2015-02-03

Integration of WiFi and MEMS Sensors for Indoor Navigation

Zhuang, Yuan

UNIVERSITY OF CALGARY

Integration of WiFi and MEMS Sensors for Indoor Navigation

by

Yuan Zhuang

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN GEOMATICS ENGINEERING

CALGARY, ALBERTA

JANUARY, 2015

Abstract

The growing demand for indoor navigation applications has promoted the implementation of navigation techniques on handheld devices. An accurate and reliable indoor navigation system hosted on handheld devices would benefit many consumer industries. MEMS (Micro-Electromechanical System) sensors can provide a short-term accurate navigation solution. WiFi-based (Wireless Fidelity) positioning is another potential technology for indoor navigation, which only uses pre-existing WiFi infrastructures and is a good source to aid the MEMS-based navigation solution. However, WiFi positioning requires databases to estimate the user position. The pre-surveys for building and maintaining the WiFi databases make most current WiFi positioning systems are not automatic. Currently, it remains difficult to find an automatic and accurate indoor navigation system on typical handheld devices. However, the complementary characteristics of MEMS sensors and WiFi offer an efficient integration for indoor navigation applications.

Two automatic WiFi Positioning Services (WPSs) based on trilateration and fingerprinting are investigated in this research, which both consist of the background survey service and WiFi positioning service. Both WPSs provide WiFi positioning solutions, with no cost to build and to maintain WiFi databases. This removes the limitations that most current WPSs require time-consuming and labor-intensive pre-surveys to build the databases. Different approaches are investigated to improve the accuracy of both the WiFi databases and the user's positions in indoor environments. The developed two automatic WPSs are also compared.

An innovative MEMS navigation solution, based on motion constraints and the integration of INS (Inertial Navigation System) and PDR (Pedestrian Dead Reckoning), is built on handheld devices. LC (Loosely-coupled) integration and TC (Tightly-coupled) integration are implemented for WiFi and MEMS sensors to further limit the drifts of MEMS sensors. The navigation performances of

PDR, INS, the PDR/INS-integrated MEMS solution, the LC integration solution, and the TC integration solution are compared in this research. The test results also show its average positioning error of TC integration in various trajectories is 0.01% of INS, 10.38% of PDR, 32.11% of the developed MEMS solution, and 64.58% of LC integration. This developed TC integration solution can be used in both environments with dense and sparse deployments of WiFi APs (Access Points).

Acknowledgements

I would like to thank Dr. Naser El-Sheimy for all his support, guidance, and continuous encouragement during my studies. He provided the opportunities for me to become a strong research engineer in the navigation field. It is my honor to have him as my teacher, supervisor and mentor. I would also like to extend my gratitude to Dr. Kyle O'Keefe for his constant support and valuable feedback on my research. I would also like to thank Dr. Aboelmagd Noureldin for all his kind assistance and constructive suggestions. I am very graceful to have all of you as my supervisory committee.

I would also like to thank to all the team members at the Trusted Positioning Inc. for valuable discussion and technical support especially Dr. Zainab Syed. I would like to extend my thanks to all my colleagues and roommates in the Mobile Multi-Sensors Systems (MMSS) research group (Dr. Yigiter Yuksel, Dr. Xing Zhao, Dr. Mohamed El-Habiby, Dr. Abdelrahman Ali, Dr. Bassem Sheta, Dr. Ahmed Shawky, Mr. Siddharth, Dr. Adel Moussa, Dr. Sara Saeedi, Mr. Naif Alsubaie, Dr. Hsiu-Wen Chang, Mr. Hussein Sahli, Mr. Amr Al-Hamad, Mr. Navid Mostofi, Dr. Daihong Cao, Mr. Hani Mohammed, Mr. You Li, Mr. Haiyu Lan, and Mrs. Chunyang Yu) for their friendship, support, and help in the field tests and valuable discussions. Dr. Hsiu-Wen Chang is specially thanked for letting me have the attitude of "you can do anything".

This work was supported in part by research funds from Trusted Positioning Inc., MITACS, and TECTERRA Commercialization and Research Centre.

Finally, and most importantly, I would like to thank my parents and my soul mate, for their unconditional love, encouragement, and understanding through all of my years of study. This work would not have been possible without their support.

