MOTION ASSISTED INDOOR SMARTPHONE POSITIONING IN SPARSE WI-FI ENVIRONMENTS

WASIQ WAQAR

Motion Assisted Indoor Smartphone Positioning in Sparse Wi-Fi Environments

by

© Wasiq Waqar

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 2013

St. John's Newfoundland

Abstract

The current generation of smartphone devices equipped with embedded sensors like gyroscope, accelerometer and electronic compass, provide new opportunities for user positioning and tracking. In addition, the rapid growth of location based applications has spurred extensive research on localization. However localization in indoor environments still remains an elusive and challenging problem as GPS (Global Positioning System) does not work inside buildings and the accuracy of other localization techniques typically comes at the expense of additional infrastructure or cumbersome war-driving. Specifically, in places where Wi-Fi access points are sparsely deployed, localization becomes more challenging when relying only on Wi-Fi based technologies. For such environments, we propose a localization scheme which uses motion information from the smartphone's accelerometer, magnetometer, and gyroscope sensors to detect steps and estimate direction changes. At the same time, we use a Wi-Fi based fingerprinting technique for independent position estimation. These measurements along with an internal representation of the environment are combined using a Bayesian filter. This system will allow us to reduce the amount of training required and work in sparse Wi-Fi environments.

Acknowledgements

First of all, I would like to thank my supervisors sincerely, Dr. Yuanzhu Chen and Dr. Andrew Vardy, who guided me throughout my research program, provided insightful advices, and most importantly constantly encouraged me to do better. This thesis would not have been possible without the generous resources provided by the Department of Computer Science and Memorial University.

I am also thankful to may research group Wireless Networking and Mobile Computing Laboratory for providing a platform to discuss and share interesting ideas and enhancing our research and presentation skills.

I want to acknowledge all my peers in the department for helping me, supporting me and providing me a rich graduate student experience. Memorial University has been a tremendously rewarding experience and the time I spent here is truly unforgettable.

Last but not the least, I owe my deepest gratitude to my loving parents, Waqar Muhammad and Roohi Waqar, who have been the people who have inspired me and made me who I am today.

Contents

List of Figures

Chapter 1

Introduction

1.1 Smartphones for Sensor-Driven Computing

Recent advances in mobile devices, embedded sensors and hardware make it possible to envision a large scale wireless network of smart devices. Today's smartphones are programmable and come with a set of cheap yet powerful embedded sensors, such as a GPS reciever, accelerometer, gyroscope, digital compass, microphone, and camera, which are enabling a new generation of personal and participatory sensing applications. Each device can be viewed as a "virtual lens" acting as eyes and ears for the surrounding physical space [20].

The smartphone is emerging as the main technology platform in the mobile marketplace with the number of users expected to exceed one billion by 2014 [35]. The so-called third screen is increasingly finding itself at home alongside the TV and computer screens. Research has found that 60% of mobile web usage is now taking place indoors, bringing smartphones closer to the promise of being "always on" devices [18].

So we see Mark Weiser's vision becoming a reality [58].

Along with mobile devices, we also saw rapid advances in network technologies while network infrastructure became more extensive and more reliable. This new level of ubiquitous network connectivity and pervasive devices has ebabled a new category of context-aware applications. Context is any information which can be used to characterize the situation of an entity. An entity can be a person, place or object that is considered relevant to the interaction between a user and an application, including the application and users themselves. Hence, smartphones bring us new opportunities to exploit user context, and make innovative mobile applications.

1.2 Mobile Phone Location Based Services

There are various aspects of context that can be useful to personalize the service to the user. User identity, orientation, history, time, purpose of use, physical surroundings, system properties, social and cultural situation are different areas of context in which research is being done [39]. One of the most important dimensions of context is location. A user's location can be physical, logical or both. Physical or absolute location can be described by geo-referenced coordinates whereas logical location is relative, for example, inside a room or near some building. This information can be exploited in a variety of applications for instance, targeted advertisement, geo-social networking, gaming etc. We are already seeing its impact on different industries like tourism, marketing, information and emergency services. In recent years we have witnessed the explosion of Location Based Applications (LBAs) with the Apple iOS App Store alone having over 6400 LBAs [50]. The Android Market also has around

1000 LBAs with many applications being added on a daily basis [50]. Loopt [5], GeoLife [4], Foursquare [3], Dodgeball [1] and more recently Facebook Places [2] are a few examples which exploit location information of the user in their applications. With advances in mobile commerce and the further development of software related to mobile location, the LBAs market is forecast to reach $21 billion by 2015 with over 1.2 billion subscribers [13]. Researchers have been working on Location Based Services (LBS) for the past few decades and we see their applications in the form of vehicular tracking and other navigation based services. However, due to the growth of mobile devices, new opportunities and challenges have come to surface for e.g indoor mobile targeted advertisement and indoor position tracking.

1.3 Indoor Positioning and Tracking

In the past most of the attention was given to LBS in outdoor environments as GPS played the dominant role in localization. Recently, we are seeing a paradigm shift in the mobile applications market, where indoor LBS is being considered the new frontier. Due to the increasing number of mega size multi-level constructions like airports, shopping malls, universities and other facilities, people tend to spend more time indoors. People only spend 10-20% of their time outdoors [6]. Same research also indicates that more than 70% calls originate from indoors which indicates great potential fot indoor LBS.

In order to provide quality LBS, it is necessary to have a reliable, accurate, and real-time location estimation of the user/device. Localization techniques can be broadly classified into two categories, i.e infrastructure-based and infrastructure-less. They can be further categorized by core technologies used: cellular, Wi-Fi, GPS, Bluetooth, ultrasound, infrared, RFID (radio frequency identification), UWB (ultra wide bandwidth), or sensor-based.

Mobile devices, such as smartphones and music players, have recently begun to incorporate a powerful yet diverse set of sensors. These sensors include GPS reciever, microphones, cameras, proximity sensors, magnetometers, temperature sensors, accelorometers and gyroscopes. In the future, other sensors like altimeters, barometers, etc may be incorporated in these devices. Today, GPS provides localization outdoors, but precise indoor tracking of people remains an open research problem. Due to the small size of these smart devices, their ability to communicate with other devices, their considerable computing power and their nearly ubiquitous use in our society, these devices open up exciting new areas for localization and indoor positioning. Some of the systems which use these sensors for mobile positioning are mentioned in [15][16][34][59].

1.4 Research Questions

According to our literature survey, there has been a lot of progress in indoor localization technologies. Active RF techniques [22][41][54] (installing special hardware in the environment) can achieve an accuracy of around a few centimetres whereas Passive RF [8][23][29][30][33][36] (using existing infrastructure) can give a decent accuracy of few metres. Using active RF techniques is not scalable because every indoor environment is unique and to setup such infrastructure requires study of environment parameters which also adds to its cost. Passive RF techniques are getting more popular because of their scalability, but extensive calibration is required for such systems. Wi-Fi, GSM, Bluetooth and other RF technologies are used for such systems. Skyhook [50] uses a hybrid combination of GSM and Wi-Fi signals. First, it is assumed that the wireless radio map is long lived which is not the case as the topology of a network keeps changing over time. To cope with this problem, frequent war-driving may be required. *War-driving* is the process in which radio data and information is collected by going to the tagged locations in a vehicle and storing the information. Second, the coverage of such systems is also a limitation as there may be areas where such a radio map can not be obtained. Examples include footpaths between buildings, inside buildings or rooftops because it is difficult for vehicles to access these locations.

An IMU (Inertial Measurement Unit), is an electronic device that measures and returns an object's acceleration, orientation, and gravitational forces, using a combination of accelerometers and gyroscopes and sometimes also magnetometers. An Inertial Navigation System (INS) is a navigation aid that uses an IMU to continuously track the position, orientation, and velocity of an object without the need for external references. An INS can detect a change in its geographic position (longitude and lattitude), a change in its velocity (linear and angular), and a change in its orientation (rotation about an axis). It does this by measuring the linear and angular accelerations applied to the system. Since it requires no external reference (after initialization), it is not only scalable but also cost-effective. This concept is not new as aircrafts, ships, rockets, robots, and space vehicles make use of inertial guidance systems.

Smartphone accelerometers have been used in some mobile localization schemes in an assistive or collaborative manner. In Surroundsense [34], they are used as one of the parameters for creating a unique multidensional vector to distinguish between-different locations, whereas CompAcc [15] uses it to count the number of steps taken to estimate the distance travelled by a pedestrian. In other work [40][24][44][27], researchers have used accelerometer data to detect human activity such as walking, standing, climbing stairs, jogging, etc.

Another aspect not considered in most localization technologies is the time required to acquire position estimates. Most active radio frequency and passive radio frequency positioning schemes use complex algorithms to calculate the user's position. The response time of such systems depend on multiple factors including the technology used, number of radio scans required, size of the training data, processing power etc. This is why most indoor localization technologies fail to provide good real-time indoor mobile positioning and tracking. Another problem arises when RF signals are sporadic in a particular environment. Due to the placement of access points (AP) and cell towers, there might be areas where Radio Frequency (RF) signals are not available. Similarly there may be disruptions, in the RF signals due to sparsity of APs, limits on radio range, energy resources, and noise which may prevent RF based positioning from being precise. In these kinds of environment it is better to rely on IMUs for localization with opportunistic RF based position correction. This leads to our fundamental research question:

> **Can we use embedded inertial measurement Unit Sensors in mobile phones assisted by sporadic Wi-Fi signals to provide near real-time indoor positioning and tracking?**

Mobile phone accelerometers are noisy and in the presence of a gravitational gradient, they are unsuitable for determining the direction of distance moved. If 1g of acceleration is applied in a direction perpendicular to the direction of gravity, it is very difficult to determine which way the mobile phone has travelled. Here g is the acceleration due to gravity. Most of the simple accelerometer based applications fail to detect this if they are not using another sensor, for example a magnetometer, coupled with it. However electronic compasses are very noisy and they show a bias especially in indoor environments. We have to explore different kinds of filters to cater for such noise. The problem of distinguishing acceleration due to gravity from acceleration due to motion can be solved if the orientation is accurately tracked. Most of the current INS based mobile localization schemes either use only accelorometers or they use external custom made sensors attached to the device like multiple accelerometers or a different combinations of accelerometers, gyroscope and magnetometers. Some also use these customized sensors by attaching on to the foot or other body parts. Attaching external devices to a smartphone is not a feasible solution whereas accelerometer only solutions are quite inaccurate. With gyroscopes added as one of the new input sensors in smartphones, combinined with an accelerometer, movement in 6 degrees of freedom can be tracked. We believe with this added IMU sensor, we have an opportunity to build a mobile tracking system which is responsive and accurate for an indoor environment.

1.5 Organization of Thesis

The reminder of this thesis is organized as follows. We discuss related work in Chapter 2. A background of the popular localization techniques is surveyed. Chapter 3 will introduce the concept of mobile indoor pedestrian tracking and localization and how embedded sensors in the latest smartphones can provide an opportunity for mobile user tracking. This includes sections describing our step counter algorithm using the accelerometer and gyroscope. Chapter 4 focuses on Wi-Fi based positioning schemes and their reliance on existing Wi-Fi infrastructure. Chapter 5 discusses our system architecture of using a motion module in collaboration with Wi-Fi focusing on sporadic RF environments. Evaluation and performance of the system is described in Chapter 6. Chapter 7 reflects back on our system. Perspectives, conclusion and possible future work are discussed in this chapter.

Chapter 2

Related Work

This chapter discusses some of the related work already done in the area of localization for mobile devices.

2.1 Overview of Current Localization Technologies

GPS [19] based localization systems are widely successful in outdoor applications but they are not applicable for indoor environments since the radio transimissions from GPS satelites waves will be attenuated and scattered by roofs, walls and other objects.

There are several range-based techniques such as Time-Of-Arrival (TOA), Time Difference Of Arrival (TDOA), Angle-Of-Arrival, and Received Signal Strength Indication (RSSI) to estimate the distance from a particular device. Absolute location then can be computed using triangulation, trilateration, fingerprint matching or other probabilistic methods.

Using the techniques mentioned above, some cellular [53] and Wi-Fi [42] based solutions are proposed which are less accurate than the GPS but give better performance in indoor environments. Place Lab [14] creates a wireless map of a region by war-driving in the area. The wireless radio map is composed of sampled GPS locations, Wi-Fi Access Point (AP) MAC addresses with RSSI, and cellular towers cell-ids at these locations. When a user travels through the mapped area, it scans for beacons from such AP's and cellular towers. The list of collected information is then compared to the wireless radio map available to estimate its location.

Active Badge [54] is one of the early centralized indoor personal positioning system making use of infrared technology. Badges worn by personnel transmit a unique infrared signal every 10 seconds. Each office within a building is equipped with one or more networked sensors which detect these transmissions. The location of the badge can thus be determined on the basis of information provided by these sensors. However, to cope with its limited range and propagation problems caused by obstacles, Active Bat [22] was developed which used ultrasound pulses. The Cricket [41] location system uses a proximity-based lateration technique to calculate the absolute location by computing the difference between the arrival time of radio frequency signals and that of ultrasound. There are also systems available which use RFID and Ultrawideband technologies for locating objects inside the building.

Computer vision has also been used in localization. Microsoft's Easy Living [25] uses real-time 3D cameras to provide stereovision-positioning capabilities in a home environment. Design based on phone cameras [45] is also attempted yielding encouraging results at the room level but the performance deteriorates in areas like corridor corners.

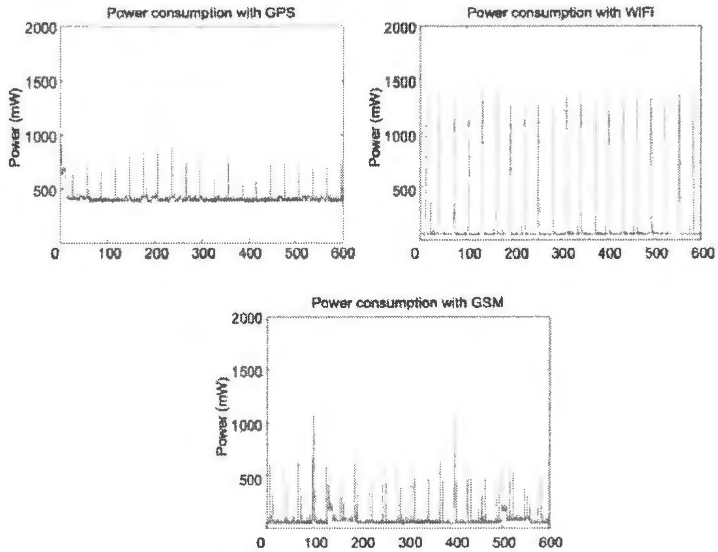
Amongst all the localization technologies mentioned in this section, Wi-Fi/cellular based solutions are the most popular [31]. Skyhook [50] collects raw data from Wi-Fi access points, GPS satellites and cell towers. It then uses advanced hybrid positioning algorithms to determine device position with 10 to 20 meter accuracy. These types of solutions are feasible for indoor environments and a valuable enhancement to GPS based localization as they reduce location acquisition time significantly. However, there is still room for considerable improvement.

Skyhook currently employs hundreds of drivers who continuously war-drive to create GSM/Wi-Fi maps of new regions and update the existing ones. Still, there are large areas which remains uncovered, including walking paths, shopping plazas, apartment buildings, parks and other indoor enviroments.

Relying on Skyhook like solutions has another problem. As they are dependent on GSM/Wi-Fi infrastructure, large portions of the world does not have such radio coverage. Hence, these solutions are not scalable. Furthermore, there is a trade-off between localization energy and accuracy [17]. GPS is more accurate but consumes more energy than both Wi-Fi and GSM based localization [12]. Figure 2.1 shows the power consumption comparison between GSM, Wi-Fi and GPS.

A lot of research is being done in activity recognition and wearable computing. The research in that area is now directly relevant to positioning and indoor localization due to the fact that similar sensors are being used [61]. Several papers have studied activity recognition using accelerometers [27][24][44]. Although most of the research assumes that sensors are fixed to human bodies, for example, hip, foot or elbow, their results are still motivating for smartphone devices.

Figure 2.1: Power Consumption among different technologies [17]



2.1.1 Performance Metrics for Evaluating Indoor Positioning Systems

Every positioning technology has its own strengths and weaknesses. Thus, it is very important to comprehensively evaluate an indoor positioning system from different aspects. In order to evaluate a localization scheme, there are many performance metrics available.

- Accuracy: Accuracy is the key metric for evaluating a localization technique. It is defined by how much the estimated position deviates from the true position.

- Precision: Precision indicates how often we expect to get the given accuracy. In other words it is the degree to which repeated measurements under unchanged conditions show the same results.

- Responsiveness: Responsiveness is defined as how quickly the location system outputs the location information. A long positioning delay will degrade the user experience and the perceived service quality. It is an important parameter, especially when dealing with mobility. Our system does not focus on this metric when evaluating the system.

- Scalability: Scalability is a very significant aspect of the system. It is the ease of deploying the system to new environments with random conditions. The positioning system should be robust with respect to large and complex environments.

- Calibration: Device Calibration is the process of forcing a device to conform to a given input/output mapping. In terms of Wi-Fi-based positioning it can mean the measurements taken as training data. Calibration plays a very important role as uncalibrated systems always have a lower accuracy.

- Cost: The cost of an indoor positioning includes the cost of the infrastructure installation, deployment, training and future maintenance. In fact, high indoor positioning accuracy can always be obtained if a massive number of sensors or anchor points are deployed, but often we cannot afford such a high deployment and maintenance cost. Another important cost factor when running the system in a real environment is power consumption. When scaling to thousands or millions of autonomous small devices, it is clearly not feasible to change or recharge batteries very often. Thus energy efficiency should be a goal of any localization mechanism meant for a large-scale system.

2.2 Sensor Driven Indoor Positioning

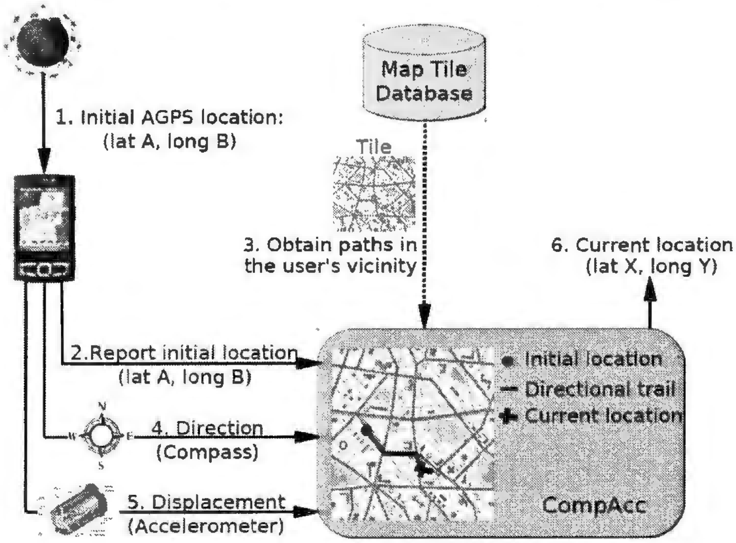
The proliferation of mobile phones is motivating researchers to look at other ways for more reliable and energy efficient indoor positioning of users which satisfy the criteria mentioned above. To minimize deployment and infrastructure costs, different techniques and technologies are being explored. In robotics, inertial sensors, laser range-finders and computer vision are used to provide accurate localization without the requirement of fixed infrastructure. One type of sensor which seems applicable to people tracking is inertial measurement units. Accelerometers and gyroscopes are being embedded in most of the latest smartphones. Accelerometers measure the 3D linear accelerations of the device whereas gyroscopes give the rotational speed. Most of these modern devices also include a magnetometer which can give raw magnetic readings and heading information.

2.2.1 Estimating Location

Most of the localization schemes are based on estimating the physical location of the entity. This can be absolute position analogous to GPS coordinate on a map or it can be a particular grid or anchor defined by a coordinate system in the environment. Some researchers have investigated pedometer based Pedestrian Dead Reckoning (PDR) techniques [15][48][60]. Woodman and Harle [60] showed that a foot mounted IMU can be used to track a user in a multi-floor building with a 0.5m accuracy for 75% of the time. They assume that the user does not know his or her starting positioning.

They evaluate their system compared to BAT [22] which is accurate up to 3cm

Figure 2.2: Flow of operations in CompAcc. [15]



for 95% of the time. Considering BAT system to be the ground truth and matching the positions estimated, their system gives an error less then 0.73m 95% of the time. Although they use foot-mounted IMUs, this kind of result is very promising for smartphone based IMUs. The IMUs in smartphones are much cheaper hence less accurate and more sensitive to noises, but it is assumed that a human-centric application does not need to be that accurate as humans can tolerate errors of a few metres if their context is fully satisfied.

CompAcc [15] is a scheme which deals with mobile phone localization without war driving. The flow of operations is shown in Figure 2.2. It uses electronic compasses and accelerometers in mobile phones to estimate walking patterns and matches it against possible path signatures generated from digital maps. Although CompAcc [15] was evaluated outdoors as it uses GPS correction in their implementation, it has great potential for indoor environment. CompAcc was tested offline as at the time a smartphone with both compass and accelerometer was not available to the authors. It was evaluated as a comparison to Skyhook. CompAcc's performance is much better than Skyhook which is biased towards roads and streets. Energy consumption of CompAcc is also much better than Skyhook and GPS according to their investigation. Although their system is not ready for deployment their results are very encouraging for similar indoor systems.

