spots within the museum, the museum administrators are able to organize exhibits more effectively or even direct other visitors to areas of low congestion [13].

1.2 Mobile Device Indoor Positioning

Indoor location-aware services can be very promising and have been researched for around two decades [16]. However, we still have not seen any product used nearly as widely as GPS-based positioning devices. The lack of development on the indoor aspect of this problem is a result of two technical challenges. First, GPS can not be deployed for indoor use because GPS signals can not reach indoor receivers. Second and more importantly, due to complicated indoor environments such as building geometry, the movement of people, and the random effects of signal propagation, triangulation-based approaches (i.e., those used for GPS) are much less effective [24]. In addition, interference and noise from other devices can also degrade the accuracy of positioning. On the other hand, such challenges provide researchers with great opportunities for innovative indoor positioning techniques.

An early approach for indoor positioning used infrared sensors [42]. In this type of positioning system, multiple infrared receivers are deployed in a building, and a mobile device with an infrared emitter transmits signals to determine its current position. However, infrared signals can be easily blocked by obstacles like furnishings and cubical walls, which may reduce the system performance. Other systems have exploited techniques such as short-range radios, ultrasonic waves, radio-frequency identification (RFID), magnetic technology, etc. Some of them have achieved fairly good accuracy and precision in field tests [16, 14]. However, the common disadvantages of these approaches are the size, complexity, and cost, which render them infeasible for mobile devices.

1.3 Wi-Fi-Based Mobile Device Indoor Positioning

A number of researchers have been working on using Wi-Fi infrastructure for indoor positioning even though Wi-Fi was not specifically designed for this purpose. Unlike the solutions mentioned above, this approach has a unique advantage of requiring only a few Wi-Fi routers (access points (APs)), and utilizing the existing wireless networking infrastructure of a building. This feature is very important for popularizing a ubiquitous indoor localization system since most current mobile devices include an integrated Wi-Fi chip.

Due to the infeasibility of indoor triangulation, most indoor positioning systems use a fingerprinting approach based on the *received signal strength* (RSS) transmitted by Wi-Fi APs [2]. Typically, such an approach consists of a training phase and a positioning phase. In the training phase, each survey position (with known physical coordinates) is characterized by location-related Wi-Fi RSS properties called *Wi-Fi RSS fingerprints* [20]. During the positioning phase, the positioning likelihood is calculated based on the current Wi-Fi RSS measurement. That is, the system estimates the position by comparing the current RSS measurement to the fingerprints in the system to generate the best match. Compared to distance estimation based on signal propagation models, such an approach is more robust and accurate in real indoor environments. However, fine-grained system training is normally required to achieve high accuracy and resolution. Also, the maintenance cost can be very high in order to continuously adapt to environment changes and Wi-Fi infrastructure alteration. A great deal of effort has been made by researchers to reduce such costs. An efficient way is to let users provide feedback to facilitate the construction and maintenance of the RSS fingerprints database. If the whole positioning process can be conducted in a collaborative manner, an user can take the advantage of position information shared by other users.

1.4 Research Question

In recent years, "folksonomy"-like systems (e.g., Wikipedia, YouTube, Flickr, and Delicious Social Bookmarking) have achieved huge successes. Such a kind of user-generated online content has gradually became a new way of generating and maintaining information. User collaboration is their defining characteristic. Nov [30] believes that the motivation of contributors in such systems is essential since the content is contributed by volunteers who offer their time, knowledge, and talent in return for no material reward. Thus, it is important to first understand and identify those volunteers' potential motivations. Several typical motivations revealed in [30] are listed as follows:

- Volunteering is an effective way for people to express humanity and self satisfaction. Participants show their concerns to others by sharing knowledge.
- Volunteering may provide people more opportunities to be engaged in valuable social activities and obtain pleasure via the interaction with others.
- Through volunteering, people may have more chances to practice their knowledge, skills, and abilities. They will obtain the feeling of fulfillment when their work receives positive feedback.
- Volunteering is also beneficial for participants' careers. Such user collaborative systems can be considered as an effective medium via which contributors are able to demonstrate their skills and abilities to future employers.

Following the same rationale, we expect that users are also willing to provide feedback to a positioning system in most cases. In terms of indoor positioning, systems may occasionally deliver inaccurate and unreliable results. In these circumstances, adding a compensation mechanism to modify the results can improve the robustness. Since the purpose for an indoor positioning system is to provide users with fast and accurate position estimation and location-aware services, soliciting assistance from end users could also be a good aspect for improvement [27]. The system performance could be improved if users are involved as a part of the system and conduct positioning tasks in a collaborative manner. This leads to the fundamental research question in this thesis:

> **What is the benefit of adding human-centric feedback to an indoor positioning system?**

Modern mobile devices have well-designed user interfaces to facilitate interaction with users. Similar to range finders used in the localization of autonomous robotics, humans are also able to "detect" the surrounding indoor environment using their senses and feed this information to devices. They are able to estimate their positions based on their perception. In order to utilize such estimations from end users, we need to define an effective user feedback model which is able to incorporate user feedback within a Wi-Fi RSS fingerprinting system.

Our user feedback model is derived from an existing relevance feedback mechanism from the domain of information retrieval, popularized by Salton's SMART system [35]. The basic idea of relevance feedback is to do an initial query, then obtain feedback from the user as to what documents are relevant or non-relevant, and then use the contents of these known relevant documents to generate subsequent queries. Similarly, we can consider a positioning system as an information retrieval system. Users initially query the positioning system and then provide feedback based on the returned estimates. The system incorporates the user feedback and modifies the search process to re-weight search results based on users' collaborative feedback.

1.5 Organization of Thesis

The reminder of this thesis is organized as follows. We discuss related work in Chapter 2. In Chapter 3, we describe a baseline Wi-Fi fingerprinting framework. Next, the detailed user feedback model is explained and interpreted in Chapter 4. We have built a prototype to evaluate the baseline system and the proposed user feedback model. The system design of this prototype is documented in Chapter 5. The user feedback model is tested and evaluated in comparison to the baseline method as reported in Chapter 6. This thesis is concluded in Chapter 7 with discussion and an overview of future work.

Chapter 2

Related Work

Different positioning systems have been built to provide different types of position information, which can be either absolute coordinates or logical location information (e.g., room No.). The enabling positioning technologies have their characteristics in architecture, performance, working field, and cost. Thus, in order to satisfy different types of user requirements, it is important to analyze the evaluation metrics and taxonomies of positioning technologies. In terms of indoor positioning, simply extending the outdoor positioning technologies to indoor environments is not feasible due to the complexity of indoor environments. In this chapter, we will introduce the work archived by other researchers to overcome the challenges in indoor positioning.

2.1 An Overview of Indoor Positioning

Indoor positioning and navigation have been an active area of research for the past two decades, with early research focusing on robot localization and navigation [39] and more recently pervasive and mobile computing [16]. Compared to outdoor positioning systems (e.g., GPS), the work area of an indoor positioning systems focus on indoor environments such as inside airports or shopping centres. Typically, indoor positioning systems can provide three different kinds of location information (i.e., absolute, relative, and proximity location information [14]) for location-aware services required by different usages. Absolute location information in the form of coordinates is normally required by indoor tracking systems or indoor navigation systems because real-time tracking and navigation services need precise physical coordinates of the targets. The relative location information measures the motion of different parts of the tracking target, e.g., detecting whether or not two mobile devices are in the same room. The proximity or logical location information is also an important type of information, which is usually in the form of logical labels or tags (e.g., office number.). A very interesting application of logical location information is location-aware advertising. For example, suppose a customer is nearby a shop. An electronic advertisement can be sent out for new products or discount information at that shop.

2.1.1 Indoor Positioning Technologies

Triangulation is the most used positioning technology for both indoor and outdoor environments. *Time of Arrival* (TOA), *Time Difference of Arrival* (TDOA), or *Angle of Arrive* (AOA) [40] are broadly used for outdoor positioning (e.g., GPS [31]) and are able to obtain good system performance in free space. The fundamental idea of triangulation is depicted in Figure 2.1.

Suppose the physical coordinates of three anchor points are known. The distance between an anchor point and the tracking target can be calculated via the time dif-

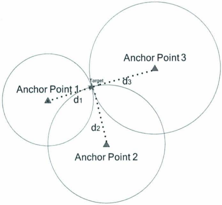


Figure 2.1: Basic idea of triangulation

ference between a transmitter and a receiver or using signal path-loss propagation models. Once the relative distances d_1, d_2, and d_3 are calculated, the position of the tracking target can be estimated using either the intersection area of the circles or the directions of the formed triangle [17]. However, due to complexity of indoor environments, these typical outdoor triangulation approaches might not be conveniently adapted to indoor environments, which makes the research of indoor positioning challenging.

After nearly two decades of research and development, numerous indoor positioning systems have been proposed by different companies, research centres, and universities [14]. Some researchers employ existing triangulation-based approaches via densely deploying infrared or ultrasonic sensors in a building. As such, the triangulation process can be conducted only in a small area (e.g., in a single office) to reduce the negative effect of complex indoor environments [16]. As a result, the system performance and robustness can be improved, but at an increased cost of installation and maintenance. Besides these modifications to classic techniques, researchers have also devised novel approaches such as location-fingerprinting [2] and vision analysis [17], which are relatively cost-effective and more robust.

Other technologies such as RFID, Wi-Fi, Bluetooth, sensor networks, ultra-wideband (UWB), and magnetic signals [14] have been developed to provide indoor location information. Each system takes advantage of a particular positioning technology or combining some of these. Usually, there is a trade-off between the price and the performance. A system with higher performance could have high complexity and cost. The designers should always strike the balance between the overall performance and the complexity.

2.1.2 Criteria of Evaluating Indoor Positioning Systems

Different indoor positioning technologies have their advantages and disadvantages in certain aspects. In order to satisfy a variety of user requirements, we should choose the indoor positioning system with the most suitable capabilities. Thus, it is very important to comprehensively evaluate an indoor positioning system from different aspects. According to Gu et al. [14], indoor positioning systems can be evaluated via the following important system performance and deployment criteria

- *Performance*: The accuracy and precision are two main performance parameters. The accuracy means the average error distance over all test points, and the precision is defined as the success probability of position estimations with respect to a predefined accuracy (e.g., $80th$ percentile positioning error within 2m). In fact, different location-aware services have different accuracy and precision requirements. For example, a 5m accuracy (room level) will suffice most of indoor location-aware services but the location-based guide in a museum might need at least $90th$ percentile error within 1m to locate an exhibit. The time consumed in the positioning process is another very important parameter to evaluate an indoor positioning system, especially for tracking and navigation services. A long positioning delay will degrade the user experience and the perceived service quality. Thus near-instantaneous responses to users' positioning queries is normally desired.

- *Cost*: The cost of an indoor positioning includes two aspects: the cost of the infrastructure installation and future maintenance, and the cost of positioning terminals (devices). In fact, high indoor positioning accuracy can always be obtained if a massive number of sensors or anchor points are deployed, but often we can not afford such a high deployment and maintenance cost. For the device or terminal used in positioning, it could be very inconvenient for users to carry a specialized device for their indoor positioning activities. Thus, an ideal solution to indoor positioning is to utilize the existing infrastructure and mobile devices at hand without any extra hardware costs.

- *Robustness and fault tolerance*: Indoor positioning systems are relatively less reliable due to large interference in their working areas. Also the alteration of positioning infrastructure could cause large positioning errors. The positioning system should be robust with respect to complex environments.

- *Security and privacy*: End users normally want their privacy to be protected when using computer systems. For positioning system users, they do not want to be tracked or have their history of past locations accessible by other users to whom they have not given prior permission. Security and privacy should be considered both during system architecture design and implementation. For stand-alone indoor positioning systems, the position calculation process is conducted locally, which ensures that no one can access the information. In contrast, client-server architectures may have more channels to expose user information. Thus, some security mechanisms such as secure data transfer, authorization, and access control are required to offer a high degree of security and privacy protection for users.

2.1.3 Taxonomies of Indoor Positioning System

We categorize indoor positioning systems mainly according to whether they are based on an existing infrastructure or specialized indoor positioning infrastructure. Also, autonomous robotics indoor positioning has a unique research problem domain, which should also be considered as a separate category.

Robotics — For an autonomous robot to navigate through indoor environments, it must have the ability to detect the current environment (using outer sensors, e.g., ultrasonic, camera, or laser) and calculate its movement trajectory (using inner movement sensors, e.g., wheel sensors) [39]. Initial approaches provisioned a robot with a pre-built map of the indoor environment, allowing it to determine its location by comparing its observed environment to the landmarks on the map and generate a belief distribution. Based on the movement trajectory calculated by inner sensors, the robot can filter out locations with low belief. As more and more low belief locations are eliminated, the robot can be localized at locations with high belief. Another significant step in the area of robotics was Simultaneous Localization and Mapping (SLAM) [39], which allows a robot to build a map of the indoor environment (in terms of the features of the environments) while simultaneously determining its location with respect to the map constructed in real-time.

The robot indoor localization is the core part of autonomous robotics. It is able to archive centimetre-level accuracy and high precision level. However, this technology is complex and expensive both in computation and the implementation of positioning module.

Extra infrastructure-based — Early ideas for indoor mobile entities positioning relied on deploying specialized infrastructure, mostly using infrared or ultrasonic signals. In such systems, infrared or ultrasonic sensors are installed on walls or ceilings in a build. Users typically wear tags in order to interact with these sensors. Once at least three sensors are in sight, triangulation approaches can be applied to estimate user's positioning.

Active Badge [42] and *Cricket* [32] are two representative indoor positioning systems using infrared and ultrasonic infrastructure, respectively. In Active Badge, one or more sensors are deployed in each located place such as a room, which is used to detect the infrared signal from an active badge carried by users. The position of the active badge can be detected by these fixed sensors receiving the infrared signal. Then, the data collected by sensors will be forwarded to central servers to generate the proximity information (e.g., room number). However, in order to cover a large indoor area, the infrared receivers need to be densely installed and connected to each other with wired or wireless networks.

Cricket [32] utilizes ultrasonic emitters as infrastructure. These emitters are deployed on walls or ceilings with known positions. They emit ultrasonic and also radio frequency messages with proximity location information (in case there are not enough ultrasonic emitters in sight). Compared to Active Badge, the tag carried by user is not an emitter but a receiver, once at least three mounted emitters are in sight, the location of the user's tag can be estimated via triangulation with a very high accuracy. More importantly, the triangulation is conducted on each tag locally, which can protect user privacy.

Although the dedicated positioning infrastructure can obtain a high positioning accuracy, the expensive system hardware requirements raise the system cost. Also, users normally need to wear specialized badges in order to be tracked by the sensors in the infrastructure.