Dedication

To My Parents
For Everything They Do for Me

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Acronyms/Abbreviations	Definition
2D	Two Dimensional
3D	Three Dimensional
AKF	Adaptive Kalman Filter
AP	Access Point
b-frame	Body Frame
COM	Center of Mass
DOP	Dilution of Precision
DPSLAM	Distributed Particle SLAM
DR	Dead Reckoning
FEKF	Fingerprint Extended Kalman Filter
EKF	Extended Kalman Filter
ENU	East, North, Up
FEKFSLAM	Fingerprint Extended Kalman Filter SLAM
GNSS	Global Navigation Satellite System
GP-LVM	Gaussian Processes Latent Variables Model
GPS	Global Positioning System
Gyro	Gyroscope
i-frame	Inertial Frame
INS	Inertial Navigation System
KF	Kalman Filter

KNN	K-Nearest Neighbour
LBS	Location Based Services
LC	Loosely-Coupled
LLH	Latitude, Longitude, Height
LSQ	Least Squares
MAP	Maximum a Posteriori
MEMS	Micro-Electromechanical System
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimator
MMSE	Minimum Mean Square Error
NED	North, East, Down
n-frame	Navigation Frame
NHC	Non-Holonomic Constraints
OS	Operating System
PDA	Personal Digital Assistants
PDR	Pedestrian Dead Reckoning
PP	Propagation Parameters
PVA	Position, Velocity, Attitude
RF	Radio Frequency
RFID	Radio Frequency Identification
RMS	Root Mean Square
RSS	Received Signal Strength
SLAM	Simultaneous Localization and Mapping

SNR	Signal to Noise Ratio
STD	Standard Deviation
TC	Tightly-Coupled
T-PN	Trusted Positioning Navigator
USBL	Ultra-Short Baseline
UWB	Ultra-Wide Band
WiFi	Wireless Fidelity
WKNN	Weighted K-Nearest Neighbour
WLAN	Wireless Local Area Networks
WPS	WiFi Positioning System
ZARU	Zero Angular Rate Update
ZUPT	Zero Velocity Update

Chapter One: **Introduction**

1.1 Background and Problem Statement

The rapid development and improvement of handheld devices, such as smartphones and tablets, has enabled them to become powerful tools for navigation applications (Kim et al. 2013; Yohan and Hojung 2011). Modern handheld devices are widely used as platforms for navigation because they have sophisticated and powerful microprocessors, efficient operating systems, and embedded multi-sensors (Zhuang et al. 2013b). The microprocessors and operating systems ensure fast computation for navigation applications, whereas embedded multi-sensors guarantee sufficient data to support the design of navigation algorithms. The growing demand for navigation applications, especially indoors, has also promoted the implementation of navigation techniques on handheld devices. Accurate and reliable indoor navigation system hosted on handheld devices would benefit many consumer industries including health care, Location Based Services (LBS), emergency services, tourism, and personnel management (Renaudin et al. 2007).

To provide indoor navigation solutions, there are several potential technologies available such as Wireless Fidelity (WiFi), Global Positioning System (GPS), and inertial sensors-based relative navigation, etc. GPS, when signal available, is the most popular and accurate navigation system (Kaplan and Hegarty 2006). However, GPS cannot provide a reliable indoor navigation solution because its signals are degraded by ceilings, walls, and other objects. Therefore, other technologies have been developed to compensate for the limitations of GPS, such as Radio Frequency Identification (RFID) (Cardullo and Parks 1973), Ultra-Wide Band (UWB) (Siwiak 2001), Micro-Electromechanical Systems (MEMS) multi-sensors (Mohamed 1999) (Zhuang et al. 2013a), and Wireless Local Area Networks (WLAN) (Chen et al. 2012). Specifically, RFID and UWB require

dedicated infrastructure and special devices to detect signals for positioning, and can provide accurate positioning solutions. On the other hand, in most current handheld devices, MEMS sensors, such as accelerometers, gyroscopes, magnetometers, and barometers, provide navigation solutions without any dedicated infrastructure. However, the accuracy of the MEMS sensors' navigation solution will decrease with time due to their drift characteristics (Zhuang et al. 2013a) (El-Sheimy 2006).

WiFi-based positioning is another potential technology for indoor navigation because it only uses pre-existing WiFi infrastructure. WiFi positioning errors do not accumulate with time which makes WiFi an excellent source to aid the standalone navigation solution based on MEMS sensors (Yunye et al. 2013). Currently, there are two RSS-based (Received Signal Strength) WiFi localization techniques: trilateration and fingerprinting (Hui et al. 2007). Both of these technologies require special databases to estimate the user position. In traditional approaches, professional surveyors are hired to build and maintain the databases. A radio map database is required for fingerprinting, where the RSSs of available Access Points (APs) are mapped to absolute positions. Pre-survey is also needed to build the database of the propagation parameters (PPs) and AP locations for trilateration. Although some approaches have been proposed to reduce the effort to construct WiFi databases, these approaches still require many professional surveyors, especially for large areas. Pre-survey is a labor-intensive and time-consuming process conducted by professional surveyors. Therefore, most current WiFi positioning system are not automatic.