Escort [16] is a war-driving-free navigation system for social environments to route mobile users to other mobile users in an indoor setting. The system uses a beacon which transmits an audio tone. Any mobile phone, when passing near this, can register itself. This beacon then becomes the origin of a virtual coordinate system, where user path signatures and spatial intersections represent an edge and a vertex of a graph, respectively. This graph keeps track of user location and their trails. Using this graph, a general map of the location can be built to locate humans and route them to their destinations. An overview of the Escort system is shown in figure 2.3.

2.2.2 Classifying Logical Location

Some researchers argue, that physical location alone, unless remarkabaly precise, may not be sufficient to express the context of the user. For example (Figure 2.4), in a scenario to identify two logical locations separated by a dividing wall, Martin et al. [34] argue that even an idealized high accuracy localization scheme can place the user on the wrong side of the wall. AAMPL [40] uses GPS and Google Maps to shortlist possible logical locations and then uses accelerometer data to classify different logical locations for example the system positions the user to a cafe instead to a bookstore

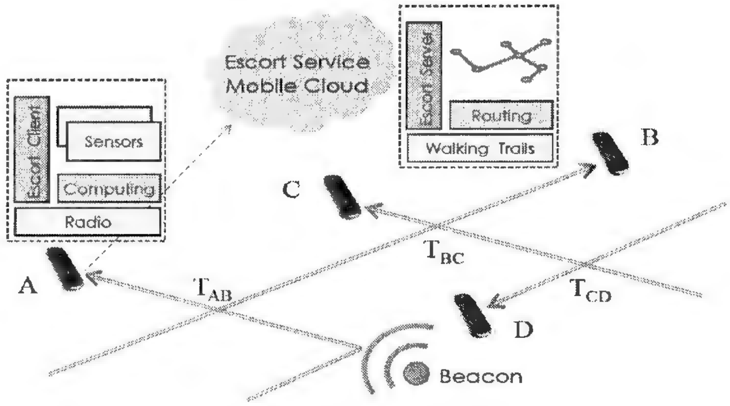


Figure 2.3: The overview of the Escort system: Users report accelerometer and compass readings as well as user encounters. The server forms user trails [16].

next doors. Here cafe and bookstore are two different logical locations. GPS is used to shortlist the number of possible logical locations and the accelerometer signature is captured for the user. It is then compared with the shortlisted possible locations. The best match is considered the location of the user.

Figure 2.4: Dividing wall problem [34]

SurroundSense [34] exploits diversity of a place by sensing the unique ambience of the surroundings from sound, light, color, human movement and Wi-Fi signals to create a fingerprint. This fingerprint can be matched from the fingerprint database to identify the logical location. Such a solution is feasible but the database would require frequent war-sensing as the ambience of locations might change over time. War-sensing is similar to war-driving, where the sensed information from the environment, for example light intensity, noise, temperature, etc is collected from all logical locations. In SurroundSense, authors compare the results of Wi-Fi based localization and variants of their system. One which uses sound, accelerometer, light and color, a second which uses sound, accelerometer and Wi-Fi and the third which uses all the sensors combined to create an ambience fingerprint. SurroundSence achieves an accuracy of 87% in identifying the correct logical location amongst the possible locations in their tests.

2.3 Localization in Robotics

Another related area of research which is close to indoor smart phone positioning is indoor robot localization. For an autonomous robot to navigate through indoor environments, it must have the ability to detect the current environment (using allocentric sensors, e.g., ultrasonic, camera, or laser) and calculate its trajectory (using egocentric sensors, e.g., wheel encoders). One of the methods is to use probabilistic technique to generate a belief distribution based on its motion model using wheel encoders. These estimates are then improved (Measurement Model) by observing the environment and finding landmarks and matching them with pre-built maps. Based on the movement trajectory calculated by internal sensors, the robot can eliminate locations with low belief. As more and more low belief locations are filtered out, the robot can be localized at locations with high belief. The robot localization is a core part of autonomous robotics in which it is required to achieve centimetre-level accuracy and high precision level. However, this technology is complex and expensive both in computation and the implementation of positioning module [52].

Existing robot localization algorithms extract features from the robot's sensor measurements. Techniques used for measurement models, such as most model matching approaches, extract geometric features such as walls or obstacles from the sensor data, which are then matched to models of the robot's environment. Landmark-based approaches scan sensor readings for the presence of landmarks to infer a robot's position. This method has become very popular in recent years. The range of features used by different approaches to mobile robot localization vary and depend on what kinds of sensors are used. They range from artificial markers such as barcodes and RF transmitters to more natural objects such as ambience and doors to geometric features such as corners and straight wall segments.

Following is a simple example of mobile robot localization. *Bel(*ξ*)* expresses the robot's belief (uncertainty) that its current position is ξ, where ξ denotes the arbitrary position of the robot within a global reference frame. The term location is used to refer to the variable: the robot's x-coordinate. Internally a robot has a belief which is a probability distribution function of the robot's possible position, although physically a robot always has a unique location at any point in time.

Figure 2.5 provides a graphical example that illustrates the localization algorithm. Initially, the location of the robot is not known except for its direction. Thus, *Bel(*ξ*)* is uniformly distributed over all possible locations shown in Figure 2.5(a). From the sensors, the robot determines that it is next to a door. This information alone is

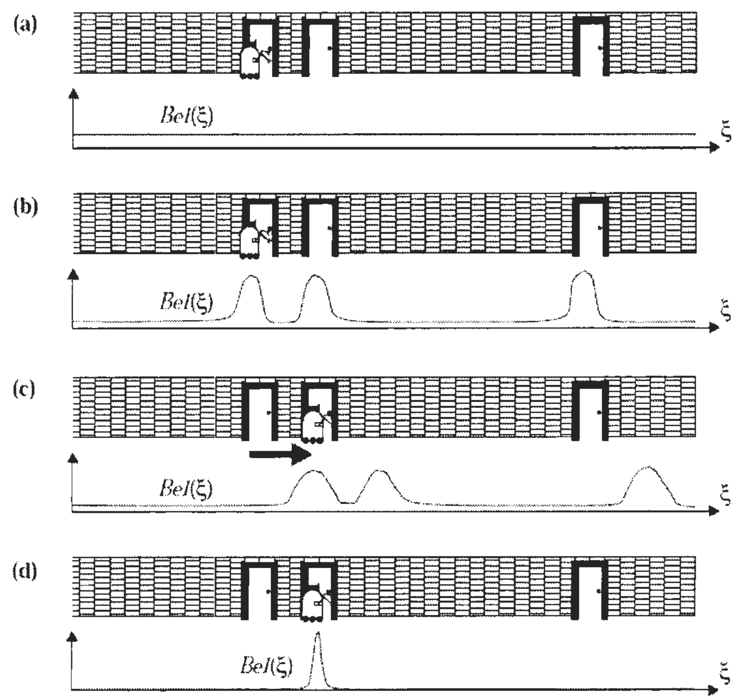


Figure 2.5: A Mobile Robot during Global Localization [51].

not enough to specify its position uniquely because of the presence of multiple doors in the environment and partially because the feature extractor might have an error. As a result, *Bel(ξ)* is higher for door locations and lower everywhere else, as shown in Figure 2.5(b). Next as the robot moves forward, the density *Bel(ξ)* is shifted in response to the robot motion as in Figure 2.5(c). Probability density is also slightly flattened out, reflecting the uncertainty introduced by movement. The robot now queries its sensors once more and finds out that again it is next to a door. The resulting belief, in Figure 2.5(d), now has a fairly accurate single peak which shows that the robot estimates with a high accuracy at where it is.

The central idea in any map-based robot positioning is to provide to the robot, directly or indirectly, a description of the landmarks expected to be found during navigation. Due to advancement in the field of computer vision, cameras are extensively used in mobile robot localization [63][49][46]. The vision system searches and identifies the landmarks observed in an image it acquires and digitizes. It detects landmarks, usually this means extracting edges, smoothing, filtering, and segmenting regions on the basis of differences in gray levels, color, depth, etc. Once they are identified, the robot can use the provided map to estimate its position (self-localization) by matching the observation (image) against the expectation (landmark description in the database). Landmark detection can be done in various ways. Some methods might require object recognition to detect landmarks and other simpler ones might just compare current images taken from camera to those stored in the database to estimate the position and orientation of the robot in the environment.

2.4 RF Based Positioning

There are several ways in which RF signals can be used for positioning. It is not easy to model the radio propagation in indoor environment because of diffraction, scattering, shading, severe multipath, low probability for availability of line-of-sight (LOS) paths, and specific site parameters such as floor layout, moving objects, and numerous reflecting surfaces. Different techniques have different advantages and disadvantages. Hence, using more than one type of positioning algorithms at the same time could yield better performance. Triangulation, scene analysis algorithms or proximity based algorithm are developed to minimize positioning errors.

2.4.1 Proximation

The most naive and simple way of localization would be to use proximity algorithms, as they provide symbolic relative location information. When there is a dense grid of base stations or antennas, each having a known location, it is easier to implement this method because of its simplicity. When a target device detects a single base station, it is considered that the device is collocated with that station/antenna. When more than one antenna detects the mobile target, then the one with the strongest signal is chosen to be the candidate where the target device is located. It can be implemented over various different types of physical medium. Infrared radiation (IR) based systems and radio frequency identification (RFID) systems are frequently based on this method.

The most prominent advantage of using infrared based solutions is its wide availability and the simplicity of the infrastructure. It does not need costly installation and maintenance as IR sensors are usually very cheap. However, due to the requirement of line-of-sight (LOS), it cannot be applied in complex indoor environment.

Another example is the cell of origin (COO) method or cell identification (Cell-ID). This method relies on the fact that mobile cellular networks can identify the approximate location of a mobile device by knowing which cell site the device is using at a given time. Cell-ID is already in use today and can be supported by every mobile device. The only problem with proximity based solutions is that it is assumed that the target is collocated with the access point (AP) in this case the Cell-ID. This can have hundreds of metres of error, which is not suitable for our applications.

2.4.2 Triangulation

Triangulation uses the geometric properties of triangles to estimate the target location. It has two derivations: lateration and angulation. The fundamental idea of triangulation is depicted in Figure 2.6. 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 methods described in following subsections. Once the relative distances d_1, d_2, and d_3 are calculated, the position of the target can be estimated using either the directions of the formed triangle or the intersection points of the circles. Most of the cellular based localization solutions adopt these techniques. The following subsection explains how we can get these distances from the transmitters.

2.4.2.1 Lateration

Lateration estimates the position of an object by measuring its distances from multiple reference points. Thus, it is also considered a range measurement technique. Figure 2.7 shows the distance d between mobile device and one such base station. There are several ways of calculating the distance d.

Received Signal Strength (RSS): In free space, the signal strength 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. In RSS based techniques, the distance is measured based on the attenuation introduced by the propagation of the signal from the transmitting node to the receiving node. A model used in [43][47] indicates that the mean path loss increases exponentially with

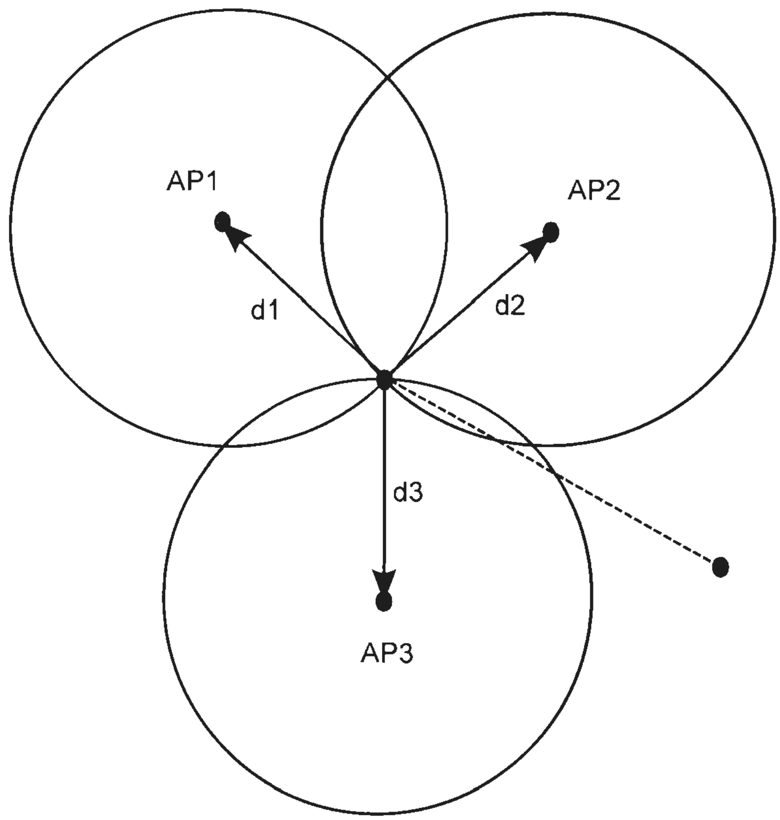


Figure 2.6: The distance between transmitter and receiver.

distance when not in free space and that the mean path loss is a function of distance to the n power.

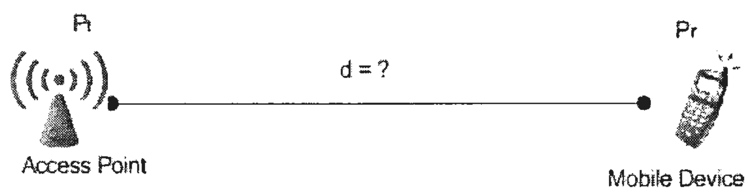


Figure 2.7: The distance between transmitter and receiver.

$$PL(d) \propto \left(\frac{d}{d_0}\right)^n \tag{2.1}$$

Here $PL(d)$ means the path loss, n is the mean path loss exponent which indicates how fast path loss increases with distance, d_0 is the reference distance, and d is the transmitter-receiver separation distance. The absolute mean path loss, in decibels, is defined as the path loss from the transmitter to the reference distance d_0 plus the additional path loss described by Eq. 2.1.

$$PL(d) = PL(d_0) + 10n_p \log_{10}\left(\frac{d}{d_0}\right) + \overline{X_\sigma} \tag{2.2}$$

The above equation estimates absolute path loss where $\overline{X_\sigma}$ is a zero mean log-normally distributed random variable. The n and σ parameters that are to be estimated empirically or theoretically. These are functions of the building types and would be unique for every building. Factors like floor/wall types, number of obstacles between the transmitter and receiver and floor level would affect these parameters. Using the above Eq. 2.2 and 2.3 d can be calculated as $PL(d)$ is calculated from Eq. 2.3 and put in Eq. 2.2, where d is the only unknown variable.

$$PL(d) = P_t - P_r \tag{2.3}$$

where P_t is the trasmission power and P_r is the receiving power.

Time-Based Methods: Instead of measuring the distance directly using received signal strengths (RSS), time of arrival (TOA) or time difference of arrival (TDOA) is usually measured, and the distance is derived by multiplying the radio signal velocity and the travel time. The distance from the transmitter to the receiving unit is directly proportional to the propagation time. TOA measurements must be made with respect to signals from at least three reference points in order to determine the position.

$$d = t \times s \tag{2.4}$$

Here s denotes the travelling speed of the signal, t the amount of time spent by the signal travelling from the transmitting to the receiving node, and d the distance between the receiving node and transmitting node. Since speed is a known constant, d can be computed by observing time.

The idea of TDOA is to determine the relative position of the mobile transmitter by examining the difference in time at which the signal arrives to multiple receivers, rather than the absolute arrival time of TOA. With two receivers at known locations, an emitter can be located onto a hyperboloid. A third receiver at a third location would provide a second TDOA measurement and hence locate the emitter on a second hyperboloid. The intersection of these two hyperboloids describes a curve on which the emitter lies. Now a fourth receiver will provide a third TDOA measurement. The intersection of the resulting third hyperboloid with the curve already found with the other three receivers defines a unique point in space. The emitter's location is therefore fully determined in 3-D.

In general using TOA has two problems. First, all transmitters and receivers in the system have to be precisely synchronized. Second, a timestamp must be labelled in the transmitting signal in order for the measuring unit to discern the distance the signal has traveled. TDOA does not have this problem as only time difference is required between the receivers.

Similarly the Return Turnover Time (RTT) method emerges with the goal of solving the problem of synchronization incurred by TOA. With RTT, the distance is calculated as follows:

$$d = \frac{(t_{RT} - \Delta t) \times s}{2} \tag{2.5}$$

t_{RT} denotes the amount of time needed for a signal to travel from one device to the other and back again, Δt the predetermined time delay required by the hardware device to operate at the receiving device, and s the speed of the transmitting signal. Time-based measurement methods are now in widespread use. However TOA based methods are limited by strict requirements of synchronization [64]. Received signal phase method and roundtrip time of flight are also used for range estimation in some systems.

2.4.2.2 Angulation

The main advantage of Angle Of Arrival (AOA) is that a 3-D position can be estimated with as few as three transmitters/receivers. For 2-D positioning only two measuring devices are required, and no time synchronization between measuring devices is required. The system employs either an array of antennas or directional antennas. Angulation basically estimates an object by computing angles relative to multiple reference points. The location of the mobile device can be found by the intersection of several pairs of angle direction lines from a base station [64].

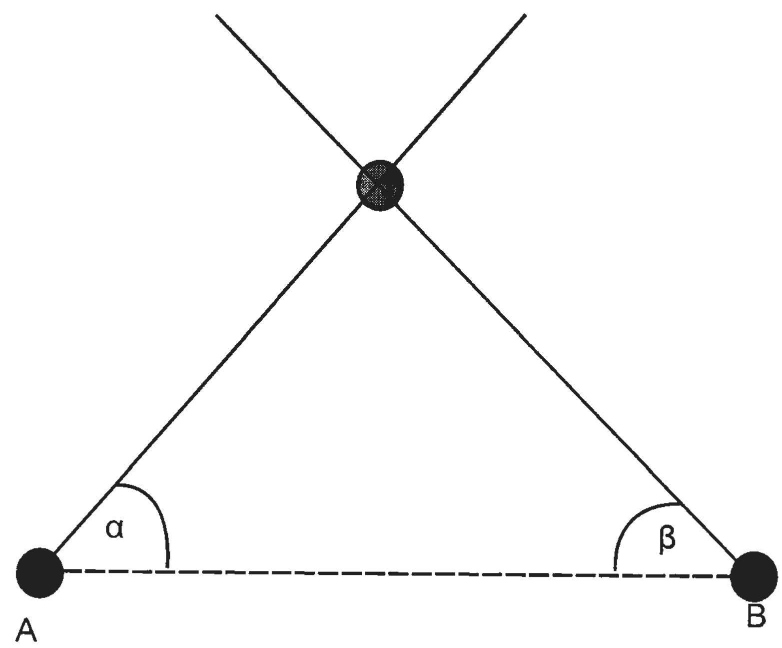


Figure 2.8: Using directional antennae to localize.

2.4.3 Fingerprinting

As an alternate to the propagation-model based localization solutions, there is Wi-Fi RSS fingerprinting technique. This technique can be generally divided into two phases: 1) an offline phase and 2) an online phase. The offline phase is called the training phase and the online phase is called the positioning phase. In the offline phase, a radio map is created by storing information about all the visible AP and their RSSI values for all locations of interest, which can be called reference points or anchor points. After collecting this raw data, for each location a fingerprint has to be created. The idea is that each location in the area of interest will have a unique vector of AP and RSSI values. It is very important that the anchor points are also chosen in such a way that they increase the accuracy and reliability of the system.

After the training phase each anchor point was associated with a Wi-Fi fingerprint, these fingerprints are then used by the positioning phase by comparing it to the current Wi-Fi measurements. The best match will yield the highest likelihood for correct location. Chapter 4 will explain in detail our process for collecting Wi-Fi data, creation of fingerprints and Wi-Fi positioning.

Algorithms used for comparison between Wi-Fi data collected in the positioning phase and the fingerprints in the database can be classified into two main categories - deterministic and probabilistic. In probabilistic techniques the device's position is modeled as a random vector. The candidate anchor γ is chosen if it has the highest probability. Usually the decision rule uses Bayes' theorem to calculate the likelihoods for all candidates.

On the other hand the deterministic framework is based on optimizing the similarity between observed online RSS measurement and the fingerprint such as using a scoring method. Various techniques are used to optimize the similarity. In the simplest case usually the Euclidian distance is calculated but other distance metrics are also possible. The case in which the closest fingerprint match is considered, is called nearest neighbour. If K anchors are considered then it is called K-Nearest Neighbour (KNN) and sometimes non-negative weights are used to compute the estimate which yields Weighted K-Nearest Neighbour (WKNN).

Although the basic idea of Wi-Fi fingerprinting is straight forward, there are still many challenges and areas where researchers are working to improve the fingerprinting techniques. Kushki et al. [26] discuss five main such challenges for Wi-Fi fingerprinting-based techniques:

- Collecting data from a large number of positions is difficult.
- Selection of APs in the positioning phase.
- Pre-processing fingerprints to increase accuracy is difficult as it is difficult to predetermine which AP's are important for positioning.
- Quantization of distance between the Wi-Fi RSS vectors in the signal space.
- Building analytical models to evaluate system performance.