Existing infrastructure-based — For years, the goal of indoor positioning has been to improve the system performance and to reduce the cost at the same time. The existing infrastructure-based indoor positioning is a promising research direction for cost reduction.

Wi-Fi and Bluetooth technologies are widely used and integrated in various electronic devices. Thus, the Wi-Fi or Bluetooth based positioning systems can also reuse these mobile devices as tracking targets to locate users, which is a less intrusive way to provide location-aware services to users.

In Bluetooth-based positioning systems, the position of a Bluetooth mobile device can be located via the signal strength transmitted by other mobile terminals in the same piconet (a master device and associated slave devices). However, a connection between the master mobile terminal and slave mobile terminals need to be created before it can obtain the signal strength, which will significantly increase the positioning delay. Compared to Bluetooth, the Wi-Fi infrastructure is more common and has been deployed in many public areas such as hospitals, airports, universities, etc. Furthermore, Wi-Fi mobile devices only need to receive beacon frames from APs without AP association (i.e., connecting to a Wi-Fi router). Such a listener-based mechanism makes Wi-Fi based positioning more convenient and secure.

In the research area of Wi-Fi-based indoor positioning, two fundamentally different approaches have been proposed: Wi-Fi triangulation-based and RSS fingerprinting-based. Wi-Fi triangulation is based on distance estimation. If we use the same time-of-flight method (e.g., TOA in ultrasonic sensors) to measure short distance on wireless waves, time measurements must be very accurate in order to avoid large uncertainties. However, it is difficult to measure the time-of-flight of Wi-Fi signal propogation in indoor environments because the signal travel distance is small. An alterative is to measure the distance based on a signal propagation model. The energy of the radio signal, viewed as an electromagnetic wave, attenuates as it propagates in space. The distance can be calculated via various signal propagation models. However, as mentioned before, the complex indoor environment introduces random fading effects. Although such effects can be reduced to a degree if anchors or sensors are densely deployed (e.g., multiple sensors in a room), it is not feasible for existing Wi-Fi infrastructure.

Instead, the Wi-Fi fingerprinting does not need to know the specific signal propagation model or AP information, which makes it more robust to the adverse effects of indoor environments. Typically, the average system performance of fingerprinting-based systems (within 2m if well trained) can satisfy most of indoor location-aware services. However, the main challenge of Wi-Fi RSS fingerprinting is the very time-consuming system training and frequent database updates that are required to deal with changing conditions within the positioning environment. Despite such disadvantages, the Wi-Fi-based positioning technology has a very promising application prospect mainly because of the ubiquitous and inexpensive nature of Wi-Fi infrastructure. In the next section, these two main Wi-Fi based indoor positioning approaches will be detailed.

2.2 Indoor Positioning Using Wi-Fi Infrastructure

2.2.1 Propagation Models

In free space, the RSS is inversely proportional to the square of the distance between transmitter and receiver. Such a relationship can be captured by theoretic or empirical signal propagation models. The *log-normal shadowing model* is one of the commonly used theoretic models in link budget analysis [33]. The basic idea of this model to Wi-Fi-based indoor positioning can be revealed in Figure 2.2. Suppose $\{p_1, p_2, p_3, \ldots, p_n\}$ is a time series of received power measurements collected by a mobile device about an AP, and P_r is the average of these values, which is assumed to be the outcome of a random variable modelled as normal distribution with mean value $\overline{P_r}$ and variance σ_{P_r}:

$$P_r \sim \mathcal{N}(\overline{P_r}, \sigma_{P_r})$$

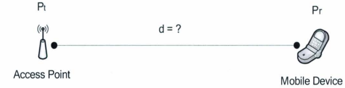


Figure 2.2: Distance between an AP and a mobile device can be estimated using RSS

The distance from an AP with transmitter power P_t (measured at reference distance d_0) can be estimated via the equation:

$$\overline{P_r} = P_t - 10n_p \log_{10}(d/d_0), \tag{2.1}$$

where n_p is the path loss exponent. The standard deviation σ_{P_r} defines the variability measured between pairs of nodes with the same separation distance, but placed at different locations and at different times. Based on the above signal propagation model, the distance between a transmitter and a receiver can be estimated. With at least three transmitters within range, the position of a receiver can be calculated using triangulation as discussed in Section 2.2.1 shown in Figure 2.1.

However, in real indoor environments, it is very difficult to determine proper parameters (e.g., n_p and P_t) for propagation models due to the diffraction, scattering, shading, and multipath phenomena [33]. In order to overcome these obstacles and make the Wi-Fi RSS triangulation methods applicable to indoor environments, researchers commonly choose to treat the specific signal propagation model as a black box [9, 43]. Therefore, a large number of RSS measurements need to be collected in real indoor environments in order to train a propagation model and identify the parameters by optimally solving the simultaneous equations system with respect to Equation 2.1. Specifically, the goal is to find a solution that minimizes the least mean absolute error [9]:

$$J = \frac{1}{N} \sum_{i \in N} \left| P_{m_i} - P_i + 10 n_{p_i} \log \frac{d_i}{d_0} \right|, \tag{2.2}$$

where P_{m_i} is the actual RSS measurement for the i-th AP, $i \in \{1, 2, \ldots, N\}$. However, J is a non-linear objective function for which it could be very computational costly if the number of APs and mobile users are large (generate large number of non-linear equations). An alternative approach is polynomial regression, the ideal nth-degree polynomial regression can be given as [9, 43]:

$$\hat{D}_i = e_0 + e_1 p_i^1 + e_2 p_i^2 + \cdots + e_n p_i^n, \tag{2.3}$$

where e_j, ($j \in \{0, 1, 2, \ldots, n\}$) are the coefficients of the polynomial, p_i is the received signal strength and D_i is the estimated distance from the mobile device to the i-th AP. The coefficient e_j can be easily solved by least squares approximation. Evaluation results in [43] and [41] show that the regression-based method has better performance than the log-normal shadowing model approach. However, these model-based approaches still require substantial training effort in terms of placing infrastructure such as Wi-Fi sniffers, obtaining information on the floor plans, and acquiring knowledge of the locations of AP and their transmission power characteristics. In addition, the system accuracy of model-based approaches is lower compared to the RSS fingerprinting method with less system training efforts [2, 44, 9].

2.2.2 Wi-Fi RSS Fingerprinting

In comparison to the propagation-model based techniques, Wi-Fi RSS fingerprinting is more robust and accurate, and thus has emerged as a very promising solution. It typically contains two phases: 1) training phase and 2) positioning phase. During the training phase, a fingerprint database is constructed to resolve future positioning queries. In the positioning phase, the position likelihood is calculated based on the current Wi-Fi RSS measurement. The general idea of the fingerprint-based approach is given as follows.

Suppose at a survey position $\mathcal{P}_a$, a mobile device can receive beacon frames from the i-th AP, $i \in \{1, 2, 3, \ldots, N\}$. The beacon frame is one type of management frame. The 802.11 standard defines various frame types that stations use for communications,

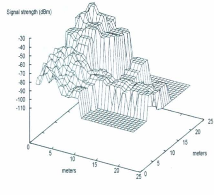
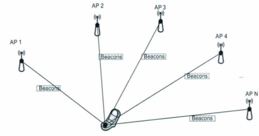


Figure 2.3: RSS readings from an AP at various survey points [5].

as well as managing and controlling the wireless link [11]. The beacon frames contain all the information about the network. They are transmitted periodically to announce the presence of a wireless network. The MAC address M_i, timestamp t_i can be extracted from each beacon frame as features. Also, the RSS p_i, which can be estimated by the receiver by analyzing the beacon frame. An AP can be characterized by these features. One interesting aspect about the characteristics of Wi-Fi RSS can be exemplified in Figure 2.3, where the RSS from an AP collected at various survey points are discriminated due to the signal attenuation.

However, such attenuation can not be efficiently modeled for indoor environments as mentioned before. Furthermore, if multiple APs are visible at the same location, the combination of such RSS features of these APs can "fingerprint" this location. The collection of beacons in a single scan by the device form a Wi-Fi RSS vector R_a. It is a 3-tuple vector, where each element contains the description of an AP, i.e., MAC address, RSS, and timestamp (Figure 2.4). If the location of the mobile device is shifted (e.g., to $\mathcal{P}_b$), we can obtain another Wi-Fi RSS vector R_b. In fact, R_a and R_b can be distinct if $\mathcal{P}_a$ and $\mathcal{P}_b$ are far apart enough. Thus, the Wi-Fi RSS vector can be used as the location "fingerprint".

However, the Wi-Fi RSS vector only reflects the instantaneous features of the Wi-Fi environment. The Wi-Fi RSS in fact can fluctuate drastically in real indoor environments. For example, Figure 2.5 shows the RSS of an AP at the same location but at different times (i.e., crowded lunch time (12:00) and at night (22:00) with few people) at the University Centre of Memorial University. The RSS fluctuation is small during night, but it is large during lunch time because many people are around. Thus, due to the variability of Wi-Fi RSS, multiple scans are needed to constitute



MAC address:	M_1	M_2	M_3	M_4		M_N
RSS:	p_1	p_2	p_3	p_4	...	p_5
Timestamp:	t_1	t_2	t_3	t_4		t_N

Wi-Fi RSS Vector

Figure 2.4: Wi-Fi RSS vector

a fingerprint which contains the summarized features of the scans (e.g., Wi-Fi RSS mean and variance) at a given survey position. The data structure of Wi-Fi RSS fingerprints is detailed in Chapter 3.

As such, a fingerprint database is created by associating each survey position with a Wi-Fi RSS fingerprint. Such a database will be used for future queries in the positioning phase. The positioning system then compares this live RSS measurement to all the fingerprints stored in the database, and returns best matching RSS fingerprints.

Although the basic idea of Wi-Fi fingerprinting is straightforward, many challenges prevent this technology from broad adoption for position estimation beyond academia.

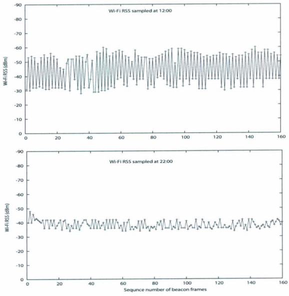


Figure 2.5: Wi-Fi RSS fluctuation over time

Kushki et al. [23] provide four primary such challenges for Wi-Fi fingerprinting-based approaches:

- preprocessing fingerprints to increase accuracy and to avoid collecting data from an excessively large number of positions,
- AP selection,
- quantization of distance between the Wi-Fi RSS vectors in the signal space (i.e., location likelihood calculation), and
- building analytical models to evaluate system performance.

In order to obtain high system accuracy, the training process can be very time-consuming and laborious, especially for future updates and maintenance. Thus, streamlining such a training phase is very important for its commercialization. Chai and Yang [7] and Lemelson et al. [26] argue that users will stand somewhere in between several survey positions in most cases. Therefore, the fingerprints of adjacent positions around the users will also yield suitable matches to the Wi-Fi RSS measurement. These similar fingerprints can be generated via a single seed fingerprint by assigning different weights. As a result, such pre-processing fingerprints can significantly reduce the system training costs. In the extreme case of reducing system training efforts, "zero-configuration" can be achieved by only involving user updates without system training [4].

Commercial Wi-Fi infrastructure is usually deployed with a large number of relatively dense APs. It may seem that a higher positioning accuracy can always be achieved if more access points are utilized. However, this is not the case as indicated by Kaemarungsi and Krishnamurthy [19]. Instead, a subset of access points can be used for the same level of system performance with a much reduced overhead. A straightforward AP selection approach would be to select the subset of APs with the highest observed RSS. More intelligently, Chen et al. [8] provide a novel selection strategy based on the discriminant power of each AP using an information gain criterion. As a result, the APs that best differentiate the survey positions are selected for positioning services.

Much research has focused on the calculation of the difference between RSS observation and fingerprints stored in the database, which is the essence of fingerprint-based techniques. Euclidean distance is a simple but effective way to represent such a difference [2, 20]. The position estimation is either the survey point whose fingerprint has the smallest distance to the observation (nearest neighbor (NN) classification) or the average of k closest survey points (k-nearest neighbor (KNN) [2] classification). Kaemarungsi and Krishnamurthy [20] indicate that fingerprints can be grouped together as a set of clusters. More than one cluster may represent one location because of the multimodal distribution of the RSS. In such a case, using Euclidean distance to determine the location may classify some patterns to a wrong location.

Another group of Wi-Fi positioning methods rely on probabilistic techniques such as Bayesian Networks or Gaussian kernel to handle uncertainty in RSS measurements [15, 34, 22, 44]. Positions are estimated using likelihood or posterior density functions. Kushki et al. [23] propose a comprehensive Kernel-based system framework and integrated elements such as spatial filtering, selection of APs, and spatial feature selection to improve the system performance. In addition to stationary estimate positions, Lee et al. [25] aim to track moving entities. In an indoor setting, the user's mobility is restricted by the environment; the users in fact move along a limited set of typical trajectories. The current set of RSS values for reachable nodes and a number of past samples are used to generate trajectories in the signal space. Such trajectories can be matched to positions on a map.

While extensive research has been performed in absolute position information, there have been fewer attempts in recognizing logical position information, such as room numbers or signboards [6, 4]. However, Wi-Fi RSS fingerprinting-based logical positioning usually lacks the accuracy to discriminate adjacent contexts like neighboring rooms. Martin et al. [29] argue that numerous local attributes already exist in the environment, which may be sensed using cameras, microphones, or accelerometers. By incorporating all these unique environment attributes within the Wi-Fi infrastructure, the system obtains the capability to identify specific logical position information.

Analytical models for analyzing fingerprint based positioning systems have been discussed in the literature [19, 38]. Kaemarungsi and Krishnamurthy [19] analyze the impact of important system parameters and radio propagation characteristics on the system performance, such as the number of APs, the grid spacing (the number of reference locations), path loss exponent, and standard deviation of RSS.

2.3 Integrating Human-Centric Collaborative Feedback into Indoor Positioning Systems

The accuracy of Wi-Fi fingerprinting thus designed is highly dependent on the number of survey positions employed during the training phase. This implies not only a high system overhead and training cost but also vulnerability to environmental changes. Indeed, maintaining such a system would require re-training the system almost from scratch on a frequent basis. On the other hand, if the system can be augmented with learning or compensation capabilities, it will be able to update its own knowledge. Since these systems may provide services to many mobile users, such a learning capability can be obtained via user feedback for free during the positioning phase.

Active Campus [3] is an early system integrating user feedback. It allows users to update the training data incrementally for future use. When the system location is incorrect, users can click on the correct location and suggest new positions. The system then takes the corrected location and MAC addresses and RSS of the currently visible APs to construct a *virtual anchor point* (VAP). Future location computations can then take advantage of these user-created VAP. Similarly, Redpin [4] uses a "folksonomy"-like approach, where many users train the system while using it.