Some crowdsourcing-based systems have made indoor positioning more practical. However, they still suffer from various limitations, such as needing a floor plan or GPS, being suitable only for specific indoor environments, and only implementing a simple MEMS sensor solution. Thus,

currently, it remains difficult to find an automatic and accurate indoor navigation system on typical handheld devices without special hardware or infrastructures. However, it is expected that the cooperation of MEMS sensors and WiFi is an efficient approach for indoor navigation applications. In this thesis, the focus primarily is on the implementation of WiFi and MEMS cooperated systems on handheld devices because the pedestrian navigation services implemented on handheld devices are low-cost, user friendly, and do not require additional hardware.

1.2 Review of Existing Literature

1.2.1 Reduce Labor for Building WiFi Databases

To ensure WiFi positioning is more practical, much work has been done to reduce the labor-intensive and time-consuming task of building the databases for both trilateration (Cheng et al. 2005) (Skyhook 2014) (Yu 2012) and fingerprinting (Cheng et al. 2005) (Yungeun et al. 2012) (Bolliger et al. 2009) (Nguyen and Zhang 2013). Fingerprinting-based research is given first. A system is proposed in (Cheng et al. 2005) to reduce the cost of offline training by automatically collecting WiFi fingerprints with the help of vehicles equipped with GNSS (Global Navigation Satellite System) receivers. However, this system is used for outdoors, and is not suitable for indoor applications. Another concept is discussed in (Bolliger et al. 2009) whereby normal users, not professional surveyors, update fingerprints to the radio map. But, this is also not an automatic system because it requires the active participation of users to update fingerprints. An inertial sensors based system is proposed in (Yungeun et al. 2012) for the offline training phase. However, the inertial sensor's navigation solution in this system is only based on simple dead reckoning by using accelerometers and magnetometers, which is not accurate and robust. Second, we summarize the algorithms for building the database containing AP locations for trilateration. In PlaceLab

(Cheng et al. 2005), AP locations are computed through the use of averaging and weighted averaging of positions derived from the measurement points collected through "war-driving". However, large estimation errors can result from measurement points with poor geometrical distribution. The research given in (Tsui et al. 2010) and Skyhook (Skyhook 2014) also uses "war-driving" to collect AP locations. Moreover, research provided in (Yu 2012) estimates the path loss exponent and a constant parameter of the propagation model through rigorous testing. Least squares (LSQ) is then used to estimate AP locations. Yet, the challenge of this method is that the pre-surveyed parameters are not suitable for the estimation of AP locations when the environment has changed.

1.2.2 Crowdsourcing-Based Systems

Until now, several crowdsourcing-based systems have been proposed for indoor navigation, see for example (Chintalapudi et al. 2010) (Rai et al. 2012) (Wang et al. 2012) (Yang et al. 2012) (Shen et al. 2013). The work in (Chintalapudi et al. 2010) proposes the EZ localization algorithm, which does not require any pre-deployment effort, infrastructure support, priori knowledge about WiFi APs, or active user participation. However, EZ's reliance on "occasional GPS fixes" in indoor environments could be problematic. Research conducted in (Rai et al. 2012) proposes the "Zee" system which has zero-effort crowdsourcing for indoor locations. Zee requires a map showing the pathways and barriers to filter out infeasible locations over time and converge on the true location by using the idea that a user cannot walk through a wall or other barrier marked on the map. However, this map is not available in many real-world cases. Also, Zee uses magnetometers, rather than gyroscopes, for calculating direction, which is usually affected by the indoor environment. Unlike the Zee, UnLoc, an unsupervised indoor localization scheme that

bypasses the need for war-driving, is proposed in the work of (Wang et al. 2012). The key idea of UnLoc is to improve the dead-reckoning-based sensor solution by using seed and organic landmarks. A floor plan or GPS is required in this system to find the location of seed landmarks. However, the location of seed landmarks could be questionable if a floor plan and GPS are not available. Another work in (Yang et al. 2012) presents the LiFS, an indoor localization system, which constructs the radio map with the help of a floor plan and sensors in smartphones. The building of the radio map is easy and rapid since little human intervention is needed. LiFS works well in buildings where the corridor connects all other office rooms that are on both sides of the corridor. However, LiFS may fail in large open environments, where users' movements are difficult to analyze. Furthermore, similar to Zee, LiFS needs a floor plan to build the database, which may not always be available. Unlike LiFS, (Shen et al. 2013) presents Walkie-Markie – a crowdsourcing-capable pathway mapping system that leverages the sensor-equipped mobile phones of ordinary pedestrians and to build indoor pathway maps without any a priori knowledge of the building. Central to Walkie-Markie is a novel exploitation of the WiFi infrastructure to define landmarks (WiFi-Marks) to fuse crowdsourced user trajectories obtained from inertial sensors on users' mobile phones. The main limitation of Walkie-Markie is that it does not work well in wide pathways where WiFi-Mark detection and clustering will deteriorate if users have a wide choice of where to walk.