To increase the accuracy of the positioning system, it is really important that the training is done in a proper way. Training process can be very laborious, especially for future updates and maintenance. In [32] the researchers have come up with a user feedback model for increasing the accuracy of the system. The user can give positive and negative feedback. Apart from the system anchor points, the user can also create new anchor points if the user is standing at a non surveyed position. Positive and negative feedback will increase the weight of the anchor points hence increasing the accuracy. A lot of work is being done in using Wi-Fi fingerprinting with focus being on maximizing accuracy and minimizing the calibration needed to achieve it. [23][26][29][30][36] all try to improve Wi-Fi fingerprinting approach for localization.

Chapter 3

Pedestrian Tracking and Position Estimation

3.1 Introducing the Smartphone Sensors

The MEMS technology and smartphone sensors market are growing rapidly. These MEMS sensors are fuelling the growth of new consumer electronics devices, which in turn helps the growth of the MEMS industry. Smartphones are getting smarter because of all the sensors being added to them. By using sensor fusion, one can take information from all of these sensors to categorize the environment the user is in. As an example, in a mall there are various types of stores next to each other. Each store will have its unique ambience. A cafe might have different type of lighting compared to a bookstore next doors. Even the light sensor maybe able to somehow differentiate between the two places. Microphone might be able to help distinguish amongst different places due to the background noise. Most of the latest smartphones are equipped with sensors such as proximity sensors, GPS receiver, Wi-Fi, magnetometers, light sensors, accelerometers and gyroscopes. The coming generation of mobile devices are set to have many new types of sensors like altimeter sensors that would be able to detect your elevation. Additionally, phones will include more microphones, temperature and humidity sensors to better determine their location and surroundings.

For our research purposes we needed those sensors to help us determine human motion and also estimate the position of the user in an environment. Apple currently is a market leader in smartphone technology with the iPhone capturing a major market share in the smartphone users. We chose iPhone 4 as our platform of choice to develop and test our system. The iPhone 4 comes with a bundle of sensors including magnetometer, Wi-Fi, accelerometer and gyroscope. The iPhone is programmed in Objective-C, which is quite a simple language to learn and use. Objective-C is a superset of the C language, with some object oriented-programming features. The IOS APIs and emulator (which runs on desktop/laptop MAC computers) make programmability, UI design, and code debugging an efficient process for developers. All frameworks are well designed and documented, abstracting the developer from low level components. In addition, the motion sensor APIs are cleanly designed and make accessing these devices simple and strightforward. In [37] the authors have done a performance evaluation of iPhone compared to another leading smartphone from Nokia. According to them the iPhone offers a rich UI architecture, high computational capability, and an efficient application distribution system through Apple's App Store compared to Nokia N95.

3.1.1 Accelerometer

An accelerometer is a device that can measure the force of acceleration, whether caused by gravity or by movement. The iPhone 4 uses the LIS331DLH 3-axis MEMS based accelerometer produced by STMicroelectroics. The magnitude and direction of acceleration can be measured, and used to sense the orientation of the device. An accelerometer can therefore measure the acceleration of an object it is attached to. Because an accelerometer senses movement and gravity, it can also sense the angle at which it is being held. This feature allows apps to automatically adjust the visual output to make it appropriate to the direction of the screen. Apart from the tilt, it can also detect vibration so different gestures like shaking can be detected and put to use for different applications. Figure 3.1 shows the axes of the 3D accelerometer as defined by Apple with respect to the iPhone.

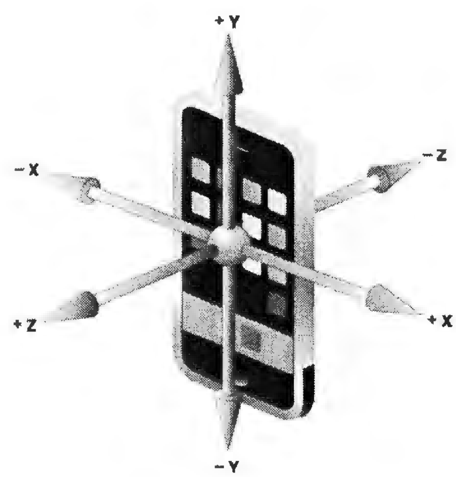


Figure 3.1: Axes of Accelerometer

3.1.2 Gyroscope

A gyroscope is a device for measuring orientation. A phenomenon called *gimbal lock* is one of the major problems with mechanical gyroscopes. It occurs when two of the three gimbal rings are aligned in the same plane due to rotation. This reduces the system's degree of freedom and the gimbal would no longer be able to rotate and maintain the orientation. In recent years, inexpensive gyroscopes manufactured with MEMS technology have become widely available. These sensors work in a similar fashion to the linear accelerometers as they provide instantaneous reading of the angular velocity. This value can be recorded and integrated over time to calculate the object's orientation. Figure 3.2 shows the axes of the 3D gyroscope.

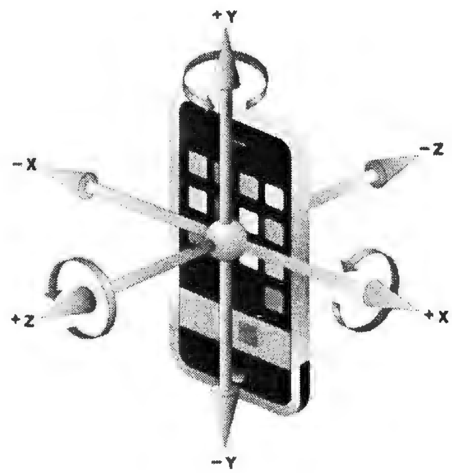


Figure 3.2: Axes of the Gyroscope

3.1.3 Magnetometer

The compass in the iPhone 4 is the AKM AK8975. The magnetometer is based on the Hall Effect, which is one of a number of methods for detecting magnetic fields. The IOS framework provides us with the raw x,y,z components of the sensed magnetic field vector in addition to the magnetic heading. Magnetic heading is a heading relative to the magnetic poles of the Earth which is different from true geodetic heading. The true heading is relative to the actual North and South Poles of the Earth. Calculating true heading requires the knowledge of the present position, hence satellite-based positioning is used to estimate the true heading. The magnetic heading also contains a two-part compass error: (1) magnetic variation due to the Earth's magnetic field and (2) magnetic deviation, which is the local magnetic fluctuations. This can be due to of metallic structures inside buildings or other electronic equipment.

3.2 Understanding Human Gait using Accelerometers

Gait is the pattern of movement of the limbs and human gait is a popular topic in Medicine and Kinesiology. A particular way or manner of moving on foot is the definition for gait. Every person has his or her own style of walking and factors like injuries, aging and operations on the feet might change a person's style of walk. The gait pattern is very important for medical diagnosis of ambulation and estimation of energy consumption. In [28] they use a 3-axis accelerometer on the waist belt to detect the acceleration of the body. They then process this information to estimate information about the subject as gait pattern, speed of the subject and total walking distance.

Gait recognition is a vast topic on its own. Biometric gait recognition has been studied for identity verification as surveillance and forensic systems are becoming important. There are three different approaches in gait recognition; *Machine Vision (MV)* Based, *Floor Sensor (FS)* based and *Wearable Sensor (WS)* based. In the MV technique, several cameras are used to capture gait images and then different algorithms can be used to determine the gait cycle. In the floor sensor approach the sensors are placed along the floor where gait data is measured when people walk across. The WS based gait approach is based on wearing motion sensors on the body of a person in different places like waist, pockets, foot or arms. The topic of accelerometer-based activity recognition is also not new. Bao and Intille [9] developed an activity recognition system to identify twenty activities using bi-axial accelerometers placed in five locations on the user's body. [21][27][24][44] are studies in the same domain where they try to classify human activity like standing, walking, jogging, running, climbing up stairs and climbing down stairs using various artificial intelligence and data mining techniques.

We assume that the user walks while holding the smartphone in hand and with the $+y$-axes pointing in the direction of walking and the -z-axes pointing downwards. Figure 3.3 shows a typical pattern of x-,y- and z- measurements corresponding to vertical, forward, and side acceleration of a walking person. This data represent 40 steps taken in a straight line at a constant pace. The raw data obtained is very similar to the data presented in [28][38][61], where the accelerometer was attached to the hip. The only difference is that when the accelerometer was fixed to the hip, the x-axes

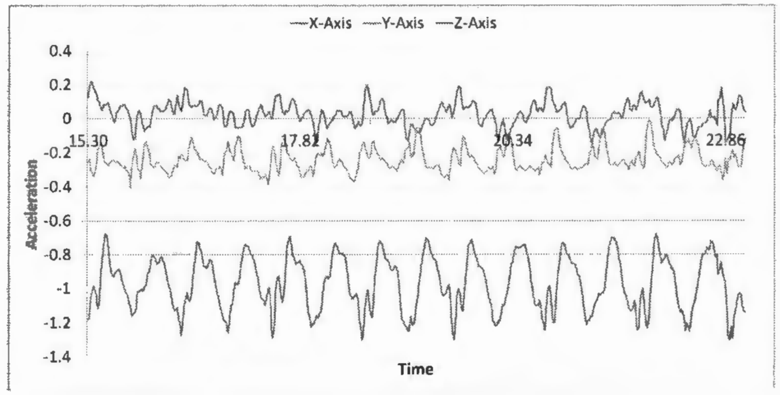


Figure 3.3: Raw accelerometer readings

readings were more stable than those presented here. The reason behind this is the fact that when the user walks, the arms sway with the motion giving higher x-axes readings. Figure 3.4 shows the different stages in the acceleration pattern.

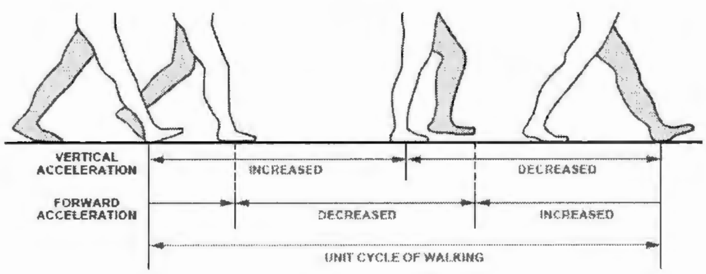


Figure 3.4: Walking stages and acceleration pattern when accelerometer attached to hip [65]

Human gait analysis shows us that we can use these vertical and forward accelerations to determine steps taken by the user. If we can estimate the steps taken by the user, it will be easier to develop a reliable motion model which can be used for dead reckoning. The following sections describe how to determine distance walked and how a reliable motion model can be used.

3.3 Pedometer Based Dead Reckoning

An inertial navigation system (INS) is a navigation aid that uses a computer, accelerometers and gyroscopes to continuously calculate via dead reckoning the position, orientation and heading of a moving object without any need for external references. Prior to satellite positioning systems, such as the United State's GPS or the EU's Galileo system, inertial navigation was relied upon to provide accurate position data for a number of vehicles, including guided missiles, aircraft, submarines, and spacecraft. The classical strapdown INS systems have lightweight computers along with inertial sensors simply attached to the body of the vehicle or object which calculate the attitude. Attitude is the orientation in space of the INS axes (body frame x,y,z) with respect to the reference frame. Figure 3.5 shows inertial reference frame which is not rotating with respect to the fixed global positions. Accelerometers and gyroscope are measuring acceleration and physical rotation in its own coordinate frame hence it is difficult to transform them to the global reference frame for localization. The axes of the Earth frame are fixed with respect to earth and usually parameterized with geographical coordinates: latitude, longitude and altitude. GPS uses the Earth frame of localization and navigation. We can define our own coordinate system too, for example by choosing a point as origin and then aligning the three axes orthogonally to each other. This can be called the local navigation frame.

The accuracy of such inertial navigation scheme is a function of the accuracy of sensor inputs and the frequency of data capture. To calculate the position, the

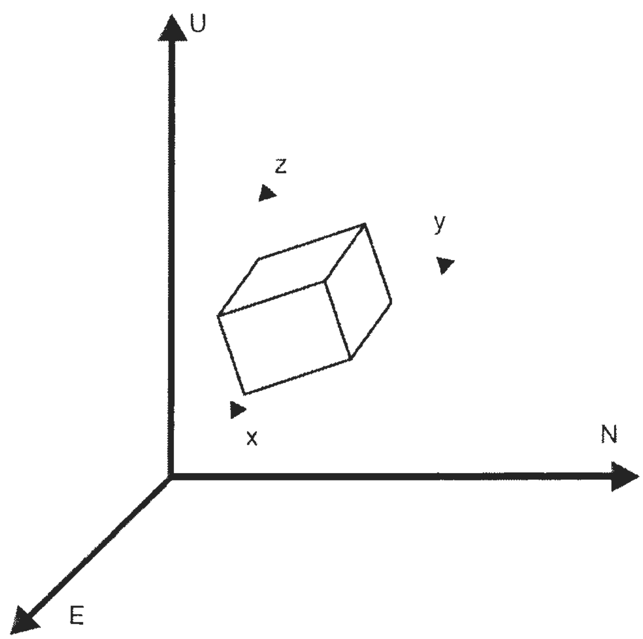


Figure 3.5: Body frame(x,y,z) and local navigation frame(E,N,U)

acceleration samples must be integrated twice to obtain position. These integrations can introduce errors in the data, known as integration drift. The problem stems from the fact that small errors in the acceleration measurements are integrated into larger errors as time progresses. Figure 3.6 shows an example of an INS. The error increases as time and travelled distance increases. Aircrafts use strap-down INS for positioning but they involve very high quality inertial sensors and also need high computation power because of the complex equations involved. This may not be possible in smartphones because of limited computing power and noisy IMU sensors.

The other method which seems to be more reliable is inspired by pedometers. Early designs of pedometers used a weighted mechanical switch to detect steps, plus

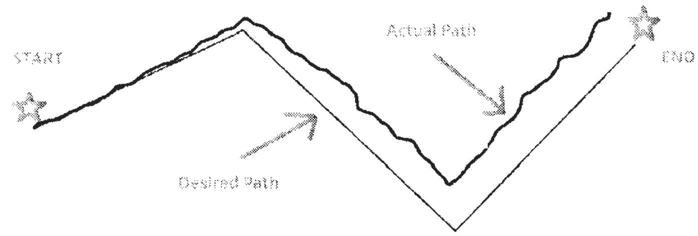


Figure 3.6: Integration Drift

a simple counter. When these devices were shaken, one could hear a metal ball sliding back and forth. The latest pedometers use accelerometers to detect steps. They then employ various methods of step length estimation. Almost all of them use height and weight of the user to state the length of the stride [11].

Step detection is the automatic determination of the moments at which footsteps occur. If accelerometer data is used to detect instant motion of the device, sudden changes in the movement have to be isolated from the constant effect of gravity. Figure 3.7 shows the magnitude of the accelerometer readings after passing through a high-pass filter. The user took forty steps in a straight line, this can be observed in the graph as forty peaks.

3.3.1 Distance Estimation Using a Step Counter

There are several algorithms available for step counters but most of them are primarily for accelerometers attached to the foot, hip or other body part. As we assume that the user will be holding the device in the hand, different algorithms were investigated.

Pan-Tompkins method is a real-time algorithm for detection of R peaks in electrocardiogram (ECG) signal. R peaks are usually the central and most visually

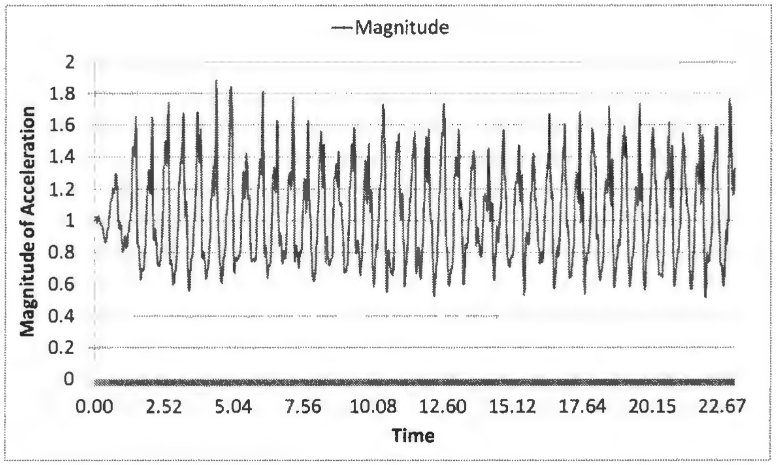


Figure 3.7: Magnitude of the accelerometer readings

Figure 3.8: Block Diagram of Pan-Tompkins algorithm

obvious part of the three graphical deflections seen on a typical electrocardiogram. Figure 3.8 shows the block diagram of the algorithm. In [61] this algorithm is applied to a foot mounted accelerometer. The Pan-Tompkins method is applied to a block of accelerometer readings. In their experiment they first pass the raw accelerometer signal through a low pass filter to reduce the influence of artefacts in the signal. The cut off frequency they chose was 20Hz. The derivative of the filtered signal is then taken to suppress the high-frequency components and enlarge the low frequency components. Then they do the squaring operation which enhances the larger values more than the smaller values. Due to the squaring and derivative operations multiple peaks arise. They are smoothed through a moving-window integration filter. In the final stage a peak-searching algorithm is applied to count the number of steps taken. Peak detection is a method which calculates the steps from the 3-axes accelerometer readings. A threshold value can be used to detect a peak. If the changes in acceleration are too small, the step counter will discard them. The step counter can work well by using this algorithm, but sometimes it seems too sensitive. When the device shakes or vibrates randomly from a cause other than walking, the step counter will also take it as a step. However, in [61] the authors used a different approach to finding the maximum. They called the points where step is detected as **fiducial marks**. From the pre-processed signal the negative slopes are transformed to -1 and positive slopes are transformed to +1. This way the step cycle is converted into pairs of $[-1, 1]$. This pair is referred to as the peak-searching interval. The local maximum is marked as fiducial mark and hence detected as step.

We implemented the same algorithm on a smartphone to see if we can get the same result. We made a small modification to the peak-searching phase as we used static threshold instead of the fidicial mark method described above. Figure 3.9 shows the graphs at different stages of the algorithm. The performance of this algorithm is not reliable for a stepcounter, as the error was always more than 60%. One of the reasons is that continuous motion is observed in the accelerometer readings when the device is held in the hands. In [61] the results are better as the accelerometer was attached to one of the feet. When the step is taken by the foot on which the accelerometer is not attached, lower magnitude accelerometer readings are observed which are smoothed out by the lower pass filter. Hence when the accelerometer attached foot's heal touches the ground, there is a spike in the accelerometer signal. From Figure 3.9 we can observe that after the filtering stage, derivative, squaring and moving-window integration does not help in detecting correct step count. In our experiment the sampling frequency was 60Hz. The data was passed through a Butterworth low-pass filter with cut-off frequency of 10Hz. The following equations were used for derivative operator and integration. Only the last four terms were used for integration because we want to capture the spike.

$$y(n) = \frac{1}{8}[2x(n) + x(n-1) - x(n-3) - 2x(n-4)] \tag{3.1}$$

$$z(n) = \frac{1}{N}[x(n-(N-1) + x(n-(N-2) + ... + x(n)] \tag{3.2}$$

where N is chosen empirically as 10.