Gallagher at el. [10] focus on the adaptation of Wi-Fi infrastructure alteration. They investigate a new method to utilize user feedback as a way of monitoring changes in the wireless environment. In real indoor environments, some APs may be added to or removed from the infrastructure. Also, due to large-scale signal fading, a mobile device may not hear certain APs in some scans. In order to solve this problem, they assign each AP with certain number of "credits points". Users are prompted to send their RSS measurement to a remote positioning server. The server then looks into the fingerprints available at this location, and compares the APs already present to the ones present in users RSS observation. If an AP is already present in the database but not in the incoming user measurement, its number of points is decremented in the fingerprint recorded at this location. When the number of "credit points" of this particular AP is reduced to 0, it is removed from the fingerprints in the database. Similarly, when an AP is present in the incoming scan result but not in the database, it is added into the database. When several users start to report this new AP, its number of points will increase each time it is reported.

Park et al. [12] propose a user promotion mechanism. In fact, there is always a trade-off between providing imprecise estimates due to the lack of fingerprint coverage, and asking users for too many suggestions, especially when the fingerprint database is only partially trained. They argue that in a human-centric positioning system, it is useful to only prompt users for their location when the system error is large. They propose a mechanism to convey the system's spatial confidence in its prediction based on a *Voronoi Diagram*, and the system only prompts users whenever system confidence falls below a threshold. The Voronoi Diagram denoted as V shown in Figure 2.6 can be described as:

- It is a set of n anchor points in the plane.
- It is the subdivision of the plane into n cells, one cell for each anchor point.
- If a point q lies in the cell corresponding to a site $v_i \in V$, then

$$D(q, v_i) < D(q, v_j),$$

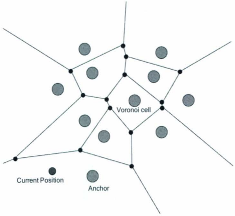


Figure 2.6: User's true position is represented by the anchor point of his/her current Voronoi cell.

for each $v_i \in V, j \neq i$, where D represents Euclidean distance.

The underlying intuition is that a user's current position will be represented by the anchor point of his/her current cell because they have the smallest Euclidean distance among all anchor points. Therefore, the size of the Voronoi cell naturally represents the spatial uncertainty associated with prediction of the bound space. Once the size of the current Voronoi cell is beyond a threshold, the system will prompt users to provide feedback. If a new survey point is associated with a RSS fingerprint generated by users, it becomes an anchor point and adds nearby spaces to the newly-formed cell. Then, the Voronoi Diagram will be updated.

The above approaches refine the existing Wi-Fi RSS fingerprints based positioning system with the integration of human-centric feedback. However, a potential pitfall is that the model constructed during the training phase could also be negatively affected by unreliable or misleading user feedback. Thus, it is crucial that the feedback from users should be given proper weights or credibility, rather than blind acceptance or rejection. The "credits point" assigned to each AP in [10] is a simple but good attempt of such a credibility assessment mechanism for user feedback

Hossain et al. [18] propose a simple credibility rating. That is, when the user does not believe in the position returned by the system, an alternative position can be suggested. In their system, positive user feedback is given a higher credibility weight if the suggested position has a small discrepancy with the system. For example, suppose the precision of a system is 95% within 5m, which means the positioning error is within 5m in most cases. Thus, if the user's estimate position is within the range of 5m of system result, it will be assigned a very high weight. However, according to the observation of our preliminary experiments, the system results are mostly close to user's true position, i.e., within 5m. However, they are occasionally very far away from the true position due to insufficient Wi-Fi RSS data or large variance. In that case, if user's feedback follows the system's estimation and is assigned a high weight, it in fact becomes an outlier feedback and could bring large interference to future positioning queries. Such negative effects from outlier user feedback should be eliminated. A straightforward solution is using clustering algorithms to filter outliers [12].

Later in this thesis work, we will devise a more general framework using a wider variety of user feedback. Such a framework is endowed with a high degree of system robustness when a large number of users provide correct feedback. Even when incorrect feedback is provided, the system is able to quickly recover by incorporating subsequent corrective feedback.

Chapter 3

Position Estimation Baseline

We start by introducing our baseline Wi-Fi fingerprint-based approach. The general idea of the baseline approach is similar in many respects to the systems reviewed in Chapter 2. However, we also refine existing fingerprinting based approaches to make them more robust and suitable for integrating and processing user feedback.

3.1 Training Phase

In the training phase, a set of grid points in the study area are selected as survey positions with known physical coordinates. The system training is conducted for each survey point in a two-step process. The first step is to collect multiple Wi-Fi RSS vectors in order to stabilize the average of RSS readings and calculate the variances. The variance is used to detect the environment interference level, where a large variance tends to cause unreliable positioning results. The following step is utilizing these RSS vectors to generate an RSS fingerprint for each survey position. By combining the positioning information and RSS fingerprint, anchor points are set

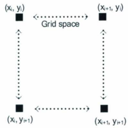


Figure 3.1: Grid space

as reference points for the positioning of mobile devices.

3.1.1 Collect Raw Wi-Fi Measurement Data

At the first stage of system training, every survey position is pre-placed on a map with known physical coordinate (x, y). The grid space between two survey positions determines the resolution or granularity of the positioning system (Figure 3.1).

A smaller grid spacing may increase the granularity or accuracy, but not the precision or the probability of correctly matching the survey position because the Wi-Fi RSS fingerprint of two survey positions may be very similar. Also, smaller grid spacing causes laborious system training and maintenance. In fact, there is no general guideline to choose the optimal grid space. In the implementation of our baseline system, the grid space is 3m (i.e., the distance between two grid cells), which is a reasonable choice considering both the size of our study area and the accuracy and performance for regular indoor positioning service.

At each survey position, system administrators use a mobile device to scan for

Wi-Fi RSS in dBm

MAC address of an AP

```
0000  00 00 14 00 ee 18 00 00  10 02 9e 09 a0 00 be 9c  ........ ........
0010  64 00 00 22 80 00 00 00  ff ff ff ff ff ff 00 22  d..".... ......."
0020  55 e0 29 d5 00 22 55 e0  29 d5 90 23 78 f1 32 ff  U.).."U. )..#x.2.
0030  76 00 00 00 66 00 31 04  00 01 00 01 08 82 84 8b  v...f.1. ........
0040  0c 12 96 18 24 03 01 0b  05 04 00 01 00 00 07 06  ....Y... ........
0050  43 41 20 01 0b 1a 0b 05  02 00 70 8d 5b 2a 01 00  CA ..... ..p.[*..
0060  30 14 01 00 00 0f ac 02  01 00 00 0f ac 04 01 00  0....... ........
0070  00 0f ac 01 28 00 32 04  30 48 60 6c 85 1e 0e 00  ....(.2. 0H`l....
0080  8f 00 0f 00 ff 03 59 00  75 63 31 31 31 77 31 00  ......Y. uc111w1.
0090  00 00 00 00 00 00 00 00  02 00 00 32 96 06 00 40  ........ ...2...@
00a0  96 00 14 00 dd 18 00 50  f2 01 01 00 00 50 f2 02  .......P .....P..
00b0  01 00 00 50 f2 02 01 00  00 50 f2 01 00 00 dd 06  ...P.... .P......
00c0  00 40 96 01 01 04 dd 05  00 40 96 03 05 dd 05 00  .@...... .@......
00d0  40 96 0b 09 dd 08 00 40  96 13 01 00 34 01 dd 05  @......@ ....4...
00e0  00 40 96 14 05 dd 18 00  50 f2 02 01 01 80 00 03  .@...... P.......
00f0  a4 00 00 27 a4 00 00 42  43 5e 00 62 32 2f 00 4d  ...'...B C^.b2/.M
0100  9a 45 c5                                          .E.
```

Figure 3.2: A beacon frame is captured and analyzed by Wireshark. RSS is generated based on the received beacon frame.

the beacon frames transmitted by nearby Wi-Fi APs. The beacon frame provides the "heartbeat" of an AP, enabling communications to be conducted in an orderly fashion. The destination MAC address of a beacon frame is always set to FF:FF:FF:FF:FF:FF, which forces all others on the same channel to receive and process beacon frames [11]. When a beacon is processed, the radio network interface card (NIC) learns a great deal of information about that particular AP (e.g., Wi-Fi RSS and MAC of an AP). Figure 3.2 shows a screenshot from a network analyzer called Wireshark [1]. It displays a captured beacon frame in hexadecimal notation. (RSS is -34 dBm in decimal and the MAC address is 00:22:55:E0:29:D5).

In each Wi-Fi scan, beacon frames from different APs are received and converted

MAC	RSS Mean	RSS Variance	Timestamp	Count
MAC	RSS Mean	RSS Variance	Timestamp	Count
MAC	RSS Mean	RSS Variance	Timestamp	Count
		...		
MAC	RSS Mean	RSS Variance	Timestamp	Count

Figure 3.3: Data format of a Wi-Fi RSS fingerprint.

to a list of 3-tuples (i.e., an RSS vector, as shown in Figure 2.4), where a 3-tuple element contains the MAC address of an AP, the RSS in dBm and timestamp. Note that a single scan may not be able to capture beacon frames from all nearby APs due to the different beacon frame broadcasting periods or severe signal fading. Also, as mentioned in Chapter 2, the collected RSS values have a natural variation when indoors, which is unavoidable. To compensate the RSS fluctuation and obtain complete AP information, a sufficiently large number of scans is needed to create an RSS fingerprint. As a result, in a given period of sampling, the device logs a time series of RSS vectors. Such vectors will then be used to construct the Wi-Fi RSS fingerprints for each measured location in the training grid.

3.1.2 Generate Wi-Fi RSS Fingerprint

The statistics are extracted from the raw Wi-Fi measurement data to generate an RSS fingerprint for each survey position. A Wi-Fi RSS *fingerprint* is defined as a vector of 5-tuples (i.e., *MAC*, *Timestamp*, *RSS Mean*, *Count*, and *RSS Variance*), describing a set of APs, as shown in Figure 3.3. The definitions and explanations for each field are given as follows.

Given the i-th AP in a Wi-Fi RSS fingerprint:

- *MAC*: The MAC field contains its MAC address, denoted as M_i. It is a unique identifier for each wireless network interface card. We use that to distinguish among the different Wi-Fi APs that are within range.

- *Timestamp*: The time of creating the fingerprint is stored in the Timestamp field, denoted as t. In indoor environments, time-dependant human activities could affect positioning activities because human bodies can absorb Wi-Fi signals. The timestamp of fingerprint could be utilized to provide time-aware positioning.

- *RSS Mean*: The RSS Mean $\overline{p}_i$ is an average of the Wi-Fi RSS over the sampling period. During the sampling period, several Wi-Fi RSS vector will be generated. Each Wi-Fi RSS vectors contains the instantaneous RSS values. Since the RSS values are normally fluctuating, it is beneficial to smooth them. At this point, we choose to average the RSS readings.

- *Count*: The value of Count is the number of occurrences of the AP during the sampling period, denoted C_i, which is a very important indicator for the reliability of this AP. For a fixed number of Wi-Fi scans, a large Count value means that the AP can be heard for most of the time, indicating that the AP will have a more reliable estimation of its RSS value.

- *RSS Variance*: RSS Variance contains the variance of the measured RSS from the AP, denoted σ_i. The fluctuation level of the current Wi-Fi environment at a certain survey position can be estimated by analyzing the Wi-Fi RSS fingerprint. Typically, the RSS fingerprint contains multiple APs. Each AP has its own mean and variance, which can not provide a global description about the current Wi-Fi environment. In order to estimate the fluctuation level of the entire environment, we use the weighted average of RSS Variance for each AP. The occurrence or the value in the Count field for each AP is utilized as the weight. The collective RSS variance for this fingerprint is defined as

$$\sigma_{F_s} = \frac{\sum_{i \in F_s} \sigma_i C_i}{\sum_{i \in F_s} C_i},$$

where F_s is its RSS fingerprint.

At the end of the training phase, each survey position is associated with an RSS fingerprint containing APs that describe the specific location. For each survey position $\mathcal{P}_s$ in the system, we define a *system anchor* A_s as

$$(\mathcal{P}_s, F_s),$$

The system anchors are reference points to determine the positions of mobile devices.

3.2 Positioning Phase

In the positioning phase, live Wi-Fi measurements will be collected and used to query the fingerprint database. Using only a few Wi-Fi scans during positioning phase may generate a large error due to the lack of informative RSS data. For experimental purposes, the prototype implementation allows for a variable number of Wi-Fi scans to evaluate system performance.

Suppose the total Wi-Fi scan number is S and each scan will generate an RSS vector R_i, $i \in \{1, 2, 3, ..., S\}$. Given N system anchors, when the first RSS vector is

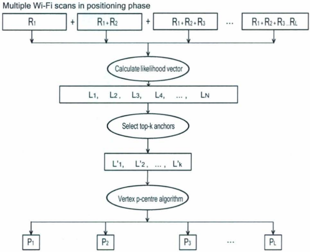


Figure 3.4: Flowchart of positioning phase.

formed, we use it to calculate the likelihood L_j, $j \in \{1, 2, 3, ..., N\}$, of it matching the fingerprint for each system anchor. Each subsequent scan should lead to a cumulative estimation result with a decreasing error. As such, the estimated result will become more and more reliable as more RSS vectors are used.

The likelihood is calculated via a Gaussian kernel, which is commonly used to estimate the likelihood between two RSS vectors [23]. Then the top-k anchors with highest likelihood are selected as candidates of a system return. A representative of them will be selected as returned result using a solution to vertex-p centre problem. The flowchart of the entire positioning process is shown in Figure 3.4, and each step will be explained in the following subsections.

3.2.1 Calculate Likelihood

Besides the large RSS variability, another challenge in real indoor environments is the variability of RSS vector dimensions. In an RSS vector R_i, the MAC address of an AP defines a dimension in the vector. Thus, the number of dimensions of R_i can be given as

$$d_{R_i} = n_i,$$

where n_i is the number of received APs in Wi-Fi scan i. Due to the different beacon frame broadcasting periods, modifications of the Wi-Fi infrastructure (e.g., APs are turned off or new APs are added), or large fading effects, the number of dimensions of RSS vector R_i and R_j, $i \neq j$ generated at the same location l could be different since:

$$d_{R_i} \neq d_{R_j}, \quad i, j \in \{1, 2, 3, \ldots, S\}.$$

As a result, a dimension mismatch between RSS fingerprints and RSS vectors in live measurement could happen during the positioning phase. It also indicates that system estimate results could be very unreliable if the number of Wi-Fi scans is small.