In summary, while these crowdsourcing-based systems have made indoor positioning more practical than before, they still suffer from various limitations, which need a floor plan (Rai et al. 2012) (Wang et al. 2012) (Yang et al. 2012) or GPS (Chintalapudi et al. 2010) (Wang et al. 2012); are suitable only for specific indoor environments (Yang et al. 2012) (Shen et al. 2013); and only

implement a simple MEMS sensor solution (Rai et al. 2012) (Wang et al. 2012) (Yang et al. 2012) (Shen et al. 2013). Therefore, the proposed system aims at reducing these limitations.

1.2.3 WiFi SLAM

WiFi SLAM (Simultaneous Localization and Mapping) is another group of algorithms (Ferris et al. 2007) (Bruno and Robertson 2011) (Faragher and Harle 2013) (Huang et al. 2011) for localization and WiFi information mapping (radio map and AP location). Researchers in (Ferris et al. 2007) implemented a WiFi SLAM system by using the GP-LVM (Gaussian Processes Latent Variables Model). More specifically, a WiFi radio map was generated by using GP-LVM to extrapolate from the existing fingerprints. However, this system is limited by its large computation load when processing large sets of data. Another WiFi SLAM algorithm is provided in (Huang et al. 2011), which builds the WiFi radio map based on GraphSLAM. The WiSLAM algorithm for improving FootSLAM with WiFi is provided in (Bruno and Robertson 2011). Yet, one drawback of this algorithm is that the path loss exponent is set to two when using the propagation model. Research in (Faragher and Harle 2013) proposes a smartSLAM scheme which contains PDR (Pedestrian Dead Reckoning), FEKF (Fingerprint Extended Kalman Filter), FEKFSLAM (Fingerprint Extended Kalman Filter SLAM), and DPSLAM (Distributed Particle SLAM). It also provides the process of building a WiFi radio map if it is not readily available. The large computation load of WiFi SLAM algorithms (Ferris et al. 2007) (Bruno and Robertson 2011) (Faragher and Harle 2013) (Huang et al. 2011) reduces the efficiency of microprocessors and increases battery consumption, which makes these algorithms unsuitable for implementation in handheld devices. If WiFi SLAM algorithms are implemented in the server, real-time transmission of high-rate sensor data to the server will increase the battery consumption of the devices.

1.2.4 Integrated Technologies for WiFi and MEMS Sensors

Most research has focused on the integration of WiFi and body-mounted MEMS sensors (Chai et al. 2012; Evennou and Marx 2006; Frank et al. 2009). An indoor positioning system for pedestrians, combing WiFi fingerprinting with foot-mounted inertial and magnetometer sensors, is proposed in (Frank et al. 2009). However, foot-mounted systems are not as convenient as handheld devices for pedestrians, and the requirement of the pre-survey makes WiFi fingerprinting impractical for a large area. An advanced integration of WiFi and INS (Inertial Navigation System), based on a particle filter, is proposed in (Evennou and Marx 2006). Nevertheless, the particle filter is not suitable for handheld devices such as smartphones, due to its large computation load. If particle filter algorithms are implemented in the server, real-time transmission of high-rate sensor data to the server will increase the battery consumption of the devices. Furthermore, by using the AKF (adaptive Kalman filter), a PDR/WiFi/barometer integrated system is proposed in (Chai et al. 2012). However, this system is also based on WiFi fingerprinting and foot-mounted sensors. Moreover, a maximum-likelihood-based fusion algorithm that integrates the PDR and WiFi fingerprinting is proposed in (Chen et al. 2014). The algorithm was implemented in smartphones which made the system practical other than the pre-survey for fingerprinting.

In addition, almost all current WiFi/MEMS integrations are loosely-coupled (LC) integrations, which means the integration is based on a MEMS navigation solution and WiFi position solution. On the other hand, tightly-coupled (TC) integration has been used for the integration of inertial sensors with GPS, RFID and USBL (Ultra-Short Baseline) (George and Sukkarieh 2005; Li et al. 2006b; Morgado et al. 2006; Ruiz et al. 2012; Wendel and Trommer 2004; Yi and Grejner-

Brzezinska 2006). Therefore, a TC integration for WiFi and MEMS sensors is proposed in this thesis to improve the performance of indoor navigation.