In [38], the magnitude of the 3D accelerometer readings is taken. In the second step the signal is passed through the Butterworth low-pass filter with order 20 with cut-off frequency of 5Hz. In the final stage a hill detection and threshold calculation is done. Hill detection is similar to peak-searching in the previous step. In this case the threshold is chosen adaptively. In the implementation, the buffer length of accelerometer readings is chosen to be 100 samples. The threshold is selected after iterating over all the readings and then the number of hills are detected which count as number of steps in that block of accelerometer readings. After detecting the number of steps, the mean of peaks is calculated. The threshold is then selected as a factor of this peak mean. The result of this algorithm was pretty accurate and also dependent on the buffer size. With 30Hz of sampling frequency if the buffer length was more then 60 samples it would give an accuracy of more then 80%. The Table 3.1 shows the comparison of accuracy between the three implementations. However, there is one

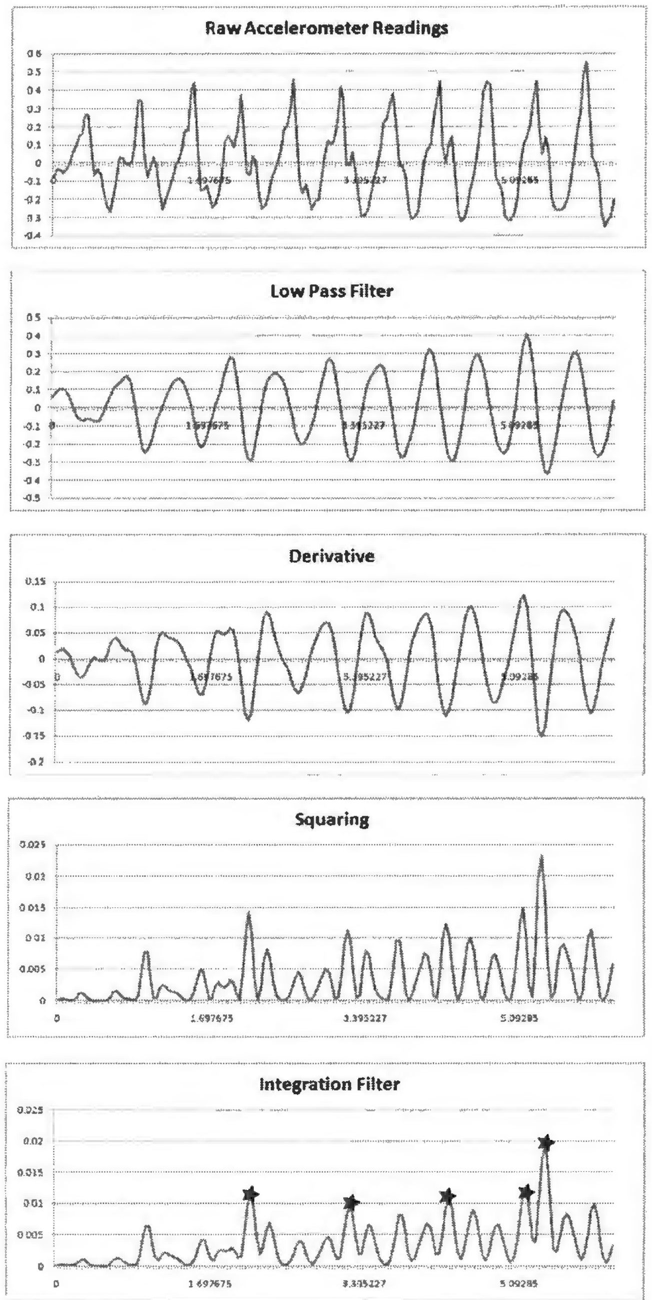


Figure 3.9: Results of the Pan-Tomkins method on the magnitude acceleration for 6 seconds. The data was collect by walking in a straight path. Stars are showing the step detection.

disadvantage of this algorithm. As we are implementing it over a block of readings, the step counting is not real time. For example if the sampling rate is 30Hz and we keep the buffer as 60 samples. The number of steps will be updated after 2s. Here is the Hill detection and threshold selection algorithm.

```
// Pseudocode for Hill Detection
//input:a[n] is the buffer which contains past n accelerometer readings.
//output:stepCount


numberOfpeaksCount = 0
peakAccumulate = 0
for all a[k] in the buffer do
            forwardSlope = a[k+1] - a[k]
            backwardSlope = a[k] - a[k-1]
            if forwardSlope < 0 AND backwardSlope > 0 then
                            numberOfpeaksCount = peakCount + 1
                            peakAccumulate = peakAccumulate + a[k]
            end if
end for
peakMean = peakAccumulate/numberOfpeaksCount


stepCount = 0
for all a[k] in the buffer do
            forwardSlope = a[k+1] - a[k]
            backwardSlope = a[k] - a[k-1]
            if forwardSlope < 0 AND backwardSlope > 0
```

```
                AND a[k] > C * peakMean
                                stepCount = numberOfpeaksCount + 1
                end if
end for
```

The algorithm that we chose for our stepcounter is inspired by an analog pedometer [65]. We used the Butterworth low-pass filter to remove the high frequency noise similar to the first step of the hill detection algorithm. The peak detection algorithm is used to detect the steps in the accelerometer readings. The threshold is empirically chosen as 0.14g, but using a static threshold may detect false steps as sudden movement of the hand held device may produce such measurements. Invalid peaks in the peak detection method must be discarded in order to find the true rhythmic steps. In our experiments we have assumed that people walk with speed between three steps per second to one step every two seconds [65]. Therefore the interval between two valid steps is defined as being in the time window $[0.33, 2.0]$. This time window is used to discard invalid vibrations. For example when a step is detected, no other step can be detected for another 0.33 seconds. When a new step is detected between 0.33 and 2 second the interval window moves and resets.

To make sure that steps are rhythmic in nature, the algorithm searches for 3 consecutive step detections in successive time windows. If this happens then the algorithm recognizes that the user is walking. The algorithm goes into a walk mode when this rhythmic pattern is recognized. Once in the walk mode, if the count manager realizes that the maximum window time has passed without step detection, it will go back to the stand mode until it detects 3 consecutive steps again. This algorithm was implemented on our iPhone and the result is shown is Table 3.1. In the experiments 500 steps were taken and the iPhone was held in the hand. The experiment was repeated three times by two different users. This algorithm exhibited similar accuracy to the hill detection algorithm but with the added benefit that it would update the step in real time whereas the hill detection outputs the total number of steps taken in a time window every fixed time interval. The accuracy of the stepcounter varies a little especially as when the mobile device is held in the hand, the sway of arms play an important factor in detecting steps. Different users may have different accuracies, but this can be fixed by increasing or decreasing the sensitivity *threshold*. We chose this algorithm for our stepcounter to be used in our localization scheme.

The template matching method [61] is also one of the ideas explored in the literature. The main concept of the template-matching method is to generate a template, which represents a typical step cycle. In the unknown signal, an event is declared to be detected when there is a match between the signal and the template to a certain degree. This method was not implemented as template matching is more computationally expensive.

After detecting the number of steps, the distance walked can be calculated by multiplying stepcount by step length. Step length is the distance from the heel print of one foot to the heel print of the other foot. This is the distance traveled forward in one stride. This can be approximated by the height of the user [11].

$$D_t = S_c \times l \tag{3.3}$$

where D_t is the total distance walked, S_c is the step count and l being step legth.

Table 3.1: Results on different implementation of step counters

Algorithm	Measured Step Count	Mean Error in Steps (percent)
Pan-Tompkins	810	−62%
Hill Detection	455	9%
Our algorithm	442	11.6%

3.3.2 Heading Estimation

Once the step is detected, it is important to know which direction the step was taken in. Smartphone magnetometers are very noisy, especially in indoor environments. Figure 3.12 shows a map of our department where we tested by walking in three corridors, changing directions two times. First the user is walking in a straight line and then turns right and walks straight till next corner of the corridor. The user turns right again and continue walking straight. The iPhone has a 3-axes gyroscope which can measure angular velocities about the axes. The motion framework of the IOS SDK also provide us access to built in functions which manage and keep track of the device attitude after the application starts. Rotation around z-axes is called yaw and at the start of the application it is calibrated with the initial stable magnetic heading. The result of magnetic heading is compared to yaw in Figure 3.10. It clearly shows that the gyroscope is more stable in an indoor environment. The only problem is that a gyroscope only maintains the local orientation of the device and hence it needs some kind of transformation to the global reference frame. A magnetometer on the other hand provides us with a global heading.

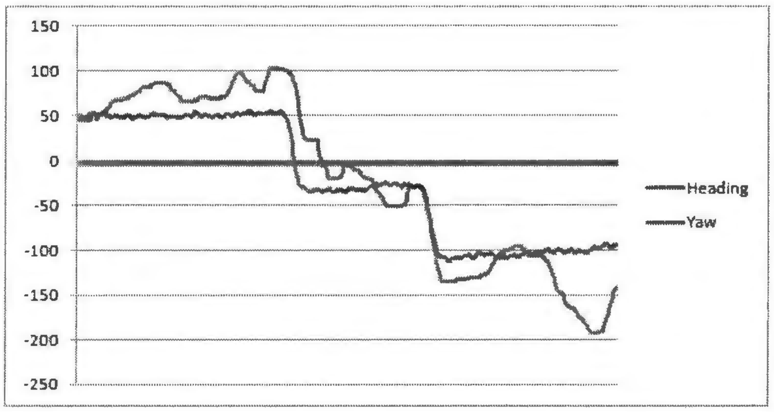


Figure 3.10: Heading and Yaw readings collected after walking in the corridors.

As stated in Chapter 2 that in an indoor environments due to magnetic interferences these headings have high errors. But in our preliminary study of using the magnetometer in an indoor environment, we found that although walking in a straight line in a corridor might fluctuate the heading $\pm 45°$, it will be enough to differentiate between different corridors in an indoor environment most of the time. Maps of an indoor environment have a small number of orientations of the corridors. For example Figure 3.12 shows a map with only four possible orientations as the corridors are at $90°$. Thus, the user can either walk in only $0°$, $90°$, $180°$ or $270°$ with respect to the coordinate system of the map. This relies on the assumption that the corridors will differ in orientation by an amount larger than the magnetometer error. During the war sensing phase explained in the next chapter, we will collect heading information of the possible orientations in a map. In this case a small table of four entries ⟨Magnetic Heading, User Orientation⟩ would suffice. For every different environment map, this information can be calculated for the possible orientations during the war sensing phase. There certainly will be areas where the local magnetic disturbances is larger than the error tolerance for that particular orientation. Hence in this situation, wrong orientation will be detected.

When the application is started, the first phase would be the initialization phase in which we calibrate the gyroscope to the map coordinate system using the magnetometer. During this phase the user can be asked to walk a few steps. If the user cheats by walking in a direction which is not paralled to the corridor, for example between the walls of a corridor or in a circle the calibration would be faulty and affect the motion model. This is a reasonable calibration process as it would allow to check for stable magnetic heading readings. During the calibration magnetic readings are recorded and voting is done to choose the initial user orientation in the map environment from the orientation selection table mentioned above. The voting process is employed instead of averaging because when the magnetometer is initialized the user might be standing in a high magnetic anomaly point and hence the wrong orientation can be selected. This orientation is then used for step direction. Chapter 6 will show the results of our calibration process.

3.4 Motion Model

In probabilistic robotics there is another key concept that of a *belief*. A belief is the internal knowledge of the robot or a system about the state of the world. In our case the state means the location of the subject in our environment. State cannot be measured directly but can be inferred from its internal belief. In probabilistic robotics

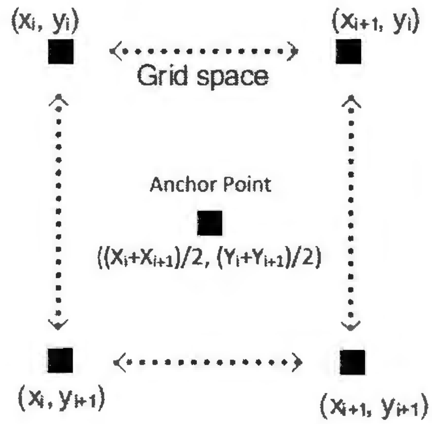


Figure 3.11: Anchor point and Grid space

beliefs are represented as a conditional probability distribution. This distribution assigns a probability to each possible hypothesis with regards to the true state. The state x_t is generated stochastically from the state x_{t-1}, meaning that the belief at time t is calculated from its past belief at time $t-1$. The most general algorithm for calculating beliefs is given by *Bayes filter.* Algorithm 1 depicts the Bayes filter. This algorithm is recursively applied every iteration when belief $bel(x_t)$ needs to be calculated from $bel(x_{t-1})$ and the current control input sensory. The Bayes filter algorithm possesses two essential steps. In Line 2, it processes the control u_t. It does so by calculating a belief over the state x_t based on the prior belief over state x_{t-1} and the control u_t. The control u_t carry information about change of state in the environment, which in our case is the motion captured from the step counter. This step of the algorithm is also called *prediction* [52].

The second step of Bayes filter is called the measurement update. In line 3, the Bayes filter algorithm multiplies the belief $\overline{bel}(x_t)$ by the probability that measurement z_t may have been observed. It does so for each hypothetical posterior state x_t. To compute the posterior belief recursively, the algorithm requires an initial belief $bel(x_0)$ at time $t = 0$. If we are ignorant about the initial condition we can initialize using the uniform distribution.

Algorithm 1: *The general algorithm for Bayes filtering*

Input: u_t, z_t, $bel(x_{t-1})$

1: **for all** x_t **do**

2: $\quad \overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$

3: $\quad bel(x_t) = \eta p(z_t|x_t) \overline{bel}(x_t)$

4: **end for**

Output: $bel(x_t)$

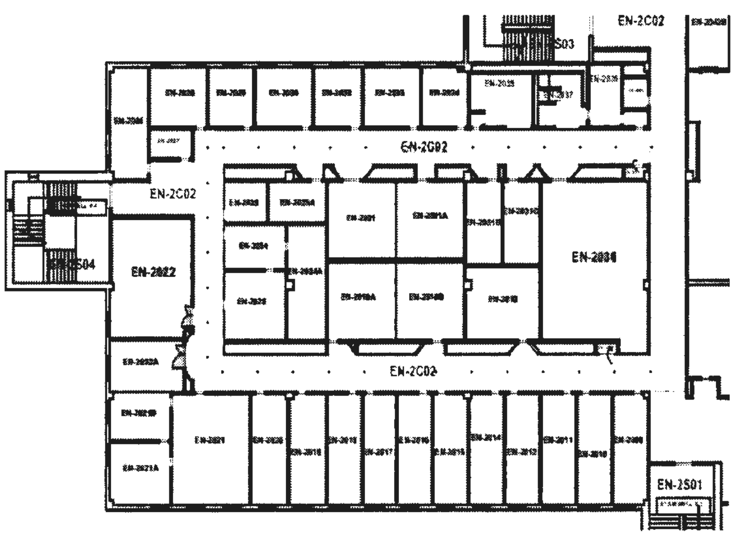


Figure 3.12: Map of the Engineering Building

To study our motion model we divided our map into grid spaces. The center of these grid space has the anchor points which have known physical coordinates (x, y). The grid space between two anchor positions determines the resolution or granularity of the positioning system (Figure 3.11). A number of issues arise when implementing grid localization. For a coarse grid, additional information is lost in the discretization process which affects the filter negatively whereas with a fine grid, the computation cost increases.

The x_t describes a list of anchor points and is the hypothesis that the subject is in one of those positions. Figure 3.12 shows the map of the second floor of SJ Carew (Engineering building) and the positions of all the anchor points. These anchor points are 6m apart. Algorithm 2 shows our motion model which uses relative motion information as measured by the stepcounter and gyroscope.

In the time interval $[t-1, t]$ the user advances from position x_{t-1} to position x_t. The step counter and gyroscope report back the relative change in position (x_{rel}, y_{rel}). As we know the initial heading and current heading of the user, we can determine the user's direction of travel. So from the last position and the new position we can determine x_{rel} and y_{rel} which are distances travelled in the x-direction and the $y-$direction with respect to our map.

$$x_{rel} = \alpha \cos(\theta + \beta) \tag{3.4}$$

$$y_{rel} = \alpha \sin(\theta + \beta) \tag{3.5}$$

where θ is the initial orientation of the device during initialization, β is the yaw of the device and α is the step length.

The corresponding relative motion parameters (x^*, y^*) for the given poses x_{t-1} and

Algorithm 2: *Motion model for computing* $p(x_t|u_t, x_{t-1})$ *based on motion captured from step counter. Here the control* u_t *is given by* (x_{rel}, y_{rel}), *with* $x_t = (x, y)$ *and* $x_{t-1} = (x', y')$. x_{rel} *and* y_{rel} *are the relative distance travelled in x-direction and y-direction in map coordinates. They are calculated using steps taken and step direction.*

Input: u_t, x_t, x_{t-1};

$x* = x' - x$;

$y* = y' - y$;

$\delta_x = x_{rel} - x*$;

$\delta_y = y_{rel} - y*$;

$p1 = norm(\delta_x, \sigma)$;

$p2 = norm(\delta_y, \sigma)$;

$result = p1 * p2$

Output: result

x_t are calculated in Lines 1 and 2. These basically come from the known positions in the map. The function $norm(a, b)$ implements an error distribution over a with zero mean and standard deviation of b which was empirically chosen as 4m. The motion model is used as step 2 in our Bayes filter implementation.

Chapter 4

Wi-Fi Positioning

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.

4.1 The Concept

Indoor positioning is challenging because of the non-line-of-sight transmission between receivers and transmitters. Walls, ceiling, equipment and humans obstruct the propagating electromagnetic waves. As discussed in Chapter 2, there are various Wi-Fi based schemes used for indoor localization. Among them the location fingerprinting techniques use existing in-building communication infrastructure to provide low-cost and accurate localization. The fingerprinting technique is relatively simple to deploy compared to other techniques like triangulation. In Wi-Fi triangulation, the goal is to map the RSSI (Received Signal Strength Indication) as a function of distance and use live RSSI readings to generate a (x, y) location using a model. It is very difficult to make a model which satisfies every indoor environment, hence making it less reliable and robust.

The basic idea of fingerprint based positioning system is as follows. Suppose there is survey position $\mathcal{P}_a$ where a mobile device can receive beacon frames from the i-th AP, $i \in \{1, 2, 3, \ldots, N\}$. These beacon frames are a type of management frame defined in IEEE 802.11 standard. These beacon frames are transmitted periodically and they announce all the information related to the network. The information in these frames are also used for managing and controlling the wireless link. The MAC address M_i, RSSI p_i and timestamp t_i can be extracted from each beacon frame. The characteristics of RSSI can be observed in Figure 4.1 as it shows the RSSI from an AP collected at different locations (anchor points) during a survey. As stated before, the signal attenuation is different and unique for every indoor environment and hence it is difficult to model.

If at each anchor point located by position (x, y), multiple APs are visible, the combinations of such RSSI values can be used to create a fingerprint for this location. To achieve this, we use a two-stage approach. In the first stage, which we call the training phase, a radio map is created for the *Location of Interest (LOI)*. Figure 4.2 shows the RSSI vectors which can be extracted from all the access points. After collecting and storing these raw data of every location, fingerprints can be generated. Each Wi-Fi fingerprint is the pattern of signal strengths of a collection of Wi-Fi access points visible in a particular area and incorporates e.g., the set of receivable APs, the average RSSI or the number of times an AP is visible. During the second stage (positioning phase), the mobile device scans for visible APs and creates a fingerprint

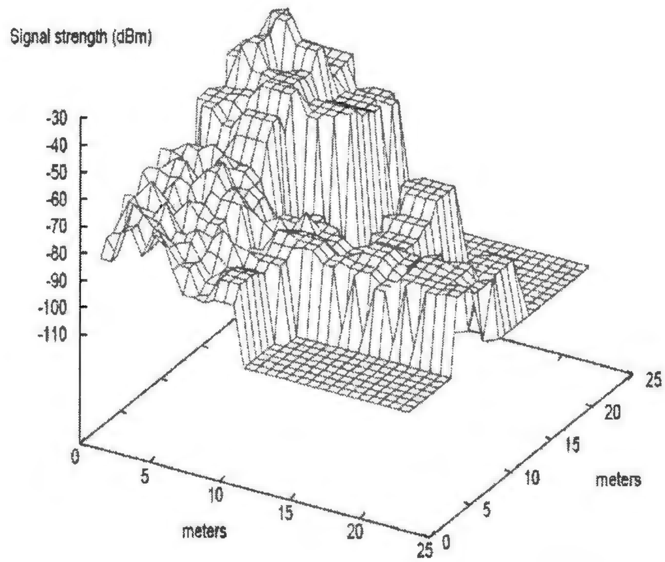
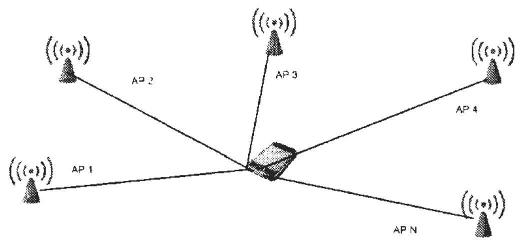


Figure 4.1: RSS readings from an AP at various survey points [10].

at the yet unknown position of the user. The positioning module then compares this fingerprint to all the fingerprints stored in the database and selects according to a system-specific similarity measure for the best matching counterpart. The location of the best matching counterpoint can be returned to the user at the position estimate.

4.2 Training Phase

In the training phase, a set of reference points in the study area are selected as survey positions with known physical coordinates. The training is conducted for each reference point.



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

Figure 4.2: RSS Vector.

4.2.1 Wi-Fi Warwalking

In the first stage of the training, all the reference points on the map with known physical coordinates (x, y) are tagged with location IDs. All the anchor points in Figure 6.1 and 6.2 are chosen as the survey points for our experiments. Smaller distance between these anchor points might increase the accuracy of the system, but it does not necessarily mean that the precision will also improve as different anchor points might have similar Wi-Fi fingerprints. Also, when we choose the anchor points closer to each other, it makes the training phase more laborious. There is no standard guideline for what the size of the grid should be. In our implementation we kept the grid size to be 3m for the Engineering Building and 5.5m for the university tunnel environment, considering our integration with motion model and the size of the experimental area.

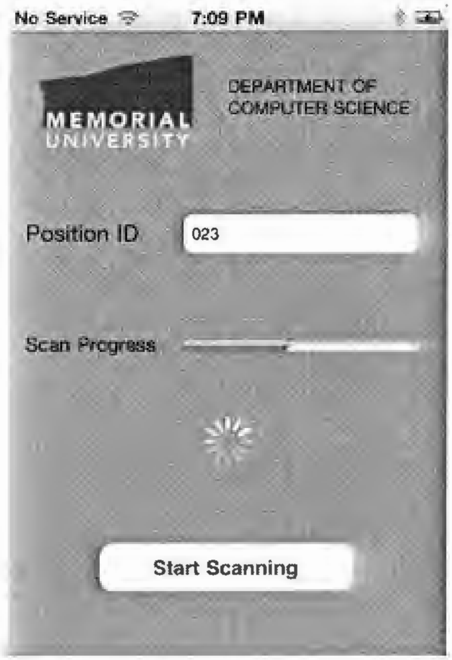


Figure 4.3: Wi-Fi Warwalking Utility.

To collect raw data, we implemented a small utility to scan Wi-Fi APs and save the data to our database. Figure 4.3 shows screenshot of our warwalking tool which takes position ID and then scans for the Wi-Fi APs. *Warwalking* is done at all the anchor positions. Warwalking is analogous to wardriving which is the act of collecting data on the move in a vehicle. Wardriving is a common practice among telecom and cellular companies as they collect data for expansion and optimization of their network. In indoor environment instead of using a vehicle, we have to walk to collect the Wi-Fi data.