If we use simple likelihood calculation mechanisms (e.g., Euclidean distance or cosine similarity), such dimension mismatching could lead to large positioning errors. However, if the influence of each dimension can be normalized, the small scale of dimension mismatching will not dominate the entire likelihood calculation. In terms of our baseline system, we use sparse vectors and a Gaussian kernel to calculate the likelihood for each system anchor, which is very efficient according to our preliminary experiments. Specifically, an RSS fingerprint is first transformed into a sparse vector which contains all n APs. At this point, we do not consider the AP selection problem; instead, all nearby APs in the infrastructure are utilized in order to obtain satisfactory system performance. As such, the dimensions of all RSS vectors are unified, a dimension without valid AP information (i.e., can not receive beacon frame from certain AP) will be assigned impossible values (e.g., -100 dBm for RSS). Then, we apply Gaussian kernel to calculate likelihood between two sparse vectors.

Gaussian kernel method was originally used in support vector machine (SVM) to classify data [36], and it has also been found to be very efficient for RSS vectors likelihood calculation [23, 41, 40, 19, 15]. In the Gaussian kernel method, a probability mass is assigned to a "kernel" around the RSS mean of each AP in fingerprint generated in the training phase. Given an RSS live measurement (observation) vector generated at location $\mathcal{P}$ as $R_{\mathcal{P}_s}$, the resulting likelihood estimate between $R_{\mathcal{P}_s}$ and fingerprint F_i in system anchor A_{s_i} is the sum of n equally weighted density functions

$$L(R_{\mathcal{P}_s}, F_i) = \sum_{k=1}^{n} K_G(p_{M_k}; p_{F_k}),$$

where p_{M_k} is the RSS of k-th AP in the live measurement vector $R_{\mathcal{P}_s}$ and p_{F_k} is the RSS Mean of k-th AP with the same MAC address in fingerprint F_i. Note that when p_{M_k} or p_{F_k} is an impossible value (e.g., -100 dBm), we just ignore this dimension. K_G denotes the Gaussian kernel or radial basis function (*Gaussian RBF*), whose value depends on the distance from the centre. It is given as

$$K_G(p_{M_k}; p_{F_k}) = \frac{1}{\sqrt{2\pi}\delta} exp\left(-\frac{(p_{M_k} - p_{F_k})^2}{2\delta^2}\right),$$

where δ is an adjustable parameter that determines the width of the Gaussian Kernel and the centre is p_{F_k}. Figure 3.5 provides four Gaussian RBF curves (with the same centre (-40 dBm) but different δ) to illustrate the characteristics of Gaussian RBF. From these curves, we can observe that the Gaussian RBF has two main features. The first one is the discrimination ability for RSS values on the same dimension. Any RSS close to the centre has a large Gaussian RBF value, as we can see in Figure 3.5. Thus, in terms of a RSS vector with n dimensions, the sum of Gaussian RBF over all n dimensions determines their likelihood. The second characteristic is that its width is determined by parameter δ. As we can see in Figure 3.5, Gaussian RBF is smooth when the δ is large. In terms of Wi-Fi RSS, whose value domain is $[-90dBm, -30dBm]$, δ less than 0.05 or larger than 0.5 lead to corresponding curves too sharp or flat, which could cause weak discrimination ability of Gaussian RBF. However, to find the optimal δ value for a particular dataset is difficult, especially for Wi-Fi RSS data with large variability in indoor environments [23]. In the partic-

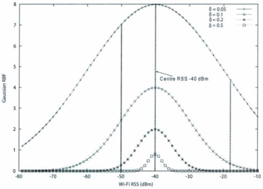


Figure 3.5: Examples of Gaussian kernel.

ular environment, we have to tune the δ value in order to archive adequate system performance.

After the likelihood calculation, each system anchor has a likelihood for being the true position of the device. Instead of just returning a single estimation, the system selects the top-k system anchors as candidates in order to provide redundant true position information. The main reason is that the true position may not always be in the system anchor with the highest likelihood. The next step is to choose a representative from these top-k candidates as the system return.

3.2.2 Present Position

A naïve approach would be to use the weighted mean of the top-k anchors as the estimation for the position. Usually, these k survey points are close to each other

Figure 3.6: A drawback of weighted mean representation.

in the physical space, and they can be considered as a cluster. Thus, their weighted mean position is a reasonable representative. One example is shown in Case 1 of Figure 3.6; four system anchor points ($k = 4$ here) are close to each other and can be considered in the same cluster. Thus, their weighted mean position can be used as a applicable representative. However, if one or more outliers exist, the weighted mean position could be pulled far away from the cluster formed by other system anchors. Also, this mean position can be a meaningless point in the physical space. Case 2 in Figure 3.6 provides such an example. Anchor point A_{S_1} is far away from the other system anchor points, which could pull the weighted mean position away from the cluster formed by A_{S_2}, A_{S_3}, and A_{S_4}. If these four system anchor points are very close in likelihood, the centre of A_{S_2}, A_{S_3}, and A_{S_4} should be a more representative than the weighted mean position.

Instead, we can use an approach to the vertex p-centres problem [21] to determine

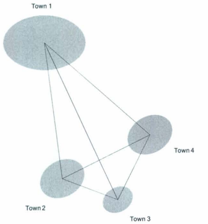


Figure 3.7: Example of fire station placement problem

the representative of the top-k anchors. The vertex p-center problem (also known as the minimax problem) is to locate p facilities (vertexs) and assign clients to them so as to minimize the maximum distance between a client and the facility to which it is assigned. It is a computationally expensive problem for general p. However, in our case, we only consider the case of $p = 1$, i.e., the 1-centre problem. Since the value of k could be very small (less than five), we do not analyze the algorithm complexity at this point.

The 1-centre problem is similar to the site selection problem depicted in Figure 3.7. Given four towns with different populations, we need to place a fire station in one of these towns to cover the entire area. The large town should be given more weight because the possibility of a fire is proportionally higher than the smaller towns. Also, if a town is very closer to other towns, firemen can reach the accident scene quickly. Thus, the town with large population and small distance to other towns should be chosen as the location for fire station. In particular, the vertex 1-centre for our positioning system is the system anchor point that minimizes the maximum distances from itself to the other top-$(k-1)$ anchor points. These distances are weighted with the likelihood estimated as above. For two indices $i, j = 1, 2, \ldots, k$, we minimize the following over all values for i

$$\max_{j \neq i} \frac{D(i,j)}{L_i},$$

where $D(i,j)$ is the Euclidean distance between anchor A_{S_i} and A_{S_j} and L_i is the likelihood of A_{S_i}. The resulting anchor point becomes the estimation for the location.

3.2.3 The Algorithm of Baseline System

In this subsection, we summarize the baseline system with two algorithms given as follows (next page). Given N survey positions with known physical coordinates, our goal in the training phase is to associate each of them with an RSS fingerprint $F_i, i \in \{1, 2, 3, \ldots, N\}$. A RSS fingerprint is generated by calculating the mean RSS from S Wi-Fi scans. As such, N system anchors are created as reference points for future positioning.

3.2.3.1 Training Phase

Algorithm:

Input: Given a temporary vector $V = \{v_1, v_2, \dots, v_n\}$, v_i is the sum of RSS collected on dimension i (initialed to 0) of all T scans. n is the number of all nearby APs;

while *($i \leq N$) for each survey position* **do**

  while *($j \leq T$) for each scan* **do**

    $\{p_{j_1}, p_{j_2}, p_{j_3}, \dots, p_{j_n}\}$, p_{j_k} is the RSS vector R_j in scan j;

    while *($k \leq n$)* **do**

      if *beacon frame from k-th AP is received* **then**

        Add p_{j_k} to v_k ;

        AP count $c_k + 1$;

        $\bar{p_{j_k}} = \frac{v_k}{c_k}$;

        Set $\bar{p_{j_k}}$ to the RSS of k-th dimension in F_i ;

      end

      else

        set -100 dBm to p_{j_k};

      end

    end

  end

  $A_{s_i} = (P_{s_i}, F_{s_i})$;

end

Output: A_s (All system anchors);

3.2.3.2 Positioning Phase

Algorithm:

Input: Also given a temporary vector $V = \{v_1, v_2, \ldots, v_n\}$ (generated when users want to find their positions), v_i is the sum of RSS collected on dimension i (initialed to 0) of all T' scans. n is the number of all nearby APs;

while *($i \leq T'$) for each scan in positioning phase* **do**

  while *($k \leq n$) for each dimension* **do**

    p_{i_k} is the RSS in RSS vector R_i, scan i ;

    if *beacon frame from k th AP is received* **then**

      AP count $c_k + 1$;

      $v_{i_k} = \frac{v_k + p_{i_k}}{c_k}$;

    end

    else

      set -100 dBm to p_{j_k};

    end

  end

  while *($j \leq N$) for each system anchor* **do**

    Calculate likelihood L_j between V_i and F_j using Gaussian Kernel;

  end

  Select top-k system anchors from L ;

  Select returned position by solving 1-centre problem;

end

Output: Estimation position in the 1-centre system anchor point.

In the positioning phase, likelihood is calculated for each system anchor when receiving an RSS vector. The top-k system anchors are selected as candidates for the position estimation. We select the representative of these candidates via solving the 1-centre problem. The whole process repeats S' times, which means S' RSS vectors are used for positioning in total. Each subsequent RSS vector is integrated with previous vectors to produce cumulative estimation result with a decreasing error.

This baseline Wi-Fi fingerprinting approach is similar in many aspects to systems discussed in Chapter 2 and the system performance is very promising if well trained. However, in order to improve system performance, we use the sparse vector and a Gaussian kernel to calculate likelihood for each anchor point. In addition, an approach to the 1-centre problem is employed to select the representative from the top-k anchors, which improves the system robustness to outlier anchor points. The evaluation of this baseline approach will be detailed in Chapter 6.

At this point, although the computational complexity is $O(n^2)$, it can be easily optimized by using pre-fingerprint clustering and tree-based search methods [8]. However, the refinement of algorithm is not our research focus. In fact, since the metre-level accuracy can be obtained via extensively system training [44], we believe that the most challenging issues are system robustness and costs. As mentioned above, although a fingerprinting-based approach is relatively more robust and accurate than a triangulation-based approach, its system performance is still highly dependent on the large amount of training data and the RSS variability (the interference in the physical environments). Thus, the goal of this research is to give the positioning system a self-learning ability to adapt to environment changes and reduce re-training or maintenance costs. We argue that such ability could come from end users if the system is enhanced with an user feedback model to efficiently receive and process human-centric collaborative feedback. In the next chapter, we will discuss the proposed user feedback model.

Chapter 4

User Feedback Model

RSS vectors in live measurement may occasionally match the system anchor points far from the true position due to large RSS variance, insufficient sampling time, or other factors. However, if this system is enhanced with a self-learning ability adapting it to the environmental changes, such inaccurate positioning outcomes can be compensated. This learning ability may include two components, absorbing new knowledge and abandoning outdated or incorrect knowledge. It could receive inputs from other channels (e.g., motion or vision information) to adjust the likelihood of anchor points, filter outliers, or even create new anchor points that best describing the current Wi-Fi indoor environments. As such, the likelihood distribution could be adjusted by reducing the likelihood of some invalid anchor points or increasing the likelihood of certain efficient anchor points (Figure 4.1).

For mobile devices carried by people, such self-learning ability and positioning compensation could come from end users for free. Users can provide feedback to the positioning service based on their knowledge of the surroundings. They may choose

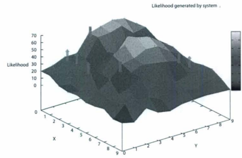


Figure 4.1: General idea of positioning compensation, green arrows mean the likelihood of anchor points at those positions are raised while the red arrows indicates that they are reduced.

to accept, reject, or modify system results after being given the estimated position. In order to utilize user feedback, we need an efficient user feedback model and to study such a model to determine if it is able to improve system performance.

Before discussing the user feedback model in detail, it is useful to begin by identifying three types of user input that can be collected within a human-centric collaborative feedback system:

- *Positive feedback* is generated when users reject the estimated position and suggest a location based on their knowledge. In such a case, the system can accept the updated information from the users. The result is that the system may create new anchors from the users' suggestions, called *user anchors*.
- *Negative feedback* indicates that the users do not believe the estimated position,

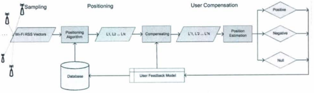


Figure 4.2: Flowchart of user feedback model

and are unable to make any suggestion as to their current location. In this case, the system should reduce the positioning likelihood of the returned location in the future.

- *Null feedback* occurs when users choose not to provide any feedback. The assumption here is that the estimated position is accurate, and that there is no need to make any modification to the positioning model.

Next, we will present the general idea of our positioning model integrating user feedback. The flowchart of this model is provided in Figure 4.2. Assume that the model has n (system and user) anchors, and the likelihood of the i-th ($i = 1, 2, \ldots, n$) anchor is denoted as L_i. Before ranking these anchors based on the likelihood vector L, our user feedback model compensates each L_i with two factors, α_i and β_i as

$$L' = \begin{cases} \beta_i L_i & \text{if } A_i \text{ is a system anchor}, \textit{and} \\ \alpha_i \beta_i L_i & \text{if } A_i \text{ is a user anchor.} \end{cases}$$

Due to the temporal or permanent random interfering factors of complex indoor environments, the reliability of system anchors will be reducing. In order to solve this problem, we design the β factor to gradually reduce the likelihood of system anchors as negative feedback is received. As mentioned before, the system estimation is provided by the vertex-1 centre of top-k anchors. However, if this estimation receives negative user feedback, this means that the user believes that they are not near this location which is an indication that the data stored for these top-k anchors may not be accurate. As a result, the model reduces their likelihood by updating the β factors for these top-k anchors. If more and more users provide negative feedback on a system anchor, it may never be selected as one of the top-k anchors. The β factor thus gives the system an ability to forget outdated or unreliable knowledge.

On the other side, new knowledge (user anchors) will be added into the database via positive user feedback. However, when a user anchor is firstly created, its likelihood is reduced by the discounting effect of the small initial α value. The rationale is that the system can not assess the reliability or credibility of a newly created user anchor (which may be from a malicious user). However, as more and more similar user anchors are generated to confirm it, its α factor will be increased. Once some user anchors become sufficient reliable, they may appear to be within the top-k anchors to affect the system estimation. Also, the β factor could affect user anchors once they receive negative feedback. The user anchor and α factor enable the system to absorb new knowledge about the Wi-Fi environment.

As such, future users can take advantage of the knowledge shared by previous users. Also, they are encouraged to provide feedback to benefit subsequent users. As a result, the positioning model can be consistently updated via the user feedback model thus designed. Later in this section, we will explain how to calculate the α and β factors in detail.