When AP's beacon is processed by NIC a great deal of information is learned about the particular AP. Each AP has a unique MAC address. Along with the MAC address, the RSSI is also recorded. In each Wi-Fi scan, beacon frames from different APs are received and converted into a 2-tuple vector (i.e., MAC address and RSSI). A single scan may not be able to capture beacon frames from all the APs nearby. This maybe because of the different broadcasting periods of the APs or severe signal fading. To avoid missing out any nearby AP which can later be found in an online scan, several scans are taken at the same position. Figure 4.4 shows raw data collected for one such survey point. It shows that 41 scans were taken and the first scan showed 9 APs. Each AP's MAC address and RSSI value is stored. There are no general guidelines on the number of scans needed for data collection but in our study and experimental area about 20 scans were enough to show all the visible APs as more scans did not reveal any new visible AP.

Figure 4.4: Format of stored raw Wi-Fi data showing one scan.

4.2.2 Wi-Fi Fingerprint

After collecting the raw Wi-Fi data, in the second stage Wi-Fi fingerprints are generated. Statistics are extracted from the raw data to generate an RSSI fingerprint of each survey/reference point. A Wi-Fi *fingerprint* is defined as a 3-tuple (i.e., MAC, Average RSSI, Count) vector containing a set of APs. We can also store timestamp or RSSI variance which can also be used as a feature for the fingerprint because research has shown that fluctuation and variance of RSSI of a particular AP also varies during the day. However in our implementation we have not considered these features, but they may be addressed in our future work. Although these features could increase the accuracy and robustness of the system however our research will focus on the improvements mobility introduces to fingerprint-based positioning.

As described before 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 within range. The average RSSI $\overline{p}_i$ is an average of the Wi-Fi RSSI over the sampling period. During data collection several scans are taken at the same location. Each Wi-Fi scan contains the instantaneous RSSI values from each AP. As the RSSI values are fluctuating, it is necessary to take the mean value. The number of occurrences of the AP during the sampling period, denoted C_i, is also part of the fingerprint. For a fixed number of Wi-Fi scans, a large C_i 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 RSSI value, which is a very important indicator for the reliability of an AP. Figure 4.5 shows the RSSI fingerprint vectors. After the generation of fingerprints, each survey point $\mathcal{P}_s$ is associated with its fingerprint F_s.

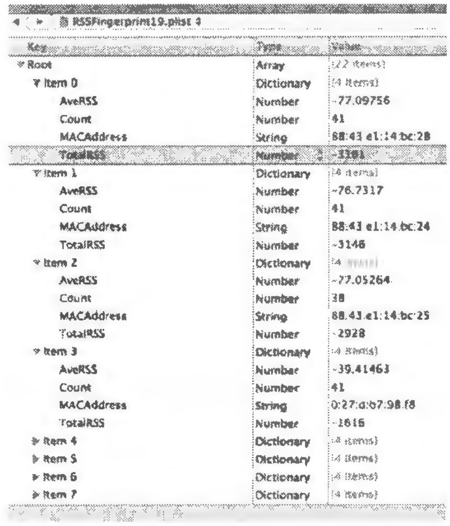


Figure 4.5: Fingerprint of an anchor point

4.3 Position Estimation

In the positioning phase, live Wi-Fi measurements are done and the system then queries the fingerprint database for a match. One Wi-Fi scan during the positioning phase may generate a poor match as the scan may lack enough RSSI data. According to Luo et al [32] the positioning error is greater if the number of scans is less than 4, but with 4 or more scans the positioning accuracy stabilizes. Their experiments were conducted in the same test environment as ours. For our experimental purposes in the positioning phase the positioning module scans 4 times. The combined vector contains the set of APs visible during this active scanning period. The next step is to calculate the most likely position estimate by matching this vector to all the fingerprints in the system.

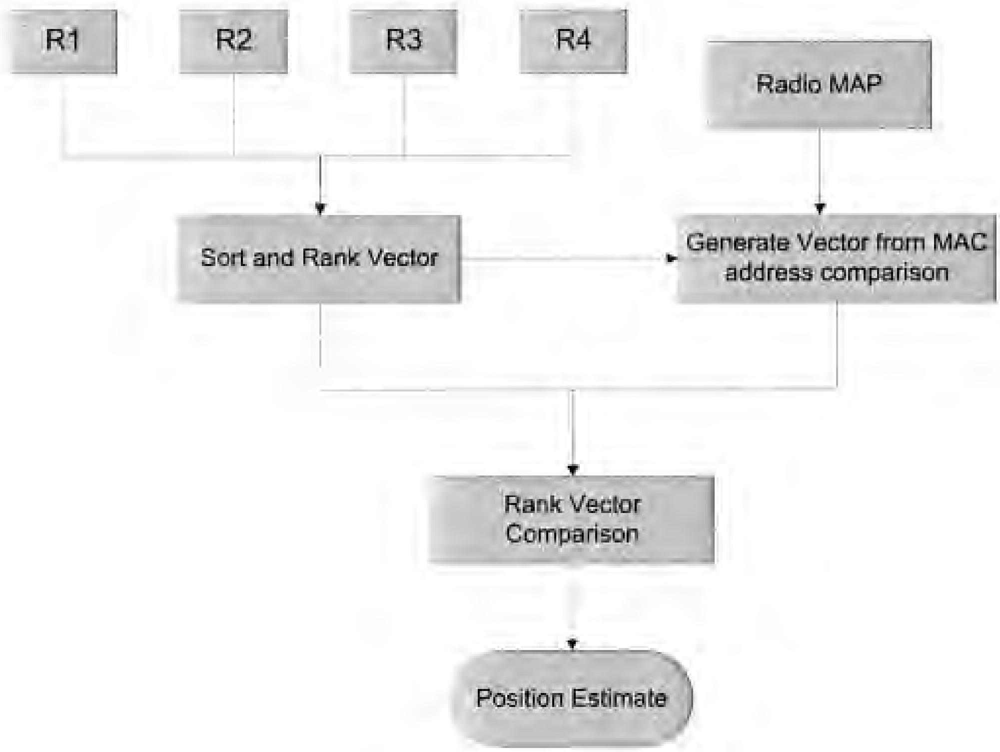


Figure 4.6: Block diagram for rank based fingerprinting algorithm

4.3.1 Rank Based Fingerprinting algorithm

In classical fingerprinting algorithms, vectors of RSSI measured in querying phase and training phase are directly compared to each other. Querying phase is the first part of the positioning face when the positioning module scans for live RSSI readings. The nearest neighbour's method simply calculates the Euclidean distance in the signal space between the live RSSI reading and the fingerprints. A major drawback of using this technique is that different devices, because of their hardware and software (sometimes devices of the same make and model), report different RSSI values which may differ from the RSSI stored in the database. This will degrade the performance of the positioning system. On the other hand, rank based localization [33] uses only ranks of the RSSI values because the rank information is less sensitive to small signal variation. Therefore, the performance should be unaffected by the calibration of the mobile device.

Figure 4.6 shows the block diagram of the rank based fingerprinting algorithm. In this algorithm first the RSSI values measured in the querying phase from different APs are first sorted from strongest to weakest. Ranks $(1, 2, 3, ...)$ are assigned to APs based on the position in the sorted vector. Rank 1 is given to the strongest AP, meaning with the strongest RSSI value. Similarly, rank vectors are created from the fingerprints stored in the database. Ranks are assigned based on the MAC address and rank of AP in the querying phase. Then this vector is also sorted strongest to weakest keeping the rank assigned to them. In ideal cases the sorted ranked vector from querying phase and sorted ranked vector fron training phase will be identical hence showing perfect similarity.

In case an AP which was in the querying phase was not found in the database, the rank vector created from the database is padded with 0, to achieve the same length as the rank vector from the query. Other techniques including the application of a Gaussian kernel [26], which calculates the likelihood of an anchor point using the RSSI value similarity between two vectors, also face the dimension mismatch problem. In real indoor environments the dimension of the fingerprints of different anchor points vary considerably. If simple likelihood calculation mechanism (e.g., Euclidean distance or cosine similarity) are used, mismatching could lead to large positioning errors.

4.3.1.1 Calculating Similarity

Spearman's footrule distance measures the total elementwise displacement between two vectors. It is similar to the Manhattan distance for quantitative variables. According to [32] Spearman's footrule performs the best amongst other similarity measures. Assuming u_k is the rank of the k-th element in vector U, v_k is the rank of the k-th element in vector V and n is the number of elements in vectors U and V then spearman's footrule distance can be computed as follows:

$$D_s = \sum_{k=1}^{n} |u_k - v_k|$$

4.3.1.2 Assigning Weight to Best Matches

The similarity measure mentioned above return scores for every anchor point. The anchor point with the lowest score is considered the best match. Ideally using k smallest reference points to calculate the estimated position yields a better result. In [32] the authors use the p-center algorithm to estimate the final position estimate. In the rank based technique the distribution of scores will differ for several reasons. The number of APs visible in the querying scan and position where the scan was done affects the distribution of the scores. For instance if the scan is done at a corner where 20 APs are visible compared to another location where only 5 APs are visible, the distribution of scores will differ a lot. A random test in the engineering building was done by selecting 13 anchor points. It was noted that the accuracy of the position estimate appears to be independent of the score distribution. For each anchor points we have a list of scores after comparing with all the fingerprints. Figure 4.7 shows the maximum and minimum score distribution.

Another important aspect to study is evaluating the certainity in our belief about the user position. As the user initiates the application, the belief is uniformaly distributed. Entropy is a measure of the uncertainty associated with a random variable and is also referred to as the expected value of the information contained in a message, which in our case is the belief. Entropy is decribed by the following equation.

$$H(X) = -\sum_{i=1}^{n}(p(x_i)\log_b p(x_i)) \tag{4.1}$$

where $p(x_i)$, is the probability mass function of x_i. Entropy is maximized if the distibution is uniform. It means that the uncertainity is maximum about the possible position of the user. We need to know how certain we should be in order to inform the user of the possible user location.

Figure 4.8 shows the normalized entropy of the score distribution at each anchor point. At positions 5 to 9 the accuracy was under 8m where as 1-4 and 10-13 the error was greater than 8m. The best match at positions 6 and 8 were estimated the correct position but both the entropy and min-max distribution does not infer a trend. From calculating entropy we wanted to find out if we can extract any information about the certainty of the correct position estimate, so that we can assign a weight accordingly. But as seen from the trends, this is not the case hence we used a different approach to use Wi-Fi for position correction. We assign weight $w1, w2$, and $w3$ to the best 3 matched anchor points only if they are all within 2 hop neighbours to each other. Otherwise we ignore the Wi-Fi scan. It means that each anchor point in the top 3 matches should be in the same neighbourhood and have not more than one anchor points between them which are not in the top 3 rank. Here we used weights of 0.4, 0.3 and 0.2 respectively for $w1, w2$, and $w3$. We use these weights because we want to give more weightage to the anchor points which more closer similarity with the live Wi-Fi reading, but we only consider the top 3 matches as the top 3 matches are more likely to be the real position of the user.

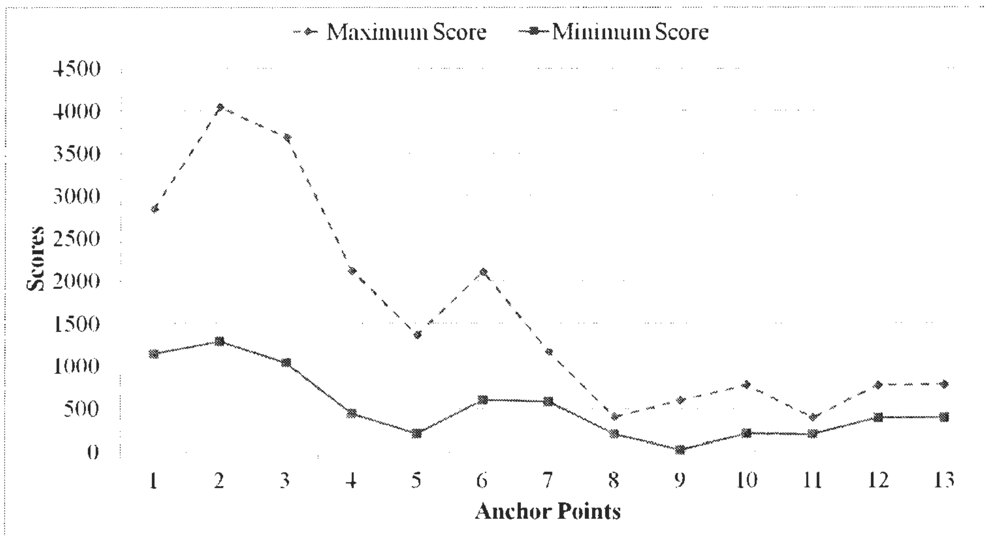


Figure 4.7: The minimum and maximum scores at different anchor points.

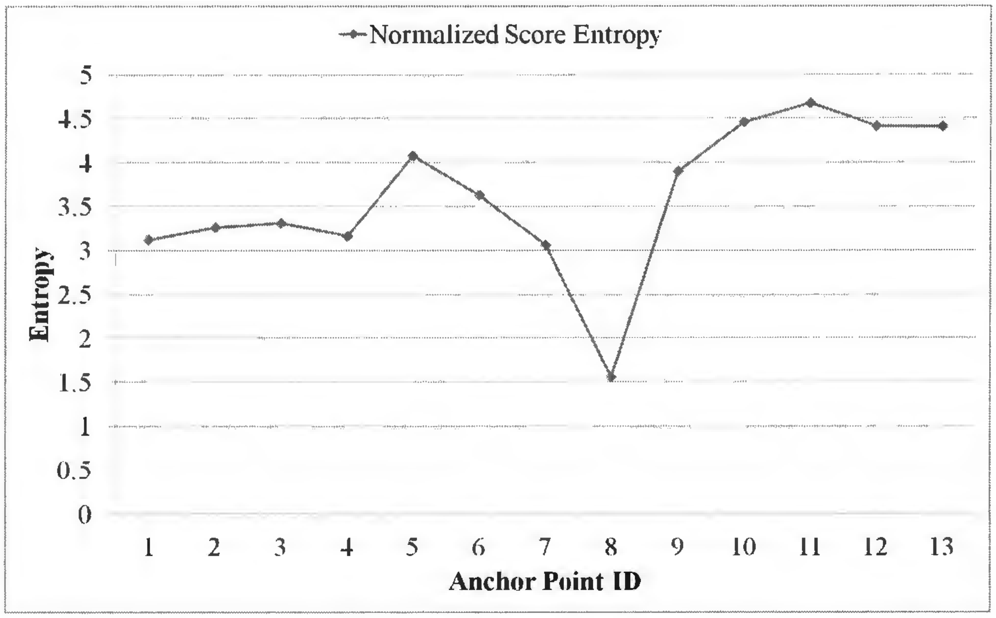


Figure 4.8: After normalizing the scores, entropy is calculated.

Chapter 5

Hybrid Motion and Wi-Fi Integrated Localization Scheme

This chapter explains the integration of motion based stepcounter described in Chapter 3 and a Wi-Fi based positioning scheme as landmarks described in Chapter 4 in collaboration to estimate positions of the users in an indoor environment.

5.1 Motivation

Different indoor localization schemes have different positioning accuracies, however there is no standard specifications availble yet requiring localization technologies to meet certain reequirements. The localization schemes which require extra equipment in the environment like [22][41] are accurate up to a few centimetres. Whereas Wi-Fi based positioning like [32] claim an accuracy of around 2-4m in areas where sufficient training data is available. Overall the main challenges in indoor environments are

- GPS delivers poor performance when there is no line of sight between the GPS receiver and the sky, so practically they do not work indoors.
- GPS and Wi-Fi exhibit high-power consumption [7][34].
- In places where Wi-Fi is available in limited areas and access points are deployed sparsely, localization becomes more challenging when relying only on Wi-Fi based technologies.

Some researchers may argue that Wi-Fi based localization techniques are sufficient for indoor environment and the power consumption of Wi-Fi may not be a big concern because we might not need localization service all the time. It might be true for a category of location-based application such as [20] in which the user just wants to geo-tag a location. Nevertheless, the majority of location-based applications require continuous localization like location-based social networks, user tracking and navigation etc.

The bigger motivation for us are locations where Wi-Fi infrastructure is not that dense for example tunnels, skywalks and other areas in buildings where Wi-Fi is not available everywhere. For example in tunnels and parking lots, Wi-Fi might not be readily available but there might be points where certain AP signals are detected. We can treat these points as landmarks and they can be used as position correction if use motion-assisted localization scheme.

5.2 Mobile Application

5.2.1 Platform

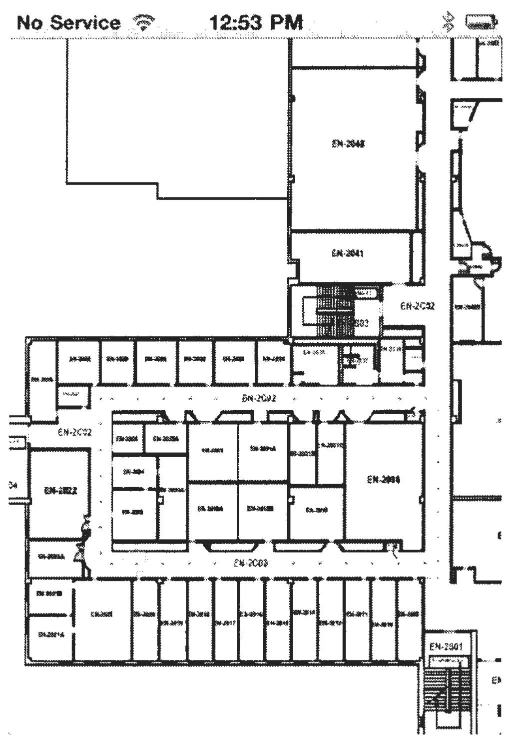
There are several smartphones available in the market from handset manufacturers like Apple, Samsung, Nokia, Blackberry, Sony Ericsson, HTC. For mobile OS the most popular ones are iOS(Apple), Android, Blackberry, (Google), S60(Nokia). According to our literature survey most of the research groups work on Nokia S60 or Android platforms although some did use iPhone in their research [37]. The reason for this is that they are open source and many third party API's are available from the developer community. In S60, different developer plugins are available, for example pyS60 for quick development using python. For our purpose we would be using the iPhone 4 as it has all the IMU sensors required for our research. Furthermore 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's 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.

Apart from the hidden private Wi-Fi framework, we use the iOS Core Motion Framework and Core Location Framework. The Core Motion Framework gives us access to the raw accelerometer readings. IT also provides us with the device attitude which uses internal calculation from the accelerometer and gyroscope. We use this to calculate the yaw of the device. Location framework is used to get the device magnetic heading, which we use for our heading estimation.

5.2.2 Interface

The goal of our touch-based UI design is to study and implement our proposed scheme and to test in the field. We used a simple map based interface showing anchor points and also displaying the relative probability distribution by overlaying circles on the anchor points. We display relative probability with a the anchor point with highest probability showing the largest circle. The map can be zoomed in and zoomed out. Figure 5.1 shows the user interface.

Figure 5.1: Map interface of mobile app. Orange circles showing the relative probability distribution.



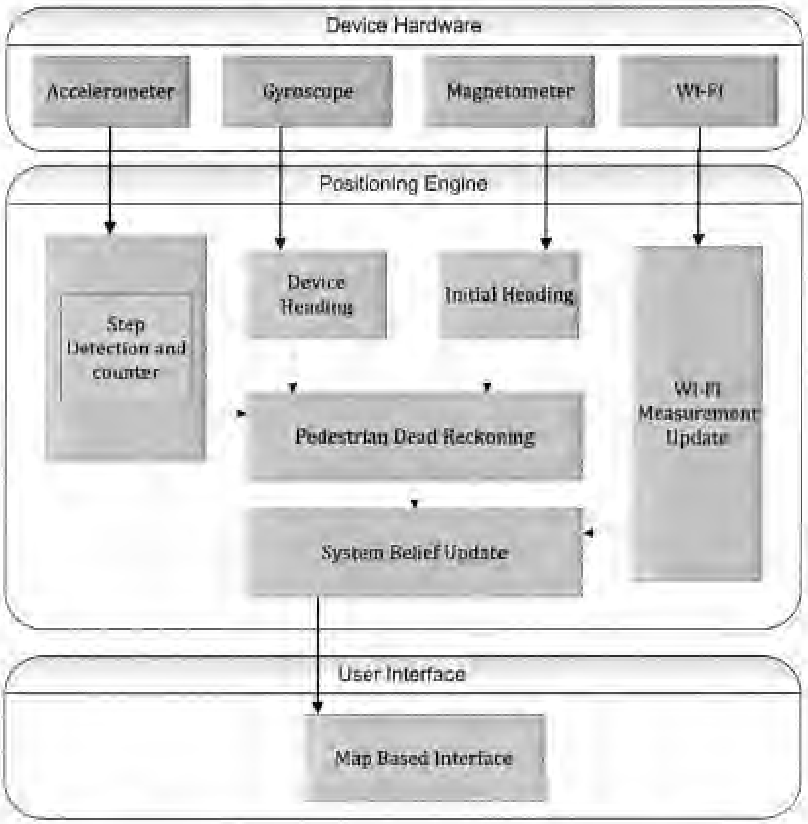
5.3 System Architecture

5.3.1 Design Overview

Figure 5.2 shows the block diagram of our proposed system. In our localization scheme we divided our map into grid spaces. The center of these grid spaces are the anchor points with known physical coordinates (x, y). The grid space between two anchor positions determines the resolution or granularity of the positioning system. The $bel(x_t)$ is the belief representation of our environment where $bel(x_i)$ is the probability that the user being at i-th anchor point. The initial *belief* of the system is assumed to be uniform as the system does not know where the user is positioned. As the magnetometer is noisier compared to the gyroscope when giving heading estimation [56], we use the magnetometer only for estimating the initial orientation of the user with respect to the environment so that we can detect which direction the user is facing. This is one of the assumptions of our system that we ask the user to face parallel to any corridor during the initialization so that the system detects the initial orientation. After this initialization/calibration process we keep track of the heading using the gyroscope. Periodic re-initialization from the magnetometer may be useful, but it was not tested in our experiments. We use the stepcounter mentioned in Chapter 3 to estimate the distance travelled and gyroscope to estimate the direction in which this distance is travelled. As shown in the Figure 5.2, accelerometers are used to detect the steps taken. The stepcounter and the gyro-assisted heading form part of the motion model described in Chapter 3. The motion model is used to update the belief where the user is in our system after every fixed amount of steps. The measurement update uses our Wi-Fi localization method described in Chapter

4. Different strategies can be used for updating the belief, which are discussed in the next section.

Figure 5.2: System Architecture



5.3.2 Update Strategies

There are various approaches to update the belief from the motion model or the measurement model. Whenever the belief is updated, the display to the user is also refreshed. The following strategies can be considered:

- **Query Strategy**: In a query strategy, the system requests an update of the position or the belief on demand.

- **Immediate Strategy**: An immediate update is triggered when the position changes with regard to the last reported position.

- **Periodic Strategy**: A periodic update is triggered if a pre-defined time interval has elapsed since the last update.

- **Distance-Based stategy**: In this strategy, the mobile device always keeps track of the distance between the current and last reported position. If this distance exceeds a predefined threshold, it performs an update.

- **Zone-based strategy**: An update is initialized if the target enters or leaves a predefined zone, where a zone can be fixed as a single point, location or area.

For the motion model, we use the immediate strategy to update our belief after a fixed number of steps. We do not use time as a factor to update our belief. When there are no steps detected, it can be assumed that the user remains at the same location. On the other hand, for Wi-Fi measurement update, we use query strategy. For example after a few hundred steps we might need to update as the error might have accumulated or the probability distribution become more uncertain.

5.3.3 Position Presentation

The user is interested in the final position estimated by the system. There can be situations where multiple anchor points have very similar probability for user position. In order to output the most likely position as the position estimate, we need to know if the belief is not very uncertain. For example, when the user starts the application, the belief is uniformly distributed and in this situation it is unwise to output a position estimate to the user. The position has to be outputted to the user after knowing some kind of certainty that the belief has converged to some probable positions. In Chapter 6 we show that we can calculate the entropy from the belief distribution to see the uncertainties.

Chapter 6

Evaluation

We will explain our experimental methodology, settings, scenarios, and results in this chapter. Our main experimental goal is to measure the benefit of using motion information to track and position the user in an indoor environment.

6.1 Methodology

The system evaluation contains multiple phases. The first phase is to test the performance of our step counter which is a major part of our motion model. After checking the accuracy we can determine if it is good enough to be used in our motion model. The accuracy and precision of our motion model is then tested in two different indoor environments.

The second phase is the evaluation of our measurement model. By analyzing the performance metrics, we can determine if it can be used for opportunistic measurement update. Furthermore, it is important to test our system in an environment which has sparse Wi-Fi coverage. Next, we explore the benefit of using motion for localization and tracking and analyse the advantages of using rank based Wi-Fi in sparsely distributed Wi-Fi environment. We measure the benefit in the following aspects:

- *System Performance*

  Hypothesis 1: *The system accuracy and precision of motion assisted indoor positioning is better than other localization systems in sparse Wi-Fi environment.* Most of the current indoor technologies used are essentially Wi-Fi only. Their performance is related to very laborious training of the environment. Our system's motion model should be able to accurately position and track a user walking in an indoor environment. The turns in the environment are helpful in shortlisting the user's possible positions. Although the error while walking in the same direction accumulates, turning into another corridor should reduce this error. We argue that using the motion model alone is sufficient for short-term user tracking. Wi-Fi based corrections are beneficial, especially in sparse Wi-Fi environments where there are only a few access points. Our system will require only few Wi-Fi training points in these environments and would perform much better than other Wi-Fi dependent indoor localization schemes.

- *Cost*

  Hypothesis 2: *The system training and maintenance cost can be reduced.* The system training effort is reduced in a sparse Wi-Fi environment as fewer survey points are needed for data collection. The motion model does not need any training. More importantly, if the environment has unique features in terms of corridor layout and number of turns, the system will require fewer Wi-Fi landmarks and can be more dependent on the motion model alone. When the indoor environment changes (e.g., Wi-Fi infrastructure or environment layout alteration), the RSSI fingerprints database has to be updated or even re-generated from scratch in order to adapt to such changes. If the number of such survey points are fewer the cost to update will be lower compared to other Wi-Fi based systems.

- *Scalability*

  Hypothesis 3: *The system can work in different indoor environments.* The system is scalable as it can be quickly adapted to any environment, both with dense Wi-Fi and with limited Wi-Fi coverage. Only environment maps are needed with internal representation of possible user position points. Moreover the resolution of the grids can also vary and the accuracy would not directly depend on the grid resolution. As accuracy depends more on the stepcounter rather than how dense is the grid.

- *Robustness*

  Hypothesis 4: *The system can recover from false position estimates.* Unusual movement of the user may confuse the system. For example, if the user is walking in a circle, it is possible the system might become more uncertain about its position. We argue that our system over time can recover from this uncertainty.

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

6.2 Experimental settings

Experiments and evaluations of our motion model, measurement model and hybrid localization scheme were carried out in two contrasting environments at Memorial University. The first was part of the 2nd floor of the Engineering Building. The space was divided into a grid using a 3×3m cell size. 42 positions were selected within the hallways for the anchor points. 33 of these anchor points were surveyed for Wi-Fi data and a fingerprint was created for each anchor points. The survey points are those anchor points where Wi-Fi training was done and we have a Wi-Fi fingerprint available. The anchor points are possible locations the user can be in the environment. The distance between two anchor points is nearly 6 steps (3.5m), so belief is chosen to be updated after every 6 steps in this environment. Figure 6.1 shows the map of the Engineering Building field test environment.

The second environment is the Tunnel system which connects different buildings of the university. There is no Wi-Fi coverage provided for the tunnels. Figure 6.2 shows the map of the tunnel system. The only Wi-Fi signals available are at entrance positions. Hence the areas of Wi-Fi AP visibility is very limited and also sporadic in nature. The Engineering Building has more sharp turns, whereas the tunnel has smaller turns. The distance between two anchor points here is 5.5m. Therefore the belief update happens after every 9 steps. Most of the commercial pedometers choose step length as $0.413 \times h$, where h is the height of the user. In our experiments step length is kept at 0.69m.

The major assumptions for our experiments are as follows

- The user is always located in the areas for which the anchor points are defined

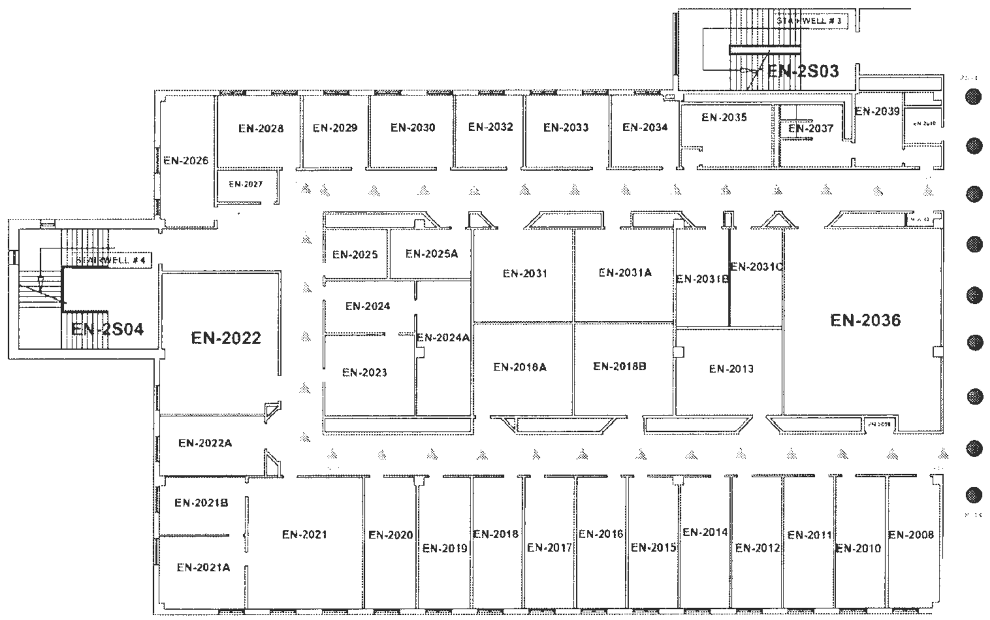


Figure 6.1: Map of the Engineering Building. Green triangles are the anchor points where data has been collected and the system has fingerprints for those locations. Red circles are untrained areas.

in the system.

- The device is always pointing in the direction of the user's motion.
- The user walks close to the corridor's center.

6.3 Motion Model Evaluation

6.3.1 Performance of Step Counter

The step counter was evaluated by two different users by walking 500 steps holding the device in the hand. The experiment was repeated 3 times by walking the same

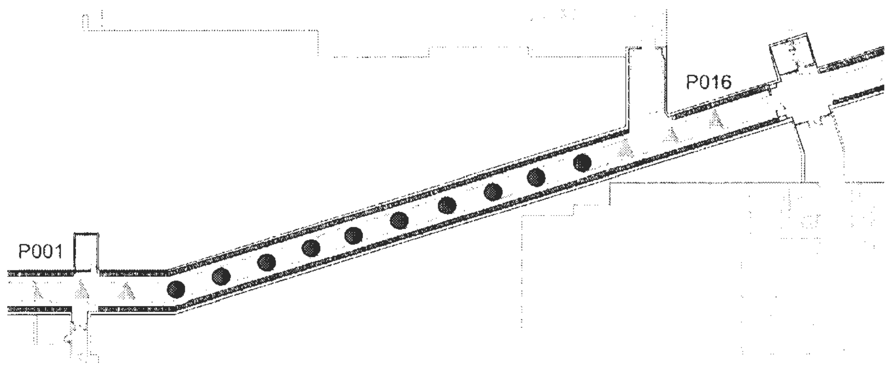


Figure 6.2: Map of part of the university tunnel. Green triangles are the points where Wi-Fi is sporadically available and red discs are positions where no Wi-Fi is available. Fingerprints for locations with green triangles are available.

path. Figure 6.3 shows the accuracy of the step counter. Intuitively it can be seen that the step detection depends a lot on human gait. Apart from this it also depends on how a user is holding the device. Some users tend to hold the device in a more stable manner while others sway their hands while walking. But this problem can be solved by multiplying a user specific scaling factor to the threshold of step detection. The accuracy of the step counter was comparable to other commercial step counters available on Apple's app store. Therefore it was considered reliable enough to use in our motion model.

6.3.2 Initial gyroscope calibration using magnetometer

Figure 6.5 shows the magnetic map of the environment to show more deviations near the corners compared to the middle of the corridors. When the application starts, the gyroscope has to be initialized to the orientation of the user in the environment using magnetometer. The magnetometer is noisy, a small experiment was done to

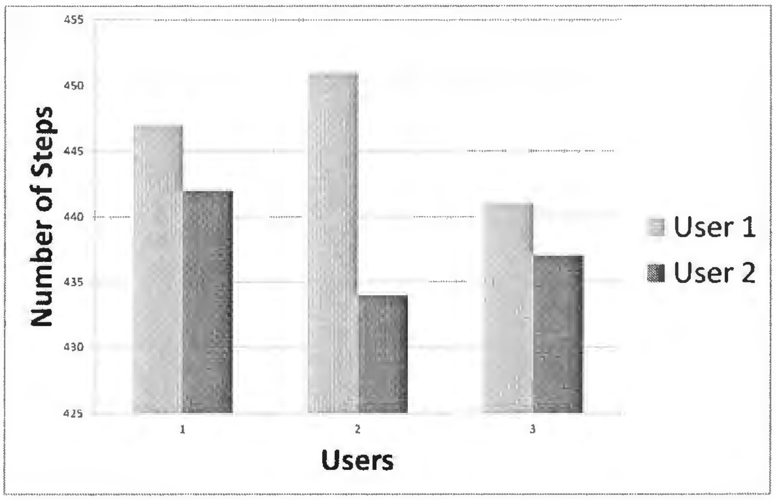


Figure 6.3: Number of steps detected when walked 500 steps

see the stability of the magnetic heading readings in the environment. It has been noted that there is greater magnetic instability and interference in the corners and intersections. The standard deviation of magnetic readings in the major parts of the corridors is 9 degrees whereas it is 21 degrees near or at the corners. In order to correctly identify the initial orientation, we set a check that in the initialization phase if the magnetic readings have a standard deviation more than 12 degrees. If so the initialization process is repeated. Figure 6.4 shows the heading readings when approaching an intersection. The horizontal axis describes the time in seconds.

6.3.3 Accuracy of Motion Model

In order to test the motion model the user walked in the corridors of the Engineering Building. Although in this experiment the Wi-Fi integration was disabled but only those anchor points were considered in which we had Wi-Fi fingerprints available. To denote the true position of the user in the map a small human figure marker is used

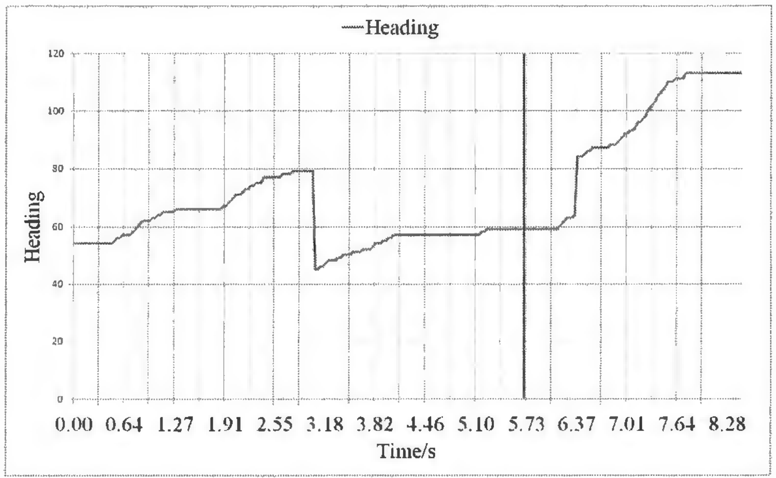


Figure 6.4: Magnetic heading readings when walking from a center of a corridor to the intersection of corridors in the Engineering Building.

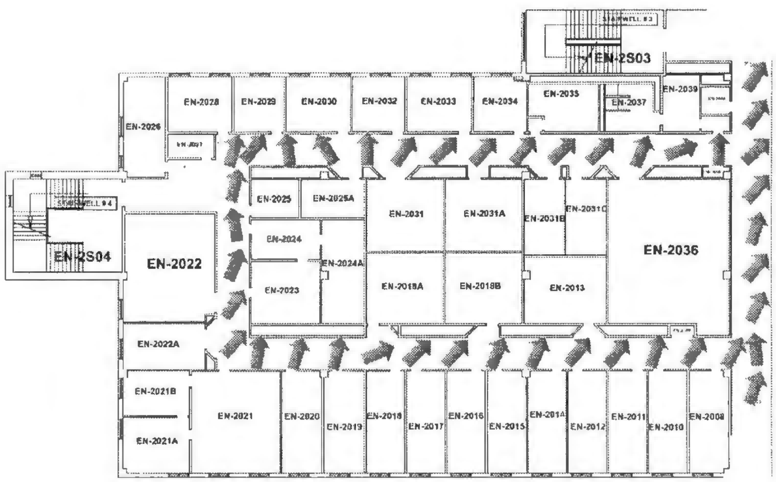


Figure 6.5: Magnetic Map of Engineering Building.

to show the true location and also the direction of walking. As the application starts the algorithm first calibrates for the heading of the device using the magnetometer. Once the calibration is done, the gyroscope keeps track of the orientation of the user while walking. The circles in the screenshots in Figure 6.6 show the belief distribution of the system. The anchor point with the highest probability will show the biggest circle and all the remaining anchor points will have circle sizes relative to it as the probabilities are normalized before belief distribution is shown to the screen. This way it is easier to visualize how the belief distribution is shifting and converging. It can be observed from Figure 6.6a that all circles are of equal size as in the beginning the belief is uniformly distributed. From Figure 6.6b it can be observed that during the application start-up the initial orientation has been detected as towards the right (East) with respect to the map, hence the probability distribution shifts towards those corridors which have a pathway towards East. Figure 6.6(c-f) shows how the probability distribution shifts along the direction where the user is walking. Although at this point the algorithm is uncertain where the user is positioned. However, it can keep track if the user turns back and starts moving in the opposite direction.

The user keeps walking towards the end of the corridor and turns right. Figure 6.7a shows that the probability suddenly converges to one of the anchor points near the corner. This happens because the algorithm detects that the user has taken a right turn. So that anchor point will have a higher probability to be the true position which will have the same relative motion from a neighbouring anchor point. Figure 6.7b shows that user is tracked as the probability shifts in the same way as the movement of the user. In Figure 6.7c two corner anchor points have almost equal probability as the belief was updated during the turn. The belief is updated every 6 steps taken by the user. This update frequency was chosen to correspond with the distance between two anchor points. The user then turns back start walking the same path the user came from. Figure 6.7(d-f) shows that the belief of the system shifts correctly with the motion of the user.

In a similar experiment, we also considered other anchor points in the area which were depicted as red circles in Figure 6.1. These anchor points do not have fingerprints as no Wi-Fi data was collected at these points. Other Wi-Fi only based solutions would not work very well in these conditions. Luo et al [32] did experiments under same conditions. Their error increased from 2m to 9m when they moved from trained area to untrained area. Figure 6.8(a-f) and Figure 6.9(a-f) depicts the screenshots of the positioning application when it walks in the untrained area.