4.1 Positive User Feedback

The general idea of processing positive user feedback can be explained via the example in Figure 4.3. Suppose likelihood calculation is finished, and each system anchor

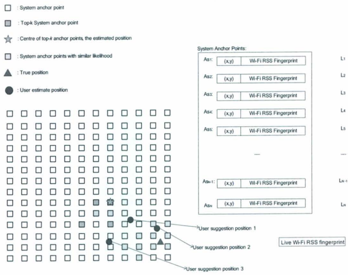


Figure 4.3: Positive feedback

$A_{s_i}, (i \in \{1 \dots N\})$ has a likelihood value L_i. The returned system anchor (green star) is the vertex 1-centre of the top-k anchors. However, it is far away from the user's true position (blue triangle). For positive user feedback, users try to tell the system their estimations by providing suggestion positions. Such estimate positions are shown in Figure 4.3 as red circles. Note that they could be close to the true position (accurate feedback) or still far away from it (inaccurate feedback).

Whenever the system receives a user-suggested location associated with its current RSS measurement, denoted as *user fingerprint*, the system creates a temporary user anchor (A_u). If this anchor is sufficiently similar to an existing user anchor in the model, it is merged with it, and the α factor is updated. Otherwise, it becomes a new user anchor, with the associated α factor set to a very small initial value. It indicates that the newly create user anchor is not as reliable as system anchors at the beginning.

4.1.1 Temporary User Anchor

Since a user's suggested position could be arbitrary, saving these suggestions separately would bloat the model significantly. Therefore, we use discrete locations by dividing the study area into an $m \times n$ grid. Note that the resolution of this grid could be different from the resolution as used in the training phase. We can set smaller grid space because the system training from users is cost-effective. This helps to efficiently reduce the grid space between system anchors. Thus, the resolution of entire system could be refined.

Within each grid cell, its geometric centre is used to represent the positions of all temporary user anchor points falling into it, as in Figure 4.4. We thus define the user anchor A_u as:

$$A_u = (\mathcal{P}_u, F_u),$$

where $\mathcal{P}_u$ is the grid cell centre that contains the user suggested position and F_u is

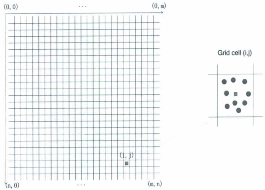


Figure 4.4: The geometric centre of grid cell (i, j) represents all user estimate positions falling into it.

the user fingerprint summarized from the current Wi-Fi RSS measurement.

4.1.2 Anchor Merge

A newly generated positive feedback could be either converted to a new user anchor point or merged with an existing user anchor point based on their similarity. As mentioned before, we believe that positive feedback represented by a user anchor point should gradually become reliable if more and more similar user anchor points are generated to confirm it. Before we discuss how to update the reliability of user anchors, we define the similarity between two user anchor points.

Given user anchor points A_{u_i} and A_{u_j} $i \neq j$, their similarity is determined by two aspects:

- *Wi-Fi RSS fingerprint similarity*: At this point, we do not measure the precise similarity. Instead, we only need a mechanism to reflect the positive coefficient. A natural measurement mechanism is the cosine similarity in the range of [0, 1], which is convenient to compare their fingerprint similarity. Thus, the Wi-Fi RSS fingerprint similarity F_u is given as:

$$s_{F_u} = \begin{cases} 1 & \text{if } \cos\left(F_{u_i}, F_{u_j}\right) > a \\ 0 & \text{otherwise,} \end{cases}$$

  where F_{u_i} and F_{u_j} are Wi-Fi RSS fingerprints of user anchor points A_{u_i} and A_{u_j} respectively. They are all sparse vectors with n dimensions; a is the threshold for Wi-Fi RSS fingerprint similarity.

- *Physical position similarity*: If two user anchor points share the same geometric centre of a grid as their position. They are considered as similar in position.

As a result, we claim that two user anchor points are similar if they satisfy both of the above two similarity conditions. Moreover, the timestamp should also be an important aspect. Different times (morning, noon, and night) or dates (weekdays, weekends, and holidays) could produce different RSS patterns. For example, in a university cafeteria, due to the interference from human bodies and electronic devices, the user fingerprints generated during dinner time could be very different from midnight. As such, user feedback should have a time-bound, wherein it is only able to affect other users within similar time period (e.g., in the same time sliding window). At this point, however, we do not consider this time factor. In Chapter 7, we will discuss this issue as one aspect of our future work in detail.

A temporary user anchor A_{u_i} is thus merged with the existing user anchor A_{u_j} in the same cell if their fingerprints are sufficiently similar. If multiple anchors already exist in the same cell as A_{u_i}, we only consider the most similar one, denoted A_{u_j}. If the similarity between A_{u_i} and A_{u_j} is greater than a threshold, the temporary user anchor is regarded as the same as the existing one, and therefore is merged with it.

4.1.3 The α Factor

Whenever a temporary user anchor is merged with an existing user anchor in the system, the associated α factor is updated. For user anchor A_{u_i}, we define α_i as

$$\alpha_i = \frac{1}{a + e^{-x}}, \text{ with } x \geq 0 \text{ and } 0 < a \leq 1,$$

where the variable x has a cumulative effect and a is a parameter controlling the initial and maximum values of α_i. When an user anchor A_{u_i} is firstly created, its original likelihood will be reduced by the small α_i. As more positive feedback is provided in support of it, its α factor gradually increases until it reaches an upper limit.

Thus, the magnification capability of the α factor is $\frac{a+1}{a}$. The increment of x is defined as

$$\Delta x = \frac{\frac{T}{T_s} + e^{-\sigma_F}}{b} \text{ with } b > 0.$$

The pace of the increase of x is controlled by a few aspects:

- An independent parameter b, which compensates the increasing velocity of x. When there are many users (e.g., in a large shopping centre), we may not want to trust their individual estimation much. Instead, we can reply on the convergence effects of large amount of users to evolve the mode. However, when there are only a few users (e.g., in a depot), we assign each individual feedback much higher weight.

- The variance of the current RSS fingerprint, σ_F. The user feedback generated in the environment with small RSS variance will have larger influence on the evolution speed of the model.

- If T is the number of Wi-Fi scans used in the positioning query and T_s is the number of Wi-Fi scans used during system training, their ratio $\frac{T}{T_s}$ also reflects the credibility of this positive feedback.

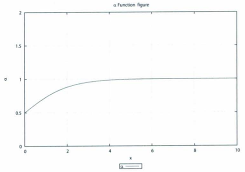


Figure 4.5: The α factor increases fastest at the beginning and becomes stable once a sufficient number of feedback events are received with an upper limit.

As a result, the α factor increases fastest with the first few instances of the user anchor, becoming stable once a sufficient number of feedback events are received, as we can see in Figure 4.5 ($\alpha = 1$). The rationale for this design is to allow the system to quickly adapt to new information provided by the users, but without this feedback overpowering the system.

4.2 Negative User Feedback

Suppose the system delivers a position from top-k anchors according to their likelihood ranking, but the user believes this location to be incorrect and cannot provide any further information regarding the actual location. The negative user feedback on this estimated position can also provide valuable information to the system. Typically, when a user rejects the position estimated by the system, the reason could be that the user is nowhere near any of the anchors known by the system. In this case, none of the top-k anchors would truly represent a good estimate. Therefore, we should try to decrease their likelihoods simultaneously.

Given an anchor A_i, we use a negative user feedback factor β_i to reduce its likelihood according to the accumulation of negative feedback received. Similar to the positive feedback model, the negative factor model also has fast adaptability. Accordingly, we define β_i as

$$\beta_i = e^{-x}.$$

When an anchor is given a negative feedback, we give x in above formula the same increment Δx used in the positive user feedback. The value of β is inversely related to x, such that β will decrease from the initial value 1 to its limit zero as x increase from zero to infinity. The curve of β factor is shown in Figure 4.6. As a result, if more and more users reject the same set of anchors, they will never be chosen as the top-k due to the small value of the β factor.

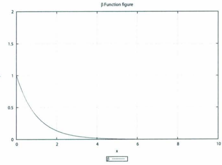


Figure 4.6: Similar to α factor, the β factor also has fast adaptability at the beginning and will decrease from the initial value 1 to its limit zero.

4.3 Null User Feedback

The null user feedback is generated if users choose to accept the location estimation or do not want to provide any feedback. In such a case, the model will not be updated.

4.4 Summary of User Feedback Model

In this chapter, the proposed user feedback model is explained in detail. It processes three type of user inputs (i.e., positive, negative, and null user feedback). Positive feedback generates a new type of anchor point called user anchor. The user anchor will be merged with an existing user anchor resulting in its reweighting, or created as a new anchor which is assigned a small initial weight. Negative feedback reduces the reliability or credibility of anchor points (both system anchors and user anchors).

Reliable user feedback will have more impact on system results. The influence of user feedback depends on three factors: 1) the convergence effect of other similar user feedback 2) the interference level of current environment, and 3) the number of Wi-Fi scans (the effort for conducting a positioning activity). As such, we believe such model should be robust to malicious feedback which normally exists as outliers in real life.

At this point, we have explained the baseline Wi-Fi fingerprinting-based approach in Chapter 3 and the proposed user feedback model in this chapter. These two chapters form the theoretical part of the thesis. In order to validate and evaluate the model in real indoor environments, we have built a prototype on the Apple iOS (which runs on both iPhones and iPod Touch devices). In the next Chapter, the features of this prototype will be introduced.

Chapter 5

System Design

In this chapter, the general design of the system will be presented. We mainly introduce our system architecture and user interface (UI). A detailed class diagram is provided in Appendix A.3.

5.1 Design Goals

There are three main techniques for system performance evaluation 1) analytical modeling, 2) simulation, and 3) measuring a prototype system. Analytical modeling and simulation provide easy ways to predict the performance or compare several alternatives, especially if the prototype is not available or in the design stage. However, they are unable to identify potential flaws in the model which could only appear in real observations. Also, for Wi-Fi based indoor positioning techniques, it is difficult to predict the system performance merely via simulation or analytical modelling. Thus, in order to conduct comprehensive and valuable evaluation, we have built a prototype to enable positioning activities and user feedback input in real indoor environments.

Based on this prototype, we can design field trails to evaluate the performance of the proposed model. Specific prototype design goals are listed as follows:

- Facilitate system training. Since the system training can be very time-consuming and tedious, the prototype should be able to help the administrators to train the system effectively and accurately.
- Reasonable UI design. The UI design is essential for human-centric computing. We thus need a well-designed UI to present the position estimates in terms of a map, along with a method for obtaining both positive and negative user feedback.
- System status monitoring and log file. System analysts should be able to monitor the system and record its running status.
- Statistical experimental results. The system should store all raw run-time data, and it should pre-process these and present the statistics result for analysis.
- Fast system responsiveness. The UI responding delay is a very important system performance metric. The prototype should provide near-instantaneous UI response.

5.2 Architecture

5.2.1 Platform

The operating system of our prototype is Apple iOS 3.1.2, which is an advanced mobile platform. It is streamlined to be compact and efficient, and taking maximum advantage of the iPad, iPhone and iPod Touch hardware. Technologies in iOS such as the OS X kernel, sockets, and OpenGL ES provide comprehensive application programming interface (API) and high compatibility. The iOS SDK combined with Xcode developer tools make it very convenient to debug the code, design the UI, manage the data, and analyze the application run-time performance.

Unfortunately, the Wi-Fi API is not publicly available even for the latest iOS SDK. Instead, we indirectly use iOS system calls via a private Wi-Fi framework called WiFiManager to scan nearby APs.

5.2.2 System Architecture

We will introduce the logical model of our prototype in this section. At this point, we focus on the system architecture, the relationship and interaction between modules. The detailed class information is provided in Appendix A.3.

In terms of system architecture, if we adopted a client/server architecture, the positioning process could be conducted using a positioning server in a centralized manner. Furthermore, a large amount of map data, fingerprint data and user data could be stored in the database at the server side. Therefore, the client running on a mobile device would only need to download the map and send a positioning request to the server and wait for the result. By doing so, the resource consumption on the mobile device could also be reduced. However, the system response time will depend on the communication quality between the client and the positioning server. If the network is congested or the RSS from associated AP is at a extremely low level, users will have to spend a long time waiting for the system results. Also, if users can not access the network for some reason, such a positioning service will be unavailable. Furthermore, in order to protect users' private information (e.g., the history of location queries), the server needs to integrate additional security mechanisms such as data encryption, secure data transmission, or access control, which increase the cost and complexity. By taking consideration of these aspects, we have implemented the positioning process locally (i.e., a lightweight stand-alone version). In such an offline operating mode, the position calculation process will be conducted on the mobile device to protect privacy and reduce the dependence on networks at the same time. Also, if users want to take advantage of collaborative feedback from other users, they can synchronize their local user feed model with a server at a different time. As such, their feedback can be uploaded to the server and benefit other users.

The architecture of our prototype is based on a variety of layers, from *UI* on the application level to *Wi-Fi* and *System Foundation* at the iOS kernel level. Figure 5.1 shows a high-level overview of these layers. Next, we will explain the general functionality of each layer and how they communicate with each other.

The System Foundation layer is designed to provide a fundamental framework for the entire prototype. It contains basic functionalities such as key object initialization, views navigation, data management, console, system configuration, and experiment management. In view navigation, users can switch to different views (e.g., training view, positioning view, console view, system configuration view) via touch activities. The system administrator can check system run-time status in the console view. We can set system preferences in the system configuration module (e.g., we can choose to enable/disable the user feedback model or store all raw experimental data). Another very important module in this layer is the experiment management. Since we

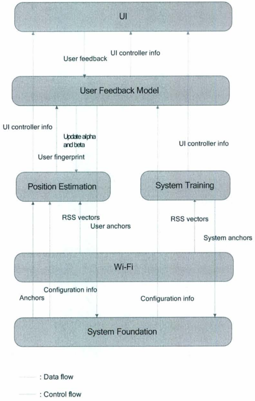


Figure 5.1: System architecture.

will conduct experiments with different settings and parameters, it is beneficial to maintain each experiment individually. We implement a file bundle which contains all relevant information for each experiment. As such, we can conveniently switch among experiments or initiate a new experiment without losing data from previous experiments.

The RSS vector is formed in the *Wi-Fi* layer. It is converted to an RSS fingerprint in the System Foundation. The RSS fingerprints are assembled into system anchors and user anchors in the *System Training* and *User Feedback Model* respectively. The Wi-Fi layer directly communicates with iOS kernels via a private framework called WiFiManager, which provides a high-level wrapper for the Wi-Fi related system calls (Apple80211). The Apple80211 is a set of system calls which are related to Wi-Fi functionalities. Some important Apple 802.11 system calls are `Apple80211Open`, `Apple80211Close`, `Apple80211Associate`, and `Apple80211Scan`. In our case, the `Apple80211Scan` is mostly used to scan nearby Wi-Fi APs. It will generate an array, where each element is a dictionary structure that contains information about an AP (e.g., MAC, SSID, RSS, Channel, etc.). The detailed data structure is provided in the `WFNetwork` class in Appendix A.3.