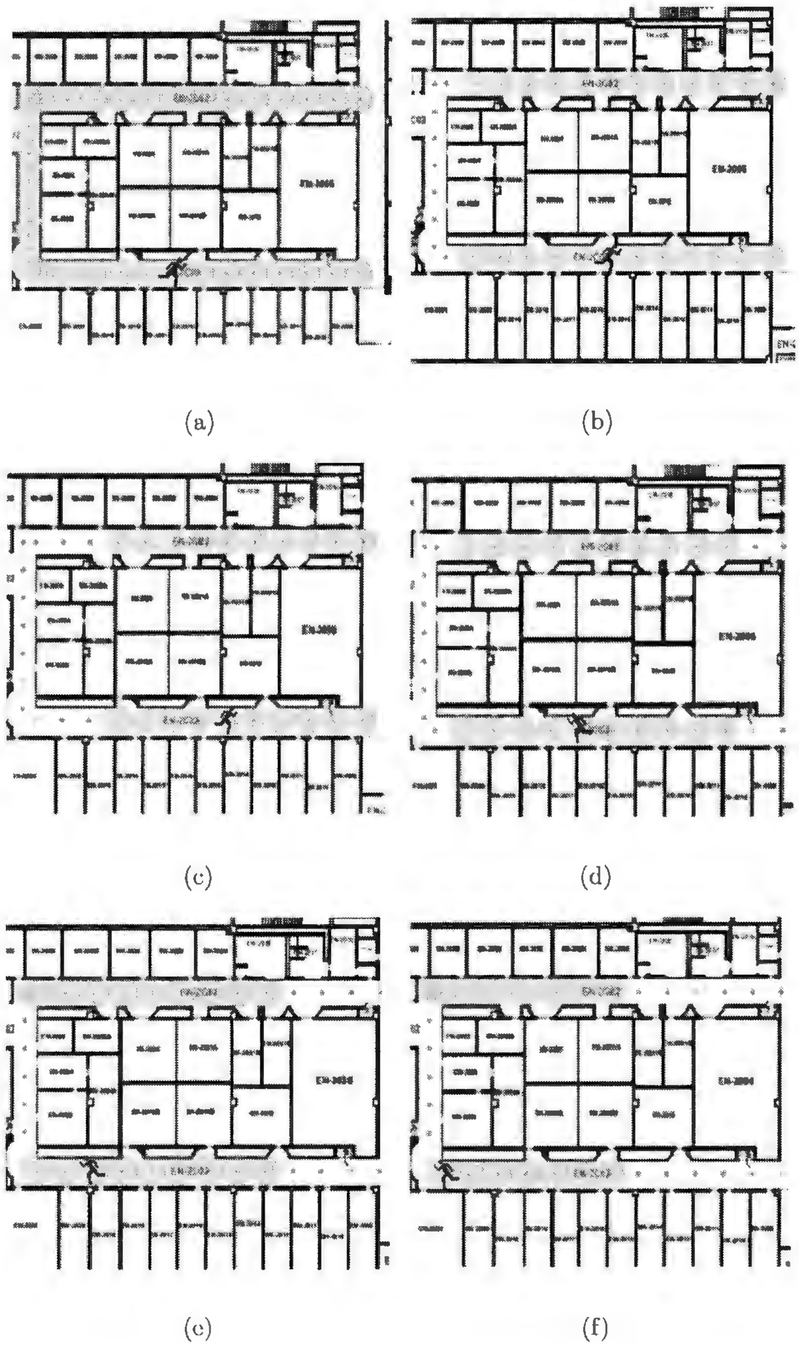
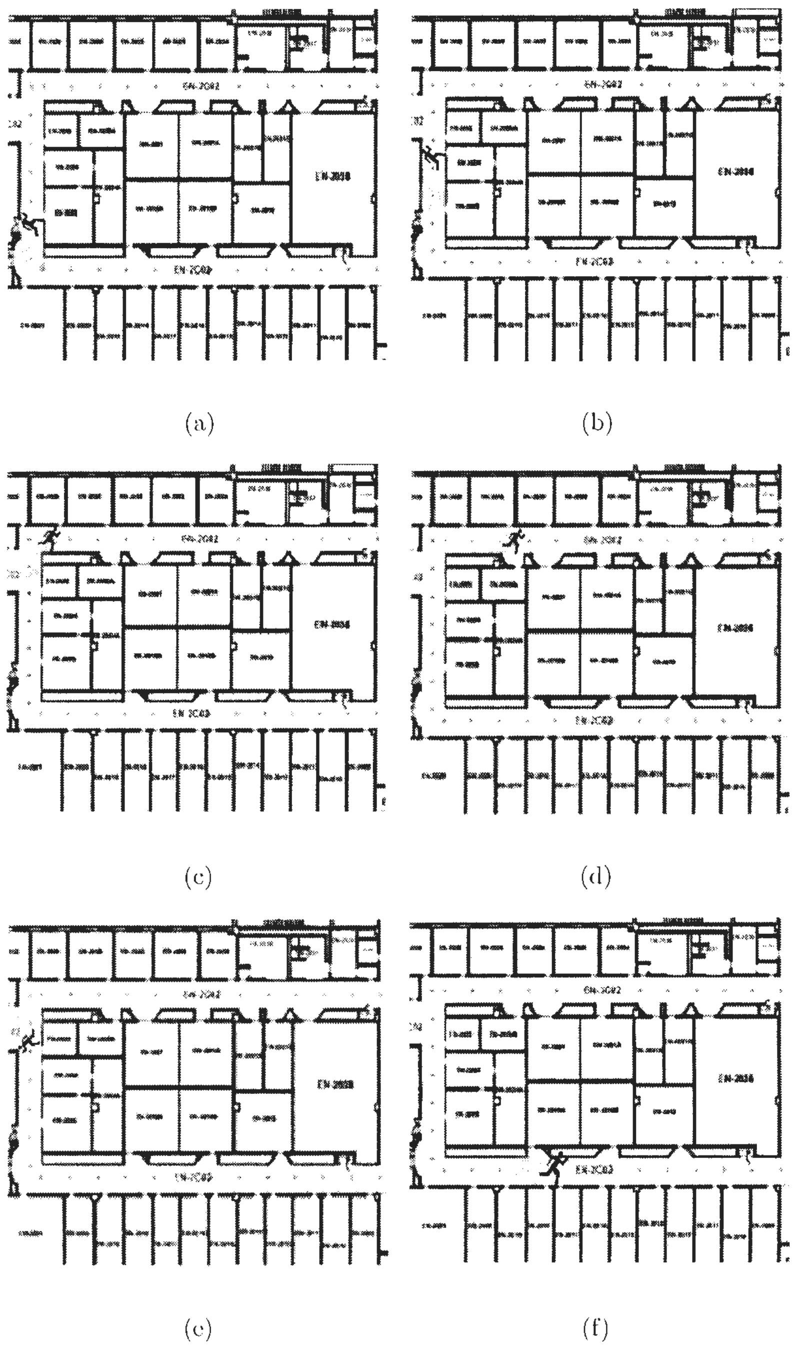


Figure 6.6: Screenshots of Motion Model in Engineering Building



(a) (b) (c) (d) (e) (f)

Figure 6.7: Screenshots of Motion Model in Engineering Building Continued from Figure 6.6

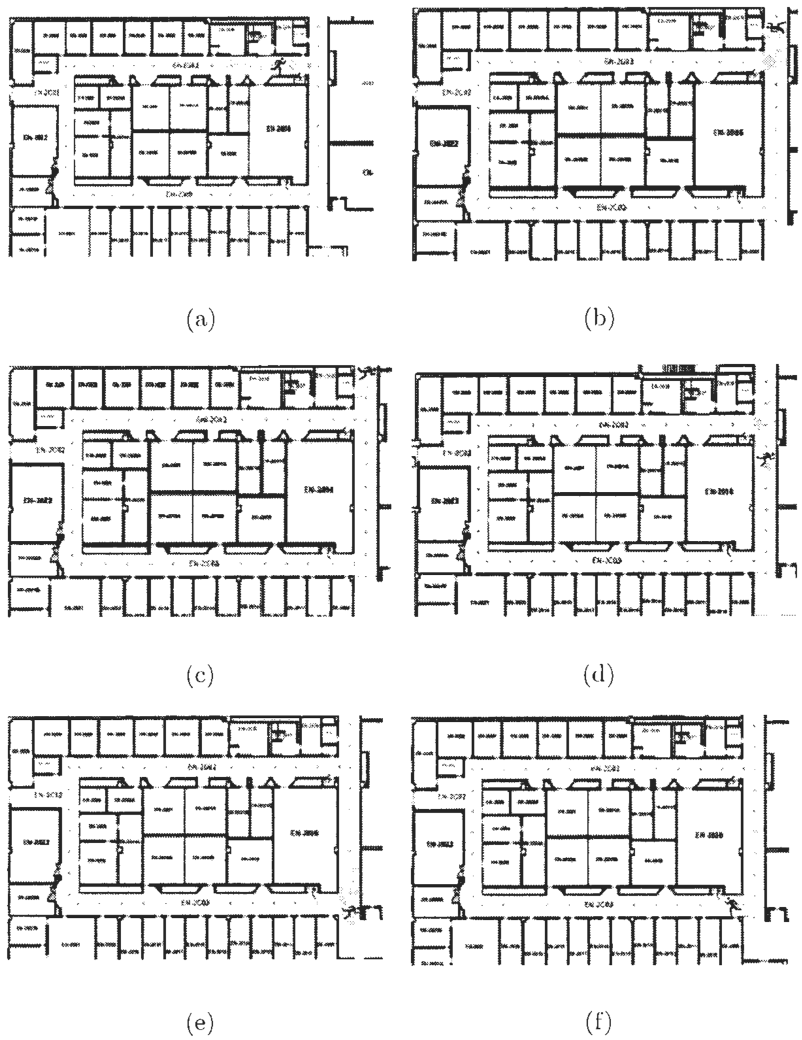
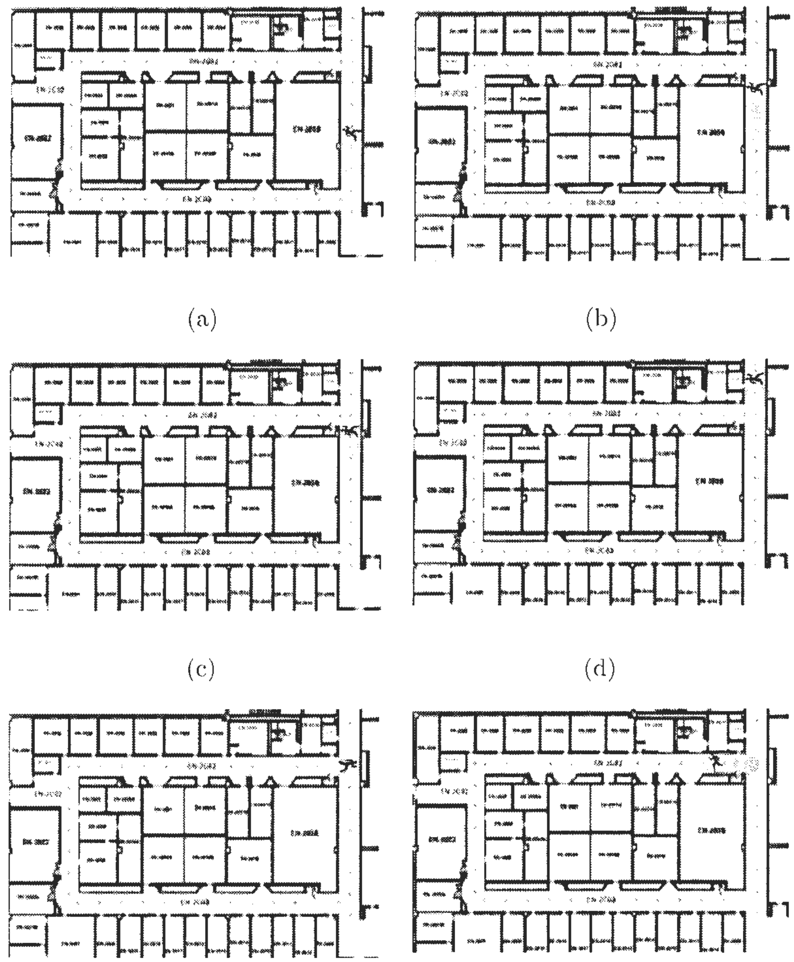


Figure 6.8: Screenshots of Motion Model in Engineering Building in Unmapped Regions



(a) (b) (c) (d) (e) (f)

Figure 6.9: Screenshots of Motion Model in Engineering Building in Unmapped Regions Conitinued from Figure 6.8

6.3.4 Entropy of belief

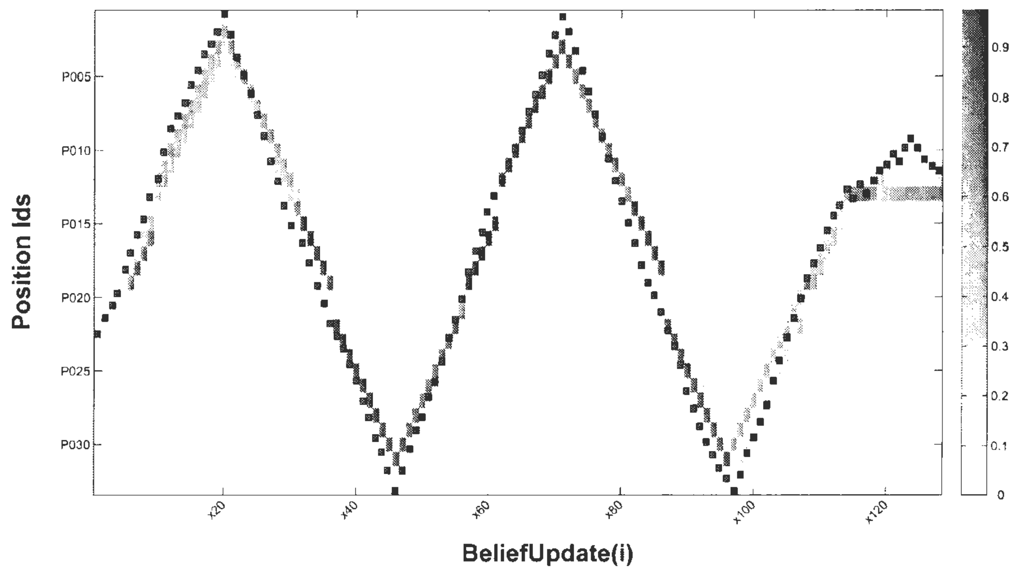
In another experiment the user was asked to walk in the corridor with our localization app in the trained areas of Engineering Building. Figure 6.10a shows the heat map of the probability distribution over time. The x-axis describe the ith update of belief. The position IDs are listed on y-axis where the color intensity shows the probability of being at each location. The belief at x36, x64 and x88 are examples where the position correction happens due to turning. Overall it can be seen that the position is tracked pretty well along the path of the user. From belief update x112 to x128 the user changed his direction of walking after a few steps a couple of times creating a to-and-fro user trail. It can be observed in the heat map that the uncertainty starts to increase as the probability distribution spreads out. Thus, a malicious behaviour by the user in terms of walking in circles and moving to-and-fro in the corridor over short distances might confuse the belief system.

Figure 6.10b shows the entropy of the same heat map. At x5 the entropy falls greatly due to a turn. Initially the probability was uniform so the entropy was maximum but as soon as the user turned the belief became more certain due to the recognition of a corner. Every time the user turns a corner, the uncertainty decreases and we can see a drop in entropy. After x112 the entropy increases, showing the confusion caused by user motion.

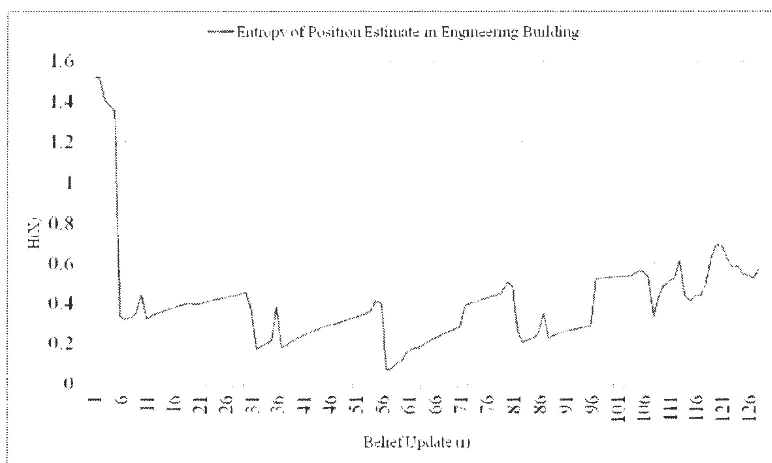
6.4 Rank Based Wi-Fi Measurement Model

Our Wi-Fi localization scheme returns similarity scores between the current measurement and every anchor point which has been surveyed for stored Wi-Fi data. The lowest score is considered the best match. To test the rank based fingerprinting technique we assumed that the best match anchor point is the estimated position. We tested this in our Engineering Building at each anchor point. The error was recorded by logging the distance between the ground truth and the estimated output position. Figure 6.11 shows the cumulative error distribution. The mean error was about 4.1m. We compared our system with the Wi-Fi based localization scheme by Luo et al [32] which uses a different fingerprinting approach for localization. They employ the Gaussian kernel, which is commonly used to calculate the likelihood between an RSSI fingerprint in system anchors and the live RSSI measurement to generate likelihood candidates. The top-k candidates are then used to determine a final position using the vertex p-centres problem.

Figure 6.12 describes a situation in which the Wi-Fi measurement was updated to a wrong location. This test was done in the Engineering Building, where the Wi-Fi APs are denser and the Wi-Fi environment is not sparse, meaning that at most of the locations, similar APs are visible. As in our Wi-Fi positioning module we create a rank of the APs visible to compare it with a fingerprint, due to fluctuations of the radio signals it is possible that it updates and positions the user at a wrong location. Similarly, there can be a scenario in which the error accumulates over time due to the motion of the user. In Figure 6.12a, it can be seen that, the user is present near the middle of the North corridor but the position estimate is in the corner. However, over time the probability distribution starts growing more uncertain, as can be seen in Figure 6.12b and Figure 6.12c. But after the turn, it again converges. Figure 6.12d shows that the motion model would be able to recover in this situation. Although in a sparse Wi-Fi environment, where the APs at one area are distinct compared to other areas, the error due to Wi-Fi will be smaller.



(a)



(b)

Figure 6.10: a) Motion model heat map at Engineering Building with dense Wi-Fi coverage. Black annotations describing actual user position. b) Entropy in the Engineering Building .

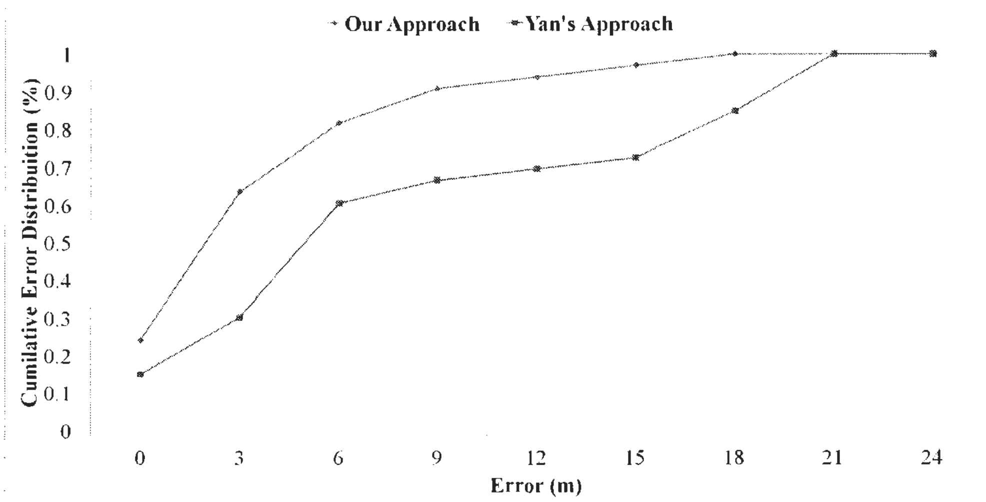
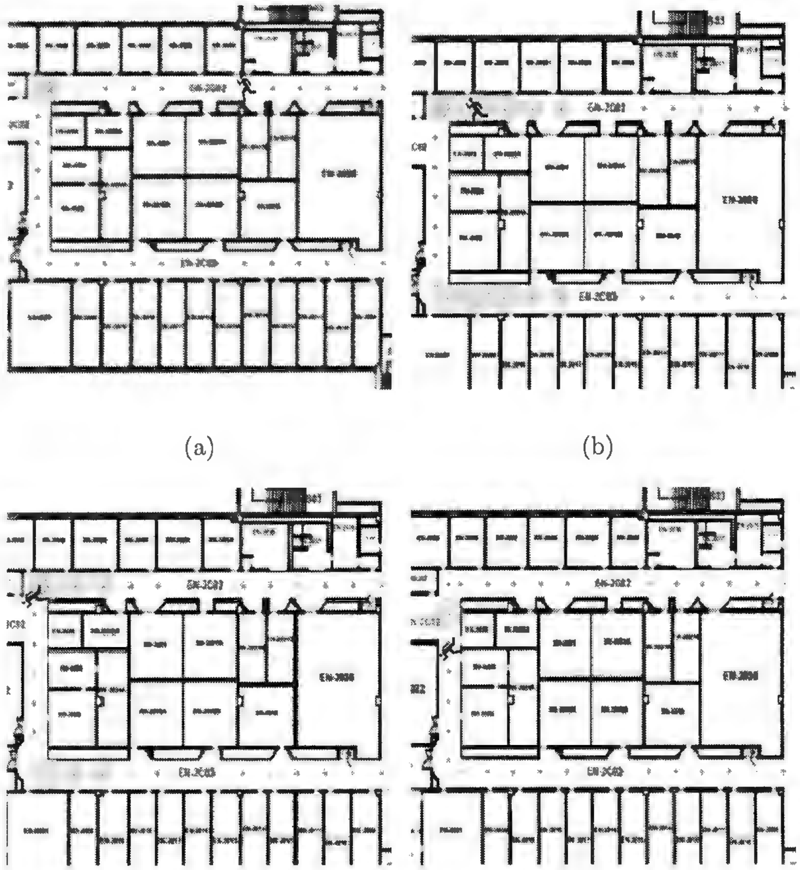


Figure 6.11: Cumulative error distribution of the rank based fingerprinting in Engineering building.



(a) (b)

(c) (d)

Figure 6.12: Recovering from an erroneous position estimate due to motion model.

6.5 Performance in a Sporadic Wi-Fi Environment

To test our system in an environment which has sparse Wi-Fi coverage, we chose the university tunnel system which has no Wi-Fi available but sporadic signals are available at the different entrances of the tunnels from different buildings. Figure 6.2 shows the map of one such section of the tunnel. This figure shows 16 anchor points from one entrance to another. All neighbouring anchor points are equally distant from each other. It is assumed that initially the system does not know the user's true position. Initializing with a Wi-Fi scan can initialize user position if the user is in one of the entrance areas.

Figure 6.13 shows the heat map of the user's walk in the tunnel. On horizontal-axis we have the belief updates and on vertical-axis we have the 16 anchor points. We annotated the map with approximate actual position of the user to compare the belief distribution with the movement of the user. At x0 the belief is uniformaly distributed but from x0 to x12 we can see that the belief slowly converges. From x12 to x45 the probability distribution is not that scattered and position estimates are more confident. From x45 to x60 the probability distribution becomes less reliable as the user changes his direction more frequently similar to the test done in Engineering Building. At x60 the Wi-Fi measurement update is triggered. At this point it detects P001 as the most likely position. The probability distribution shifts heavily towards that position as we give higher weight to the anchor points with higher Wi-Fi similarity. In the tunnels the Wi-Fi is sporadically available in only P001-P004 and then P015-P016 as described before. No Wi-Fi is detected in any anchor points between them. Hence when the Wi-Fi update step is triggered, due to the diversity of visible

AP's between these two regions, the position correction has smaller error.