The *System Training* layer implements most of functions required for system training. At each survey point, it periodically calls the Wi-Fi scan function in the Wi-Fi layer to generate RSS vectors and sends them to the System Foundation to assemble the RSS fingerprint. Then, the RSS fingerprints and the physical coordinates of the survey point are combined to form system anchors. We use a mutable list data structure to maintain the system anchors. The administrators can add, remove, or modify RSS fingerprints of system anchors. The trained survey points are marked on a map as reminders, allowing the system administrators to keep track of which survey points are trained and which are not.

The *Position Estimation* layer estimates a user's position and the degree of certainty in this location estimation. For experimental purposes, the prototype implementation allows a variable number of Wi-Fi scans. After a scan, it calculates the likelihood for each anchor using the generated RSS vector. Then the likelihood vector is compensated by the α factors of corresponding user anchors and β factors of selected top-k anchors maintained by the User Feedback Model. Then, the Position Estimation layer selects a representative (vertex 1-centre) from these top-k anchors and delivers it to the UI layer. Besides the estimated position shown in the UI layer, the region of uncertainty will also be presented to users for providing additional position information. The uncertain area is a circle enclosing all top-k anchors because they all have a large possibility of being the true position. From a usability perspective, it is more informative to present these high possibility anchors in a manner that allows them to understand the range of possible locations, but also in a manner that will not confuse them as to where the system has estimated their position.

The User Feedback Model layer receives and processes user feedback from the system UI. If a user provides a positive feedback, it generates a temporary user anchor by combining the user-suggested position from the UI layer and the RSS fingerprint from the Position Estimation layer. Such a temporary user anchor will be either merged with an existing user anchor or considered as a new user anchor based on the similarity calculation. In either case, the α vector will be updated. If the User Feedback Model layer receives negative feedback, it will update β vector by reducing the corresponding β value of the top-k anchors.

The UI layer contains different views which are controlled by the lower layers to enable user interaction and present system results. It visualizes training and positioning results to end users, receives user feedback and delivers them to the User Feedback Model layer. In order to provide a better understanding about our prototype, we will explain these views in the subsequence section.

5.3 User Interface

Since our prototype is built on iOS, the touch-based user interaction enables a superior user experience on the mobile devices. The goal of our touch-based UI design is to make the human-centered positioning activities as simple and efficient as possible. Next, we will introduce the UI of our prototype in detail.

5.3.1 The Main Panel

The basic functionality of main panel is view navigation. Users are able to switch among views and modules in the main panel by touching corresponding icons (Figure 5.2).

5.3.2 The trainingView

The main component of the `trainingView` is a scrollable map view which enables a trainer to zoom in/out and locate survey positions. When system administrators touch the map, the `trainingView` will record the current touch position. Then, a mutable table of survey positions will be loaded for the trainer to manipulate system anchors. In addition, the survey positions will be marked with tags on the map in

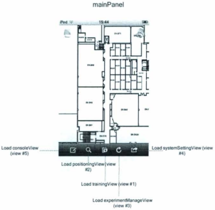


Figure 5.2: Main panel

order to remind the trainer whether they are associated with any RSS fingerprint (trained), as we can see in Figure 5.3.

System administrators maintain system anchors using the `surveyPositionsView` presenting mutable table (Figure 5.4). At each survey point, system administrators can create, delete or renew their associated RSS fingerprints. The create or renew action is triggered by touching one of the table cells to start a new sequence of Wi-Fi scans.

When a table cell is touched, the `surveyPositionsView` will load the `WiFiScanView` to start a new Wi-Fi sampling process at that survey position. The main component of the `WiFiScanView` is a table which contains all AP information in one scan, as shown in Figure 5.5. Multiple Wi-Fi scans will be conducted to collect as much in-

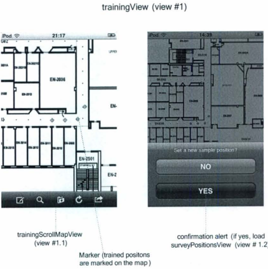


Figure 5.3: The `trainingView` provides an interface to system administrators. The physical coordinates of touched points on the map will be recorded and used to create system anchor.

surveyPositionsView (view #1.2)

Figure 5.4: System anchors are maintained in `surveyPositionsView`.

formation as possible. System trainers can also stop/resume Wi-Fi scanning. When it is finished, this survey point is associated with an RSS fingerprint.

5.3.3 The positioningView

The `positioningView` shown in Figure 5.6 is a root navigation view for users' positioning activities. It contains a scrollable map which presents system result and receives user's suggestion position. The "finder" icon can be touched to load the `positioningStart` for variable Wi-Fi scan number selection. When the system returns a position estimate, the `positioningView` will ask users to provide feedback via loading the `userFeedbackView`. Otherwise, users can touch the "notebook" icon to load the `userFeedbackView` and provide feedback.

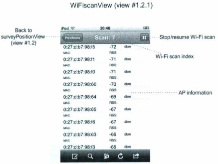


Figure 5.5: When entering the the `WiFiScanView` via selecting a survey point, system will start the Wi-Fi scan and generate Wi-Fi RSS vectors.

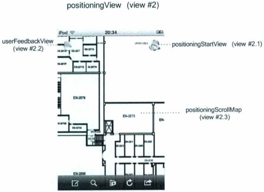


Figure 5.6: The positioningView.

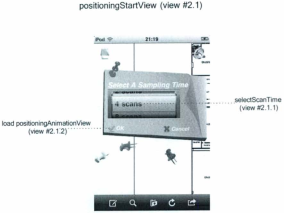


Figure 5.7: The picker in `positioningStart` is used to select a Wi-Fi scan number.

The purpose of `positioningStart` is to allow a variable number of Wi-Fi scans in positioning. We can select a Wi-Fi scan number from a picker as shown in Figure 5.7.

During the position calculation, the system will generate massive intermediate results (e.g., intermediate uncertain area and estimated position). For example, if the Wi-Fi scan number is four, the system will generate four uncertain areas and estimate positions for each cumulative scan before the positioning is finished. Each scan may lead to a cumulative estimation result with a decreasing error because more AP information is collected. The `positioningAnimationView` presents animations showing a gradually decreasing uncertain area (the area of circle). At the same time, users will be experiencing a more accurate position estimation (the pin is approaching

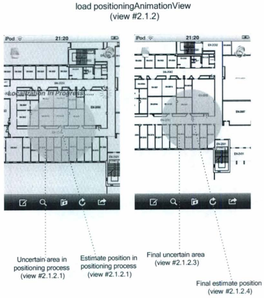


Figure 5.8: The positioningAnimationView provides intermediate system results to users.

to user's true position), as we can see in Figure 5.8.

5.3.4 The userFeedbackView

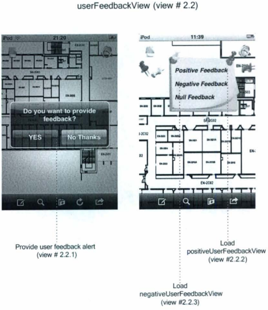


Figure 5.9: The `userFeedbackView` enables users to provide three kinds of feedback.

When the positioning process is complete, the system will ask users whether or not they want to provide feedback. If yes, it loads the `userFeedbackView` as shown in Figure 5.9. Users can provide three kinds of feedback (i.e., positive, negative, and null user feedback) by touching a corresponding tag.

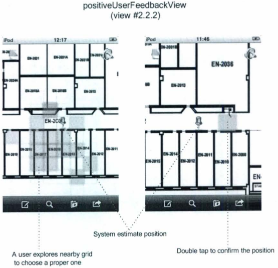


Figure 5.10: The `positiveFeedbackView` enables users to explore grid cells for positive feedback, confirming this choice with a double tap.

If users choose to provide positive feedback, they need to touch the map to suggest a new position. However, users may need to explore the map to make a satisfactory estimation. Also, the size of the finger touching on screen may generate errors. In order to solve this problem, users are able to explore surrounding grid cells and double tap to confirm their estimation in our implementation 5.10.

If users choose to provide negative feedback, the `negativeFeedback` will place a

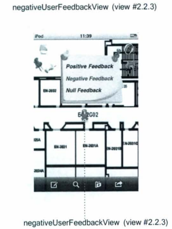


Figure 5.11: A red cross placed on system estimate position means a rejection.

red cross on the system estimated position to indicate a rejection, as shown in Figure 5.11. If users trust the system's estimation, they can just choose the null feedback with a simple close

5.3.5 The experimentManagementView

In `experimentManagementView` (Figure 5.12), we use a picker to select a build-in experimental environment, which includes the vectors containing α and β of the model, created user anchors, and historical results of a experiment. Otherwise, we can start an new experiment by touching the "add" icon.

experimentManageView (view #3)

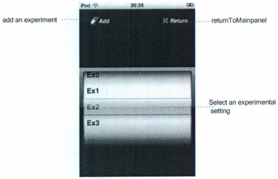


Figure 5.12: The `experimentManagementView` allows us to select different built-in experimental environments or create a new one.

5.3.6 The systemSettingView

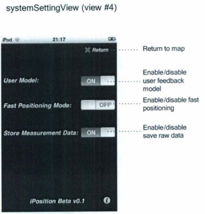


Figure 5.13: The `systemSettingView` enables system configuration and preferences setting.

The `systemSettingView` provides an interface for system preferences setting (Figure 5.13). The proposed user feedback model can be turned on/off in order to compare the system performance with/without user feedback. The fast positioning mode means the system will only use the first Wi-Fi scan to estimate user's true position. It is fast but may have large positioning errors. When conducting experiments, we can also choose to to save all raw experimental data or not in the `systemSettingView`.

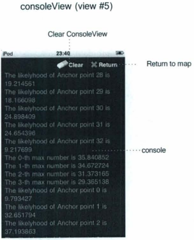


Figure 5.14: The consoleView

5.3.7 The consoleView

The `consoleView` displays detailed system information (e.g., likelihood for each anchor, the index of returned anchors, etc) for system analysis.

Since the prototype is available, we can conduct experiments to evaluate it and the proposed human-centric collaborative user feedback model. In the next chapter, we will evaluate the baseline Wi-Fi fingerprinting-based system and our user feedback model.

Chapter 6

Evaluation

We will explain and interpret experiment methodology, settings, scenarios, and results in this chapter. Our main experimental goal is to measure the benefit of adding human-centric feedback to a baseline indoor positioning system.

6.1 Methodology

The system evaluation contains two phases. The first phase is to analyze the performance of the baseline system without user feedback in field tests. The accuracy and precision of the baseline system will be calculated. By analyzing these two performance metrics, we can determine whether or not our baseline system is suitable when compared to experimental results of other similar Wi-Fi fingerprinting based indoor positioning systems. Furthermore, the time consumed in positioning is an important aspect of service quality and user experience. Typically, long Wi-Fi scan durations tend to bring more reliable results. However, the users might not be willing to spend too much time waiting for the results. Thus, experiments will also be designed to investigate the relationship between the time consumed in positioning process and system performance. For our baseline positioning system, the number of Wi-Fi scans dominates the positioning duration. Experiments will be conducted to compare the average positioning error and precision in terms of the number of Wi-Fi scans.

Next, we will explore how the proposed user feedback model improves the system performance. We will measure the benefit of integrating the proposed human-centric collaborative user feedback into a Wi-Fi fingerprinting-based indoor positioning system from the following three aspects:

- *Cost*

  Hypothesis 1: *The system training and maintenance cost can be reduced.* The training effort is reduced if system administrators only train the major part of the objective positioning area or train the system at a coarse granularity (i.e., large grid space). However, in doing so, the positioning accuracy and precision will be reduced. More importantly, if the indoor environment changes (e.g., Wi-Fi infrastructure or environment layout alteration), the RSS fingerprints database has to be updated frequently or even re-generated from scratch in order to adapt to such changes. At this point, we argue that such system training and maintenance cost can be reduced by taking advantage of human-centric collaborative feedback.

- *System performance*

  Hypothesis 2: *The system performance can be improved, as measured in accuracy and precision.* The newly created user anchors in fact include the "fresh" Wi-Fi RSS data, which can best characterize the current Wi-Fi environment.

If we can keep integrating such data into our fingerprint database, future positioning queries can take advantages of the timely knowledge shared by other users, resulting in an improvement in system performance. Furthermore, the resolution of the positioning system should be gradually increased because user anchors are generated between system anchors. As such, the grid space is reduced and the positioning resolution is refined.

- *Robustness*

  Hypothesis 3: *The system is robust with respect to malicious user feedback.* One potential risk of opening a user channel to the positioning database is that malicious user feedback will disturb the functionality of system anchors. Thus, the proposed user feedback model should have considerable robustness to the interference or even intended attacks from malicious user feedback. In the worst case, the system is continuously interfered with by malicious user feedback, which could lead a very large average positioning error. However, after the attack stops, the system should be able to recover from the low accuracy state by integrating benign and knowledgeable feedback.

We will discuss the experiments designed to validate these hypotheses in subsequent sections.

6.2 Experimental settings

Experiments and evaluations with this feedback model were conducted in an indoor office environment, which is the part of the 2nd floor of the Engineering Building at

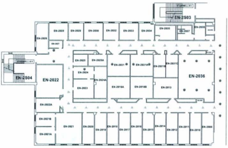


Figure 6.1: The experimental field includes both the training cells (green triangles) as well as measurements taken outside of the training area (red discs).

Memorial University. The space was divided into a grid using a 3×3m cell size. 33 positions were selected within the hallways for training the baseline system (denoted the *training area*), and an additional 20 positions were selected as untrained positions for testing purposes (denoted the *non-training area*). A diagram of the setting is provided in Figure 6.1. System anchors were created in the training area only. The non-training area lacks valid system or user anchors. It can be treated as the result of environment alteration, new Wi-Fi coverage area, or a neglected region.

As mentioned earlier in Chapter 4, the parameters in the feedback model are used to adjust the rate of change of the α and β factors (i.e., the sensitivity of our user feedback model). In production environments, the sensitivity of the user feedback model will depend on the number of users and the degree of trust in those users. For the purpose of evaluation, we increase the sensitivity of the user feedback model in order to speed up the the rate at which the system is able to learn from the user feedback. Based on these principles and our experimental settings, we set the value of parameter a to be 1, which means that the magnification factor of parameter α is 2. The value of parameter b is set to be 0.6. As such, according to the design of our user feedback, these parameter setting will weight the first four users much larger than subsequent users.

6.3 Baseline System Evaluation

Since the time that a user is willing to spend waiting for a positioning result influences the service quality, we have conducted an experiment to investigate the relationship between time (i.e., the number of Wi-Fi scans) and system performance. We use the baseline system to determine the smallest number of Wi-Fi scans (measured at one scan per second) needed for the system to produce a reasonably accurate result. At the same time, the performance of our baseline system can be evaluated with respect to other similar systems described in the literature. In the training area, for each survey point, we have collected 20 scans of the Wi-Fi RSS, using these incrementally to query the positioning system. The average positioning error after each scan is plotted as the bottom curve in Figure 6.2. We can observe that for a small number of scans, the system has an error between 2 and 4m. As more scanned RSS data are used (i.e., greater than four), the accuracy stabilizes at around 2m.

The system precision, as another very important metric for system performance, is plotted in Figure 6.3. It specifies how often we could attain an accuracy. For example, if a positioning system can determine positions within 3m for about 90 % of the measurements, that particular system qualifies to be 90 % precise in providing 3m

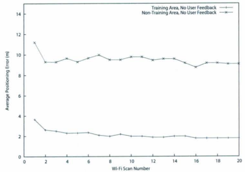


Figure 6.2: Using the baseline system, the positioning error becomes relatively stable using just four Wi-Fi scans. Note that the system is significantly more accurate within the training area.

accuracy. We selected the positioning precision for 9 out of the 20 scans, illustrating three phases of Wi-Fi sampling. The early phase consists scans 1, 2, and 3 (red curves). In this phase, due to the insufficient Wi-Fi RSS data, the precision is low. The second phase includes scans 5, 10, and 15 (green curves), it is in the middle of the Wi-Fi sampling and has more Wi-Fi RSS data than the first phase. The last phase is at the end of Wi-Fi sampling (scans 18, 19, and 20), which includes all RSS vectors (blue curves). From Figure 6.3, we can see that , the green and blue curves are very close to each other, which means that scan number larger than four will not generate significant precision improvement. However, if the Wi-Fi scan number is small (i.e., less than four), the probability of generating outliers is considerably high.

Similarly, in the non-training area, we also collected 20 scans for each position. We

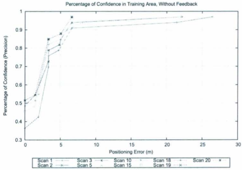


Figure 6.3: The precision of first three scans (red curve) is much lower than later scans (green curves for scan 5,10, and 15 and blue curves for 18, 19, and 20). However, the blue and green curves are very close to each other, indicating the precision after four scans is not improved significantly.

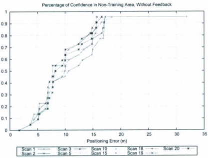


Figure 6.4: Similar precision trend can be found in non-training area, blue curves and green curves are similar but both apart from red curves.

plotted the positioning accuracy for the number of scans as the top curve in Figure 6.2 and positioning precision in Figure 6.4. In this case, the system performance is significantly lower than in the training area due to the lack of system anchors. However, in both training area and non-training area, four scans provide a reasonable trade-off among performance and positioning time. Therefore, we use this as the number of scans in the rest of our experiments.

According to the analysis of our baseline system, the average positioning error is between 2m and 4m, respectively, depending on the Wi-Fi sampling time. It is in fact only marginally worse than the 0.7m to 4m average positioning error yielded by the best-performing but intensively trained Horus system (using 100 Wi-Fi scans and much smaller grid space (1.52 m and 2.13 m)) [44]. Thus, we believe this baseline system is qualified to evaluate the value of adding the user feedback model.

6.4 User Feedback Model Evaluation

6.4.1 Knowledgable and Helpful Feedback

Next, we investigate how the user feedback model improves the system performance. In this scenario, whenever the system returns a position that does not match the true position of the user, feedback was provided. We modelled the user as being knowledgable and helpful; whenever the position was inaccurate, the user suggested positive feedback 80% of the time, and negative feedback 20% of the time. We believe it is a reasonable choice for situations where users are highly motivated to provide accurate and positive feedback. In fact, there may be many other users who are providing null feedback (i.e., using the system and trusting the results). However, since such types of users do not affect the evolution of the model, they are not discussed at this point.

Within the training area, we define a *round* as a traversal of all grid cells. In a round, the user stops at each survey position to scan the RSS for nearby APs (using four scans). If the result is correct, the user moves to the next position. Otherwise, the user provideds feedback before moving on. The average positioning accuracy after nine such rounds of visiting and testing each position is plotted in Figure 6.5. In the course of providing this user feedback, the positioning error within the training area improved from approximately 2.5m to 1.5m after just four rounds. From there, little change was observed. Note that the baseline system accuracy is from 4m to 2m without feedback. At this point, with the integration of human-centric collaborative feedback, the system performance is furthermore improved even in the well trained area.

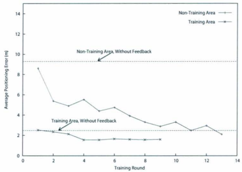


Figure 6.5: The system accuracy is significantly improved when integrating knowledgeable and helpful user feedback.

The precision is also improved after four rounds of user-involved positioning within the training area, as we can see in the green and blue curves which are closer to the y axis than red curves shown in Figure 6.6. Furthermore, green and blue curves are close to each other, which indicates that the model reaches its optimal performance after approximately four rounds of knowledgeable and helpful feedback.

Within the non-training area, the experiment followed the same procedure as in the training area, producing the data plotted in Figure 6.5. Because there was no training data in these regions, the initial positioning error was rather large. However, after 13 rounds of collecting user feedback, the error decreased from 9m to 2m. The precision is also significantly increased as plotted in Figure 6.7. As a result, the system performance in an area that had not been previously trained became comparable to the training area.

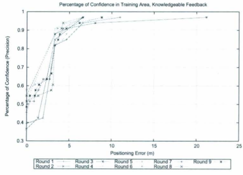


Figure 6.6: In training area, the precision is improved via integrating knowledgeable user feedback. The green curves and blue curves are close, which indicates that the model is optimally trained after four rounds.

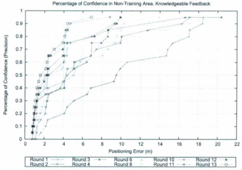


Figure 6.7: In non-training area, the system precision is significantly increased as more and more knowledgeable user feedback is integrated.

At the beginning of the testing within the non-training area, the model contained only system anchors, and therefore could only return the position of a system anchor (i.e., within the training area) to the user. These positions were often far from the true position of the user. As a result of the positive feedback, user anchors were added and the relative weight of these anchors were enhanced by the α factor. Similarly, with the negative feedback, the weight of the system anchors were reduced by the β factor. As a result, the positioning accuracy increased as more user anchors become valid candidate positions.

What this means for indoor positioning systems is that the system training and maintenance costs can be reduced significantly by relying on knowledgable and helpful end users working on a partially trained system, eventually achieving the same level of accuracy as a fully trained system. Also, the resolution of the positioning system is improved because many reliable user anchors fill the gap between system anchors, thus reducing the grid space or increasing the grid resolution.

At this point, the optimal combination of different types of user feedback is not considered. To conduct experiments testing each possible combination is impractical within a limited time period. In fact, this problem can be explored if we could use a simulation testbed. We can collect a large amount of real Wi-Fi RSS data to simulate the Wi-Fi scans. When the simulated positioning process is finished, virtual user positive or negative feedback can be generated to the evolve the model. As such, the system performance with an arbitrary combination of positive and negative feedback could be estimated.

6.4.2 Mixed Feedback

In a real environment, user feedback can be either helpful or malicious. In this experiment, we test the model to determine its ability to recover from incorrect feedback. In particular, we model the user feedback as completely malicious at the beginning and as completely informative thereafter. Such a behaviour is not typical but it provides a "worst case scenario" study of the system, followed by its ability to recover from incorrect or malicious feedback.

Our focus here is on the training area only. As seen in the previous experiments, the non-training area can become nearly as good as the training area with sufficient user feedback. As such, we expect similar results within the non-training area as the training area with respect to mixed feedback.

During the initial phase of this experiment, whenever the system returns a correct

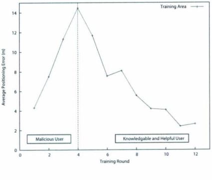


Figure 6.8: Providing malicious user feedback, followed by knowledgeable and helpful user feedback illustrates the ability of the model to self-recover.

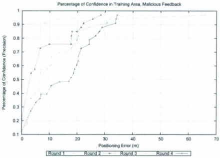


Figure 6.9: Providing malicious user feedback also reduces the system precision significantly.

position estimation, the malicious user has a 50% chance of either providing negative feedback of suggesting a random false position. When the system is incorrect, the malicious user provides null feedback. Following a similar methodology as the previous experiments, such malicious feedback was provided for four rounds. Another eight rounds of feedback from a knowledgeable and helpful user was then collected.

The position errors for this experiment are plotted in Figure 6.8. We observe that the system error starts out with around 4m and quickly increases to 14m as a result of the malicious feedback. At the same time, the system precision is also reduced to an unacceptable level, shown as the red curves in Figure 6.9. With an error of 14m and extremely low precision, the system is considered to be fairly disturbed by the malicious users. At this point, we turn the user into knowledgable and helpful to provide positive feedback whenever the system is incorrect. The user behaviour

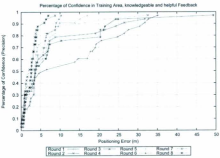


Figure 6.10: Providing malicious user feedback, followed by knowledgeable and helpful user feedback also recovers the system precision to a normal level.

in this case is the same as in the previous subsection. The helpful feedback quickly corrects the significant positioning errors, recovering to the starting accuracy after five rounds of feedback, and below 3m after eight rounds. At the same time, the system precision is stabilized as indicating by the blue curves in Figure 6.10.

As a result, our system has recovered from the low accurate state by integrating helpful and knowledgeable feedback.

To conclude this chapter, we will review the entire evaluation process and whether the hypotheses proposed in Section 6.1 are validated. We have designed three different experimental scenarios and divided the study area into two areas, i.e., training area and non-training area. Positioning in the non-training area only relies on the anchor points in the training area, which could cause large errors.

In Scenario 1, the impact of Wi-Fi scan number on system performance has been studied. Within both the training area and non-training area, we found that four scans are able to obtain fairly good system performance without adding too much system overhead.

In Scenario 2, knowledgeable and helpful user feedback has been integrated into the model. This scenario is designed to present the maximum improvement of the baseline system.

The "worst case scenario" study has been conducted in Scenario 3. That is, the model has been designed to endure continuous malicious feedback (attack). As such, the positioning error has gradually increased. When the system has eventually became unreliable, knowledgeable and helpful user feedback used in the second scenario is fed into the model in order to measure its self-recovering ability.

Next, we will validate the hypotheses via experimental results from above scenarios.

- **Hypothesis 1**: *The system training and maintenance cost can be reduced.* As mentioned above, non-training area exists in real indoor environments, i.e., the area without anchor points and the neglected or modified regions within the training area. In both cases, system administrators need to frequently update the fingerprint database in order to ensure the system reliability. With the integration of knowledgeable and helpful user feedback, such a maintenance cost is reduced as illustrated by the experimental results from Scenario 2. During the initial rounds, the positioning error is about 9m. With more and more user feedback provided, the positioning error is reduced to 2m. The system precision is also considerably improved as we can see from the blue curves in Figure 6.7.

Such performance can be considered as the same level as the performance in training area, however, without any extra training or maintenance cost. Thus, the Hypothesis 1 is validated.

- **Hypothesis 2**: *The system performance can be improved, as measured in accuracy and precision.* The reliable user feedback contains information (user fingerprint) that best characterizes the current Wi-Fi RSS features. Such helpful information can help the system to improve the performance as illustrated by the experimental results of the training-area in Scenario 2. The system positioning error is reduced from around 2.5m to around 1.5m after four rounds and becomes stabilized. Figure 6.6 indicates that the forgotten area in the training area is gradually eliminated since the blue curves are more vertical than red curves. This indicates that the system performance can be improved with the integration of helpful and knowledgeable user feedback, which validates Hypothesis 2.

  In addition, when more and more user anchors become valid, the system resolution is refined because they reduce the grid space between system anchors. This indicates that system administrators can use coarse granularity during the training-phase, and prompt the user to provide feedback in order to refine the system resolution. Thus, the training cost is reduced, which validates Hypothesis 1 from another aspect.

- **Hypothesis 3**: *The system is robust with respect to malicious user feedback.* In real life, helpful and malicious feedback are often mixed together to feed the model. As such, the phenomena described in Scenario 3 might be rarely observed. However, Scenario 3 in fact provides the "worst-case". If the model can eliminate the negative effect introduced by continuous malicious or unreliable user feedback, then it is reasonable to deduce that it is robust to malicious user feedback in more moderate or general cases. According to the experimental results in Scenario 3, the model is shown to be robust with respect to malicious feedback, quickly recovering to normal performance level with helpful user feedback. As a result, the last hypothesis is validated.

Chapter 7

Conclusion and Future Work

7.1 Primary Contributions

Wi-Fi RSS fingerprinting is relatively robust, accurate, and cost-effective in real indoor environments because it does not depend on specific signal propagation models or extra positioning infrastructure. However, its system performance is highly dependant upon the elaborate training process and future maintenance efforts. Also, in the positioning phase, random propagation effects of signal propogation introduced by complex indoor environments may result in large RSS fluctuations or AP loss (i.e., APs which cannot be heard), which could cause anchor points created during the training phase to be ineffective for the task of positioning. In order to ensure that the system remains effective, it may be necessary to re-train the system on a somewhat frequent basis.

These shortcomings not only imply a high system overhead and training cost, but also vulnerability to environmental alteration. We believe that addressing the problems of reducing the training and maintaining cost and increasing the system robustness are very promising research directions. As such, we set our main research goal to enhance such a system with a self-learning or self-updating ability. The information necessary for this process can be derived directly from end users, whereby they can provide their feedback after they have been presented with the positioning results. We thus open a channel for end users to access and modify the system inbuilt dataset, which enables them to participate in positioning activities via a well-designed UI. We believe that this human-centric collaborative positioning mechanism could effectively facilitate the system learning process.

In this thesis work, the primary contribution is the presentation and evaluation of a user feedback model which receives and processes human-centric collaborative feedback. The proposed user feedback model adjusts systems result via placing a compensation mask over the likelihood vector (distribution) generated in the positioning phase. The history of both positive feedback and negative feedback will affect the compensation ability of such a mask. In general, positive feedback generates user anchors and enhance their reliability. On the other hand, negative feedback reduces the reliability. All user feedback will be assigned low compensation power when first created and be enhanced with similar feedback events. We employ exponential functions to model such an evolution process; the influence of user feedback will increase fastest with the first few instances, becoming stable once a sufficient number of feedback events are received. This design allows the system to quickly learn new information provided by the users, but without this feedback overpowering the model. As such, this user feedback model should be able to gradually update the systems knowledge and guide the system to learn the changes of Wi-Fi indoor environments.