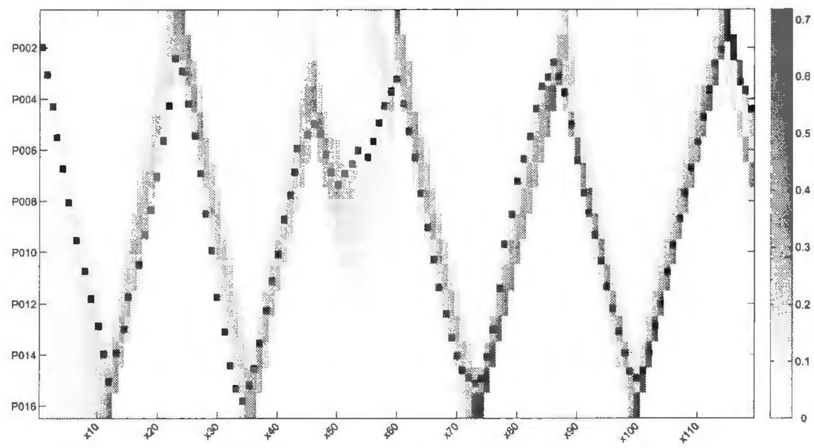


Figure 6.13: Heatmap of Motionmodel in the tunnel.

Figure 6.14 shows the entropy of the belief in the tunnel. If we compare the entropy plotting of Engineering Building and tunnel it can be observed that the entropy in the tunnel does not drop as much as compared to the entropy in the Engineering Building. This is because the tunnel lacks sharp turns as compared to the Engineering Building. Although the accuracy from the most probable position estimate is comparable in both locations the certainty is less because of the absence of sharp turns. At x51 to x59 it can be observed that due to the to-and-fro motion in the same corridor the entropy increases. It sharply decreases again at x60 when Wi-Fi measurement update is triggered.

Next, we will consider the hypotheses mentioned in section 6.1 in light of our experimental results.

- *System Performance*

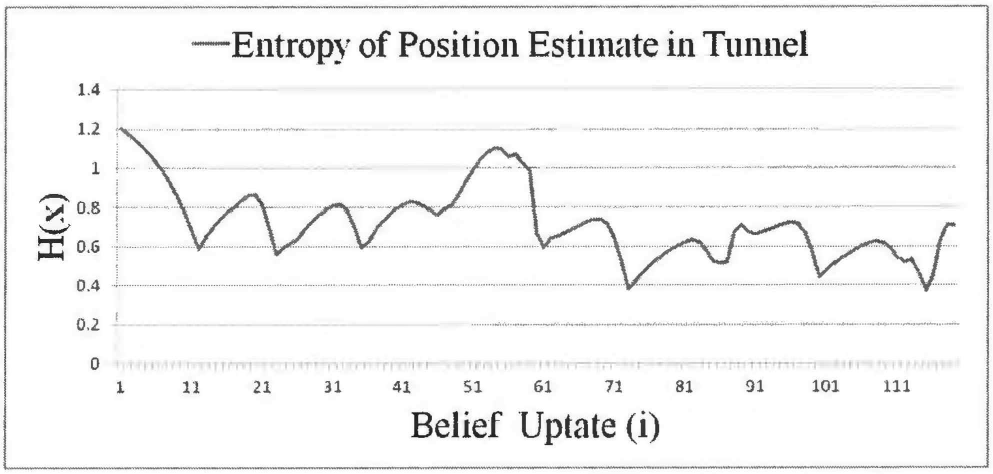


Figure 6.14: Entropy in the Tunnel

Hypothesis 1: *The system accuracy and precision of motion assisted indoor positioning is better than other Wi-Fi only localization systems in sparse Wi-Fi environment.* As it can be seen from the heatmaps of both environments that the system tracks and positions the user with fairly good accuracy regardless of the density of Wi-Fi coverage. For our experiments the accuracy in the Engineering Building was under 4m whereas in the tunnels it was around 6m on average. The best-performing but intensively trained Horus system [62] has a 0.7m to 4m average positioning error using 100 Wi-Fi scans and much smaller grid space (1.52 m and 2.13 m). Generally for our system a single accuracy figure can not be given as it depends upon the shape and size of the environment. Sharp turns help reduce positioning error estimates and long corridors accumulate errors. The second factor is the amount of Wi-Fi landmarks available for position correction.

- *Cost*

Hypothesis 2: *The system training and maintenance cost can be reduced.* We tested our system in two different environments. One had dense Wi-Fi coverage and had training data available for all the anchor points. On the other hand in the tunnel environment, the Wi-Fi was sporadically available at only 6 locations. No survey was done for those anchor points which had no Wi-Fi coverage so they were treated as untrained anchor points. As different areas in such environments have distinct Wi-Fi visibility, this can be exploited to our advantage to correct the position only and rely more on human motion for positioning. In our motion model evaluation, we observed that in the environment where there are more turns, the position estimate is better than the environment with less turns. Turns help the motion model to detect change in orientation and inherent map matching in the motion model help to converge the belief. Due to less reliance on Wi-Fi, minor changes in Wi-Fi infrastructure will have less impact on the system perfprmance.

- *Scalability*

Hypothesis 3: *The system can work in different indoor environments.* We tested our system in two completely contrasting environments. One had sharper turns with denser Wi-Fi coverage and the other had less turns but sparse Wi-Fi environment. The grid size in both the environment was also different as it was 3m in the Engineering Building and 5.5m in tunnels. This system is more scalable than other indoor positioning systems as it would require less training and would even work in sporadic Wi-Fi environments where Wi-Fi only systems would fail.

- *Robustness*

  Hypothesis 4: *The system can recover from false position estimates.* In both the environments during our field test we confused the system by walking in to-and-from (Figure 6.10b and Figure 6.13) fashion to create more uncertainty in the belief. When triggered Wi-Fi updates remove this ambiguity. If Wi-Fi is updated in the wrong location, it can be recovered in two different ways. The first one is due to the motion model the belief starts to become more uncertain. It starts to converge again if there is a turn which can uniquely position it. The second way it can be recovered is when another Wi-Fi update is triggered. Although Wi-Fi update can be erroneous too, but there is a chance that the error is reduced.

Chapter 7

Conclusion and Future Work

7.1 Primary Contributions

Wi-Fi based localization technologies are relatively robust and accurate compared to other indoor localization technologies. One of the main factors for these technologies to be popular is that the infrastructure often already exist. The RSSI fingerprinting based schemes perform better than triangulation based schemes because they do not depend on specific signal propagation models. However, the system performance greatly depends upon the rigorous training process and regular system maintenance in the form of regular fingerprint updates. These regular fingerprint updates are required if there has been any changes in the environment in terms of replacing a access points or moving furniture etc. In addition to that, these systems do not work in areas where Wi-Fi coverage is sparsely distributed.

These shortcomings can disable above mentioned localization systems. Moreover, because of high system overhead in terms of training data and cost of war-driving, we believe there is a need for more efficient and cost effective techniques. We believe that reducing training and maintainence cost and increasing the system robustness are very promising research directions.

In addition, we see that the current generation of smartphones have various embedded sensors including motion sensors like accelerometers and gyroscope. Although GPS receivers are present in most smartphones, they are of no help indoors. But magnetometers can be used to detect direction and heading. We recognize the opportunities presented by these sensors to detect human motion and the possibility to incorporate this knowledge to help position users in an indoor environment. Hence, we would also not rely on any external infrastructure except Wi-Fi coverage which is likely to exist in many environments.

In this thesis work, the primary contributions are evaluation of a motion assisted indoor positioning system for an indoor environment especially focused on sparse Wi-Fi coverage. We can use ideas from robotics in which a belief is maintained about the possible position estimate rather than relying on dead reckoning to output one final pose estimate. The distance moved by the user is calculated by the number of steps taken and then estimating the user trail by calculating the direction of each step. The user trail is matched with possible path signatures from the environment map using the motion model. The best match yields a higher likelihood for position estimate. Hence more distinct features in terms of turns and direction of corridors will give us higher accuracy. But in environments with similar corridors in terms of length and orientation, we will get multiple hypotheses for the user's position. In this situation we use Wi-Fi based position correction. Our Wi-Fi position estimation techniques uses rank on the visible APs based on their strengths rather than the actual

RSSI values. This technique has an additional benefit of being device independent as different manufacturers of networks cards have different standards for RSSI values but rank information is invariant to any monotonic increasing transformation (bias and scale) [33]. Wi-Fi AP's is used as landmarks to update the position belief when it is required by the system to update its position. This can happen after a fixed number of steps to avoid error accumulation due to the motion model.

One of the major benefits of this system is cost effectiveness. The initial training required by doing war-driving and collecting Wi-Fi data decreases significantly. Although the tradeoffs between accuracy and cost of training will depend on the environment, we can see the real benefit in such a system in sparse Wi-Fi coverage area.

Based on these principles we built a prototype mobile application for the iPhone and conducted experiments to evaluate it. Our experiments showed encouraging results and indicate motion assisted positioning as a viable option for indoor environments. The system is scalable and more cost effective than Wi-Fi only schemes because it requires less training.

During the course of this research, a number of publications have been made. An overview of related indoor localization technologies which are using smartphone sensors are summarized in [55]. The research work in developing a stepcounter using smartphone accelerometer which is mentioned in Chapter 3 is presented in [56]. Finally, a short overview of our research with some results in Chapter 4, Chapter 5 and Chapter 6 are published in [57].

7.2 Discussion and Future Work

We believe that this system can be further improved in a few interesting ways. For both motion model and Wi-Fi position estimation, we did not use the best strategy available because our goal was not to improve either separately. For example in the step counter we are detecting the number of steps taken but using the height of the user as a parameter to determine the stride length. A more adaptive approach could be taken here which uses information from accelerometer to also calculate the stride length. Artificial intelligence techniques can be employed in the initialization phase for the system to learn the human walking pattern and determine the style of the user to more accurately determine the number of steps.

Similarly for Wi-Fi based localization, more accurate schemes could be employed. Pre-processing the APs after observing the environment for fluctuations could improve the localization error.

Another interesting aspect in which the system can be improved is to integrate human-centric collaborative feedback. Positioning accuracy and precision can be improved by collecting both positive and negative feedback from users in terms of orientation. Luo et al [32] user collaboration to improve system performance. If the system gives a position estimate to the user which the user feels is true, the user can leave a positive feedback which will result in putting higher weight to current system parameters. When the user is not happy with the position estimation by the positioning system, the user can leave a negative feedback similarly. In areas where there are no survey points, the user can help in creating one. This will also be helpful for decreasing system maintenance costs and improving accuracy of the system over time.

Developing a magnetic map is also one idea which can be explored. In their case we have to observe how stable is the magnetic environment over time. In indoor environments there may be areas due to electronic equipment or wiring, where the magnetic field perturbations are distinctive. They can be used as landmarks similar to how we use Wi-Fi.

Camera based localization is also feasible, and it would be an interesting approach to use it in collaboration with our system. In [34] they use vision as one of the fingerprint parameters for logical localization to differentiate between two locations. Normally when the user is holding the phone as in our assumption, the phone's forward camera is always pointing down at the floor. Most of the indoor environment have tiles as floors or carpets. Tile counting or some kind of floor recognition during walking would be beneficial in improving the accuracy of the localization system.

We believe that some organizations or companies will devise specifications for indoor positioning system in the near future. With the potential rapid growth of location-aware services for public indoor environments such as airports, subway systems, museums, university campuses, shopping centers, etc there will always be areas where Wi-Fi infrastructure will not be available and hence some reliable and scalable alternative technology would be needed. At this time we believe human motion based localization schemes have great potential and look to be very promising in reducing the cost both in the sense of maintenance and energy consumption. We also believe that more and more researchers will be attracted to exploit the various sensors now available in smartphones for indoor localization.

Bibliography

[1] Dodgeball. http://www.dodgeball.com.

[2] Facebook. http://www.facebook.com.

[3] Foursquare. http://www.foursquare.com.

[4] Geolife: Building social networks using human location history - microsoft research. http://research.microsoft.com/en-us/projects/geolife/.

[5] Loopt. http://www.loopt.com.

[6] Indoor LBS And Hyper - Local Content Is the Next Gold Rush for Mobile Commerce. http://www.indoorlbs.com/2010/06/indoor-lbs-and-hyper-local-content-is.html, Oct. 2010.

[7] M. Amir. Energy-Aware Location Provider for the Android Platform, 2010.

[8] P. Bahl and V. Padmanabhan.

[9] L. Bao and S. S. Intille. Activity Recognition from User-Annotated Acceleration Data. pages 1–17. Springer, 2004.

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

[11] W. Bumgardner. Step Size in Pedometers . http://walking.about.com/cs/pedometers/a/ pedometerset.htm.

[12] A. Carroll and G. Heiser. An analysis of power consumption in a smartphone. In *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*, USENIXATC, pages 21–21, Berkeley, CA, USA, 2010. USENIX Association.

[13] P. Chapman. Location-based Services (LBS) - A Global Strategic Business Report, 10 2010.

[14] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy Characterization for Metropolitan-Scale Wi-Fi Localization. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, pages 233–245, New York, NY, USA, 2005. ACM.

[15] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. Towards Mobile Phone Localization without War-driving. In *Proceedings of the 29th Conference on Information Communications (Infocom)*, pages 2321–2329, 2010.

[16] I. Constandache, X. Bao, M. Azizyan, and R. C. Romit. Did you see Bob?: Human Localization using Mobile Phones. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 149–160, New York, NY, USA, 2010. ACM.

[17] I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. Cox. EnLoc: Energy-Aware Localization Using Mobile Phones. In *Proceeding of the 28th Conference on Computer Communications (Infocom)*, pages 2716 –2720, 2009.

[18] M. P. R. Distributiont. Number of Smartphone Users to Quadruple, Exceeding 1 Billion Worldwide by 2014, 10 2010.

[19] P. Enge and P. Misra. Special Issue on Global Positioning System. *Proceedings of the IEEE*, 87(1):3 –15, 1. 1999.

[20] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-Blog: Sharing and Querying Content through Mobile Phones and Social Participation. In *Proceedings of the 6th international conference on Mobile systems, applications, and services (MobiSys)*, pages 174–186, 2008.

[21] N. Györbíró, A. Fábián, and G. Hományi. An Activity Recognition System for Mobile Phones. In *Mobile Networks and Applications*, volume 14, February 2009.

[22] A. Harter and A. Hopper. A Distributed Location System for the Active Office. *IEEE Network*, 8(1):62 –70, jan. 1994.

[23] K. Kaemarungsi and P. Krishnamurthy. Modeling of Indoor Positioning Systems Based on Location Fingerprinting. In *Proceedings of 23rd the IEEE Conference on Computer Communications (Infocom)*, pages 1012–1022, 2004.

[24] A. M. Khan, Y.-K. Lee, S. Lee, and T.-S. Kim. Human Activity Recognition via an Accelerometer-Enabled-Smartphone Using Kernel Discriminant Analysis. In

Proceedings of the 5th International Conference on Future Information Technology, 5 2010.

[25] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Halel, and S. Shafer. Multi-Camera Multi-Person Tracking for EasyLiving. In *Proceedings of the 3rd IEEE International Workshop on Visual Surveillance*, pages 3–10. IEEE Computer Society, 2000.

[26] 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, 6 2007.

[27] J. R. Kwapisz and G. M. W. S. A. Moor. Activity Recognition using Cell Phone Accelerometers. In *Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data*, 2010.

[28] C.-Y. Lee and J.-J. Lee. Estimation of Walking Behavior Using Accelerometers in Gait Rehabilitation. *International Journal of Human-friendly Welfare Robotic Systems*, 2002.

[29] 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 first ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT '08, pages 73–78, New York, NY, USA, 2008. ACM.

[30] H. Lemelson, S. Kopf, T. King, and W. Effelsberg. Improvements for 802.11-Based Location Fingerprinting Systems. In *33rd Annual IEEE International*

of Computer Software and Applications Conference (COMPSAC) ., volume 1, pages 21 –28, july 2009.

[31] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 11 2007.

[32] Y. Luo, Y. Chen, and O. Hoeber. Wi-fi-based indoor positioning using human-centric collaborative feedback. In *Communications (ICC), 2011 IEEE International Conference on*, 6 2011.

[33] J. Machaj, P. Brida, and R. Piche. Rank based fingerprinting algorithm for indoor positioning. In *Indoor Positioning and Indoor Navigation (IPIN)*, sept. 2011.

[34] 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 (MobiCom)*, pages 261–272, 2009.

[35] MediaPost. Mobile - It's Not Just For Out-Of-Home Anymore. http://www.mediapost.com/, Oct. 2010.

[36] W. Meng, W. Xiao, W. Ni, and L. Xie. Secure and robust Wi-Fi fingerprinting indoor localization. In *Indoor Positioning and Indoor Navigation (IPIN)*, sept. 2011.

[37] E. Miluzzo, J. M. H. Oakley, H. Lu, N. D. Lane, R. A. Peterson, and A. T. Campbell. Evaluating the iPhone as a Mobile Platform for People-Centric Sensing Applications In Proceedings of the International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems (UrbanSense08) . 2008.

[38] M. Mladenov and M. Mock. A Step Counter Service for Java-Enabled Devices Using a Built-in Accelerometer. In *Proceedings of the 1st International Workshop on Context-Aware Middleware and Services: affiliated with the 4th International Conference on Communication System Software and Middleware (COMSWARE 2009)*, pages 1–5, New York, NY, USA, 2009. ACM.

[39] K. Muthukrishnan, M. Lijding, and P. Havinga. Towards Smart Surroundings: Enabling Techniques and Technologies for Localization. In *Proceedings of the First International Workshop on Location and Context Awareness (LOCA)*, 2005.

[40] A. Offstad, E. Nicholas, R. Szcodronski, and R. R. Choudhury. AAMPL: Accelerometer Augmented Mobile Phone Localization. In *Proceedings of the 1st ACM international workshop on mobile entity localization and tracking in GPS-less environments*, pages 13–18, New York, NY, USA, 2008. ACM.

[41] N. B. Priyanthai, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 32–43, New York, NY, USA, 2000. ACM.

[42] A. Quigley and D. West. Proximation: Location-Awareness Though Sensed Proximity and GSM Estimation. In *Lecture Notes in Computer Science*, volume 3479, pages 363–376, 2005.

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

[44] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. 2005.

[45] N. Ravi, P. Shankar, A. Frankel, A. Elgammal, and L. Iftode. Indoor Localization Using Camera Phones. *IEEE Workshop on Mobile Computing Systems and Applications*, 2006.

[46] S. Se, D. Lowe, and J. Little. Mobile Robot Localization and Mapping with Uncertainty Using Scale-Invariant Visual Landmarks. *International Journal of Robotics Research*, 21:735–758, 2002.

[47] S. Y. Seidel and T. S. Rappapor. 914 MHz Path Loss Prediction Models for Indoor Wireless Communications in Multifloored Buildings. *Antennas and Propagation, IEEE Transactions on*, 40(2):207 –217, 2 1992.

[48] A. Serra, D. Carboni, and V. Marotto. Indoor Pedestrian Navigation System using a Modern Smartphone. In *Proceedings of the 12th international conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, 2010.

[49] R. Sim and G. Dudek. Mobile Robot Localization from Learned Landmarks. In *Proceedings of Intelligent Robots and Systems*, 10 1998.

[50] Skyhook. Location Apps Research. http://www.skyhookwireless.com/locationapps/, Oct. 2010.

[51] S. Thrun. Bayesian Landmark Learning for Mobile Robot Localization. *Journal of Machine Learning.*, 33(1):41–76, 10 1998.

[52] S. Thrun. Probabilistic Robotics. *Communications of the ACM*, 45(3):52–57, Mar. 2002.

[53] A. Varshavsky, E. de Lara, J. Hightower, A. LaMarca, and A. LaMarca. GSM Indoor Localization. *Pervasive and Mobile Computing*, 3(6):698–720, 2007.

[54] R. Want, A. Hopper, V. Falco, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

[55] W. Waqar, Y. Chen, and A. Vardy. Exploiting smartphone sensors for indoor positioning: A survey. In *Proceedings of the Newfoundland Conference on Electrical and Computer Engineering*, 2011.

[56] W. Waqar, A. Vardy, and Y. Chen. Motion Modelling using Smartphones for Indoor Mobilephone Positioning. In *Proceedings of the Newfoundland Conference on Electrical and Computer Engineering*, 2011.

[57] W. Waqar, A. Vardy, and Y. Chen. Positioning using Motion Information in RF Reference Frames. In *Proceedings of the Newfoundland Conference on Electrical and Computer Engineering*, 2012.

[58] M. Weiser. The Computer for the 21st Century. *SIGMOBILE Mobile Computing and Communications Review*, 3(3):3–11, 1999.

[59] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, 2008.

[60] O. Woodman and R. Harle. Pedestrian Localisation for Indoor Environments. In *Proceedings of the 10th international conference on Ubiquitous computing*, UbiComp, 2008.

[61] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek. Automatic Step Detection in the Accelerometer Signal. In S. Leonhardt, T. Falck, and P. Mhnen, editors, *4th International Workshop on Wearable and Implantable Body Sensor Networks BSN 2007*, volume 13. Springer.

[62] 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.

[63] D. C. Yuen and B. A. MacDonald. Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison. *IEEE Transactions on Robotics*, 21(2):217–226, 2005.

[64] D. Zhang, F. Xia, Z. Yang, L. Yao, and W. Zhao. Localization Technologies for Indoor Human Tracking. In *Future Information Technology (FutureTech), 2010 5th International Conference on*, 5 2010.

[65] N. Zhao. Full-Featured Pedometer Design Realized with 3-Axis Digital Accelerometer. http://www.analog.com/library/analogdialogue/archives/44-06/pedometer.html, 10 2011.