Based on these principles, we have built a prototype and conducted experiments to evaluate it. Experimental results show the ability of the model to improve upon the positioning accuracy and precision in both regions that have been trained, as well as in nearby regions that do not include efficient anchors. The model is also shown to be robust with respect to malicious feedback, quickly recovering based on helpful user feedback.

7.2 Discussions and Future Work

We also believe that such a feedback model can be further refined and enhanced in a number of interesting ways. The first is the temporal aspect of user feedback since different time (morning, noon, and night) of a day or date (weekdays, weekends, and holidays) could generate different RSS data patterns. For example, in a university cafeteria, due to the interferences from human bodies and electronic devices, the RSS measurement generated during dinner time could be very different from that in the morning. As such, the user feedback generated during dinner time may mislead the positioning activities during the morning. In order to solve this problem, the model should take advantage of the timestamp within the RSS fingerprint, limiting the candidate anchors to those that were created at approximately the same time of the day. This could increase the accuracy of the system in environments with time-related changes in human activities. Also, we can introduce a forgetting mechanism which could remove user feedback from the database. It could be used to address situations where malicious feedback has been received but subsequent helpful feedback is not available.

The second is the way to prompt of user feedback. The system seems to be helpful if users are frequently asked for feedback. On the other hand, the system will not evolve if no user feedback is received. It is beneficial for the system to know when and where to ask for feedback from users. Thus, we need a user prompting mechanism. We want to convey the system status (e.g., positioning uncertainty to users) so that the users only provide feedback when the system is unreliable.

The third aspect is cross platform validation. In real indoor environments, users could carry different types of mobile devices. Due to the diversity of manufacture technologies in wireless network interface cards, the RSS generated by different Wi-Fi chips could also be different. However, our entire implementation and experiments are conducted on Apple iPhone and iPod Touch, which indicates its limitation in field validation. At this point, we argue that the system performance could be improved if the diversity of Wi-Fi chips in different mobile devices is considered. The most simple but efficient is approach to create individual fingerprints database for each type of mobile device. It might improve system performance with high system overhead. More intelligently, a RSS compensation mechanism can be integrated to automatically adjust RSS patterns among different mobile devices.

As defined in previous chapters, our system includes many dependent and independent parameters. A long-duration study of user involved positioning should be helpful in order to investigate the effect of different parameters on system performance, We could operate the system for a long time (e.g., a year) with a great deal of users working on a variety of devices. As such, the experimental results of this long-time evaluation will provide further, real-world validation of our user feedback model.

In our implementation of user feedback model, the parameters are adjusted dynamically via a pre-specified formula. With the statistical results, we could have the ability to find a more efficient algorithm to adjust these parameters based on real-time system performance.

Also, in our previous experiments, we merely consider the worst-case scenario. However, if we can take the advantage of long-time evaluation, we can study the phenomenon of mixed user feedback (malicious and knowledgeable user feedback) and try to find more effective ways to detect malicious feedback.

However, if such resources are not available, an alternative is to simulate such feedback. We can in fact build an add-on experimental positioning engine to simulate RSS observations and send them to the positioning system. When the positioning is finished, such an engine can also generate virtual user feedback according to specifical experimental requirements. As such, the different combinations of parameters in the model can be conveniently tested without the actual time-consuming system evolution.

As an expectation, we believe that some organizations or companies will devise specifications for indoor positioning system in the short future. It may start with developing indoor location-aware services for public indoor environments, such as airport, subway systems, museum, campus, shopping centres, etc. Travelers may want to find the nearest coffee shops or ATM machine in a large airport. Customers are enabled to manipulate their location-aware shopping list. We can also easily find their friends or families if they have wandered away from each other in a very crowed shopping centre. Beside these, many more other potential services can come true with the development of indoor positioning systems. Such large scale of indoor services might be provided and end users can conveniently access them via their mobile devices at hand. If successes can be made in such public areas, other private enterprisers may be inspired to customize their own indoor location-aware services. By following existing specifications, high scalability and compatibility can be guaranteed.

With the potential rapid growth of indoor positioning systems, the system maintenance could become an issue. At that time, the human-centric indoor positioning systems will have a very promising foreground in reducing the cost and improving the service quality. We also believe that more and more researchers will be attracted by the potential advantages of integrating human-centric collaborative feedback within the positioning process.

Bibliography

[1] Wireshark documentation. Wireshark, 2010. http://www.wireshark.org/docs/.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of 19th IEEE Conference on Computer Communications*, pages 775–784, 2000.

[3] E. S. Bhasker, S. W. Brown, and W. G. Griswold. Employing user feedback for fast, accurate, low-maintenance geolocationing. In *Proceedings of the 2rd IEEE International Conference on Pervasive Computing and Communications*, pages 111–120, 2004.

[4] P. Bolliger. Redpin-adaptive, zero-configuration indoor localization through user collaboration. In *Proceedings of the 1st ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, pages 55–60, 2008.

[5] M. Brunato and C. K. Kalló. Transparent location fingerprinting for wireless services. Technical Report DIT-02-071, University of Trento, 2002.

[6] A. Carlotto, M. Parodi, C. Bonamico, F. Lavagetto, and M. Valla. Proximity classification for mobile devices using Wi-Fi environment similarity. In *Proceed-*

ings of the 1st ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments, pages 43–48, 2008.

[7] X. Chai and Q. Yang. Reducing the calibration effort for location estimation using unlabeled samples. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, pages 95–104, 2005.

[8] Y. Chen, Q. Yang, J. Yin, and X. Chai. Power-efficient access-point selection for indoor location estimation. *IEEE Transactions on Knowledge and Data Engineering*, 18(7):877–888, 2006.

[9] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proceedings of 16th Annual International Conference on Mobile Computing and Networking*, pages 173–184, 2010.

[10] T. Gallagher, B. Li, A. G. Dempster, and C. Rizos. Database updating through user feedback in fingerprint-based Wi-Fi location systems. In *Proceedings of Ubiquitous Positioning Indoor Navigation and Location Based Service*, pages 1–8, 2010.

[11] M. Gast. *802.11 Wireless Networks: The Definitive Guide.* Colleen Gorman, 2nd edition, 2005.

[12] J. geun Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an organic indoor location system. In *Proceedings of 8th International Conference on Mobile Systems, Applications, and Services*, pages 271–284, 2010.

[13] G. M. Giaglis, A. Pateli, K. Fouskas, P. Kourouthanassis, and A. Tsamakos. On the potential use of mobile positioning technologies in indoor environments. In *Proceedings of the 15th Bled Electronic Commerce Conference eReality: Constructing the eEconomy*, pages 413–429, 2003.

[14] Y. Gu, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. *IEEE Communications Surveys and Tutorials*, 11(1):13–32, 2009.

[15] 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 *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pages 70–84, 2004.

[16] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. *Wireless Networks*, 8(2):187–197, 2002.

[17] J. Hightower and G. Borriello. Location sensing technique. Technical Report UW-CSE-01-07-01, University of Washington, Computer Science and Engineering, 2001.

[18] A. M. Hossain, H. N. Van, and W.-S. Soh. Utilization of user feedback in indoor positioning system. *Pervasive and Mobile Computing*, 6(4):467–481, 2010.

[19] K. Kaemarungsi and P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. In *Proceedings of 23rd the IEEE Conference on Computer Communications*, pages 1012–1022, 2004.

[20] K. Kaemarungsi and P. Krishnamurthy. Properties of indoor received signal strength for WLAN location fingerprinting. In *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 14–23, 2004.

[21] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. I: The p-Centers. *SIAM Journal on Applied Mathematics*, 37(3):513–538, 1979.

[22] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg. COMPASS: A probabilistic indoor positioning system based on 802.11 and digital compasses. In *Proceedings of the 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, pages 34–40, 2006.

[23] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos. Kernel-based positioning in wireless local area networks. *IEEE Transactions on Mobile Computing*, 6(6):689–705, 2007.

[24] A. M. Ladd, K. E. Bekris, A. P. Rudys, D. S. Wallach, and L. E. Kavraki. On the feasibility of using wireless ethernet for indoor localization. *IEEE Transactions on Wireless Communications*, 5(8):555–559, 2006.

[25] H. Lee, M. Wicke, B. Kusy, and L. Guibas. Localization of mobile users using trajectory matching. In *Proceedings of the 1st ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, pages 123–128, 2008.

[26] H. Lemelson, T. King, and W. Effelsberg. Pre-processing of fingerprints to improve the positioning accuracy of 802.11-based positioning systems. In *Proceedings of the 1st ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, pages 73–78, 2008.

[27] H. Lemelson, S. Kopf, T. King, and W. Effelsberg. Improvements for 802.11-based location fingerprinting systems. In *Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference*, pages 21–28, 2009.

[28] Y. Luo, Y. P. Chen, and O. Hoeber. Wi-Fi-based indoor positioning using human-centric collaborative feedback. In *Proceedings of 24th IEEE International Conference on Communications*, 2011.

[29] A. Martin, C. Ionut, and R. C. Romit. Surroundsense: Mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, pages 261–272, 2009.

[30] O. Nov. What motivates Wikipedians? *Communications of the ACM*, 50(11):60–64, 2007.

[31] E. Per and M. Pratap. Special issue on GPS: The global positioning system. *Proceedings of the IEEE*, 87(1):3–15, 1999.

[32] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 32–43, 2000.

[33] T. S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, 2nd edition, 2001.

[34] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen. A probabilistic approach to WLAN user location estimation. *International Journal of Wireless Information Networks*, 9(3):155–164, 2002.

[35] G. Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing.* Prentice Hall, 1971.

[36] B. Schölkopf and A. J. Smola. *Learning with Kernels.* MIT Press, 2002.

[37] S. Shen, T. J. Hart, N. Ingelbrecht, A. Zimmermann, J. Ekholm, N. Jones, and J. Edwards. Dataquest insight: The top 10 consumer mobile applications in 2012. Technical Report G00171017, Gartner, Inc, 2009.

[38] N. Swangmuang and P. Krishnamurthy. Location fingerprint analyses toward efficient indoor positioning. In *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications*, pages 100–109, 2008.

[39] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* The MIT Press, 2005.

[40] M. Vossiek, L. Wiebking, P. Gulden, J. Wiehardt, C. Hoffmann, and P. Heide. Wireless local positioning. *IEEE Microwave Magazine*, 4(4):77–86, 2003.

[41] Y. Wang, X. Jia, H. Lee, and G.Y.Li. An indoors wireless positioning system based on wireless local area network infrastructure. In *Proceedings of the 6th International Symposium on Satellite Navigation Technology Including Mobile Positioning and Location Services*, pages 179–186, 2003.

[42] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

[43] J. Yang and Y. Chen. Indoor localization using improved RSS-based lateration methods. In *Proceedings of the 28th Global Telecommunications Conference*, pages 4506–4511, 2009.

[44] M. Youssef and A. Agrawala. The Horus WLAN location determination system. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, pages 205–218, 2005.

Appendix A

Appendix

A.1 Index

Index

A.2 List of Notation

Symbol	Meaning
P_r	Random variable, represents the average of received power
$\overline{P_r}$	Mean of P_r
σ_{P_r}	Variance of P_r
P_{m_i}	The i-th received power in measurement
J	Least mean absolute error
p_i	The i-th received signal strength in dBm
$\overline{p_i}$	Mean of p_i in dBm
σ_{p_i}	Variance of p_i
t	Timestamp
e_i	Coefficient of i-th degree in polynomial regression
M_i	MAC address of i-th access point
R	Wi-Fi RSS vector
d_R	Dimension of RSS vector
$\mathcal{P}$	2-D Physical coordinates
V	Voronoi Diagram
v_i	Voronoi site
C_i	Number of occurrences of access point i
σ_i	RSS variance of access point i
F	Wi-Fi RSS fingerprint

σ_F	Variance of the Wi-Fi RSS fingerprint
s_f	Similarity of two Wi-Fi RSS fingerprints
A_s	System anchor
$\mathcal{P}_s$	The physical coordinates of system anchor
F_s	Wi-Fi RSS fingerprint of system anchor
$D(i,j)$	Euclidean distance from anchor point i to j
L_i	Likelihood of anchor i
L	Likelihood vector
A_u	User anchor
$\mathcal{P}_u$	Physical coordinates of user anchor
F_u	Wi-Fi RSS fingerprint of user anchor
α	Positive user feedback compensating factor, $\alpha = \frac{1}{a+e^{-x}}$
x	Independent variable accumulating via similar user feedback
T	Current Wi-Fi scan times
T_s	Wi-Fi scan times used in system training phase
Δx	Increment derives from a similar user feedback
a	Parameter adjusting the initial and maximum value of α

b	Parameter adjusting the increasing velocity of x
β	Negative user feedback compensating factor, $\beta = e^{-x}$
L_i'	Compensated likelihood of anchor i

A.3 System Class Diagrams

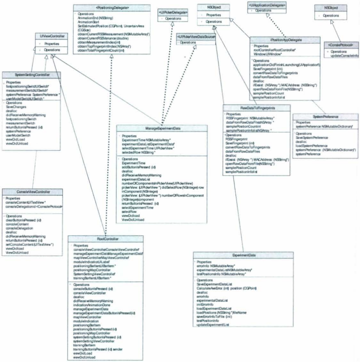


Figure A.1: Class diagram, System Foundation

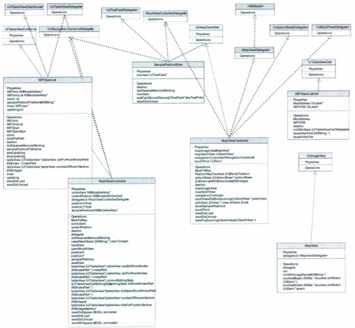


Figure A.2: Class diagram, `Training`

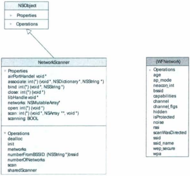


Figure A.3: Class diagram, `Wi-Fi`

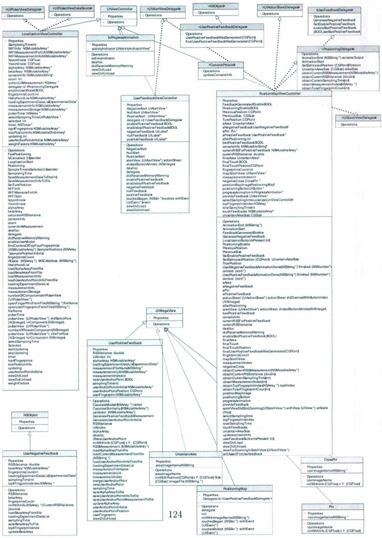


Figure A.4: Class diagram, the `Positioning Estimation` and `User Feedback`