1.3 Research Objectives

The main objective of this research is to develop an automatic and seamless indoor navigation solution on handheld devices through the cooperation of WiFi and MEMS sensors. The accuracy objective in the proposed solution is to achieve the best accuracy for indoor navigation based on current hardware of handheld devices (e.g. smartphones and tablets). However, this accuracy objective has a lower priority than the automatic and seamless characteristics of the navigation system. Current handheld indoor navigation systems based on WiFi and MEMS sensors usually work in one mode whereby WiFi helps MEMS sensors to limit the drifts, or in the other mode whereby MEMS sensors help WiFi build the databases. But, systems seldom work in both modes and are not really cooperative. The proposed system in this thesis works in both modes and aims to provide an automatic and seamless indoor navigation solution. To achieve the main purpose, several important implementation and development issues must be addressed.

1. ***Design and implementation of an automatic trilateration-based WPS (WiFi Positioning System):*** Trilateration requires current RSS values, propagation parameters, and AP locations to estimate the user's position. A pre-survey is usually required to build the database, which consists of AP locations and propagation parameters. Thus, to implement an automatic trilateration-based WPS, the following two issues will be investigated:
    - ✓ ***Crowdsourcing-based WiFi database building:*** The pre-survey for the trilateration-based database is time-consuming and labor-intensive, which makes most current WiFi positioning systems impractical and not automatic. To automatically build the

trilateration-based database by using crowdsourcing and MEMS-based navigation solution, several issues need to be investigated as follows: (1) How do we estimate the AP locations and propagation parameters from the MEMS solution and RSSs? (2) How can we remove unreliable estimates from the database? (3) How do we automatically build the database for trilateration through crowdsourcing? (4) What is the accuracy of AP locations and propagation parameters in the database?

- ✓ ***WiFi positioning:*** We also need to investigate the issues for WiFi positioning by using the trilateration-based database as follows: (1) How do we estimate user's position by using the crowdsourcing-based database? (2) How can we remove unreliable estimates?

2. ***Design and implementation of an automatic fingerprinting-based WPS:*** Fingerprinting requires current RSS values and a radio map database to estimate the user's position, and a pre-survey is usually required to build the radio map database. Thus, to implement an automatic fingerprinting-based WPS, the following two issues will be investigated:
   - ✓ ***Crowdsourcing-based WiFi database building:*** Similar to the automatic trilateration-based WPS, several issues need to be investigated to automatically build the fingerprinting-based database by using crowdsourcing and MEMS-based navigation solution as follows: (1) How can we generate fingerprints from the MEMS solution and RSSs? (2) How do we remove unreliable fingerprints from the database? (3) How can we automatically build the database for fingerprinting through crowdsourcing?
   - ✓ ***WiFi positioning:*** We also need to investigate the issues for WiFi positioning by using the fingerprinting-based database as follows: (1) How to estimate user's position by using the crowdsourcing-based database? (2) How to remove unreliable estimates?

3. ***Comparison of two automatic WPSs:*** After the two automatic WPSs are implemented, they will be compared in terms of: accuracy, memory cost, and implementation complexity.
4. ***Design and implementation of the WiFi/MEMS integration for indoor navigation:*** There are different approaches for implementing a MEMS-based navigation solution and a WiFi/MEMS integrated navigation solution. Therefore, the following research questions will be addressed: How do we implement an advanced MEMS solution to reduce the drifts? Also, what are the navigation performances of loosely-coupled and tightly-coupled WiFi/MEMS integrations?

Complete answers to these research questions will be provided in this thesis including some tests and analysis based on test results.

1.4 Thesis Outline

This thesis covers the design and implementation issues of an automatic and seamless indoor navigation solution on handheld devices through the cooperation of WiFi and MEMS sensors. The thesis consists of six chapters, and the outline of chapters two through six is as follows.

Chapter 2 covers the necessary background information for the development and analysis of an indoor navigation system, and typical technologies for MEMS-based and WiFi-based navigation are summarized. The integrated navigation solutions using MEMS sensors and wireless signals are discussed, and three estimation approaches for navigation applications are presented in this chapter as follows: KF (Kalman Filter), EKF (Extended Kalman Filter), and nonlinear LSQ (Least Squares).

Chapter 3 focuses on the issues of design and implementation of an automatic trilateration-based WPS. The design, implementation, and performance evaluation of a trilateration-based automatic

WPS is presented. In this chapter, the overview of the proposed system is discussed as well as the T-PN (Trusted Positioning Navigator) solution. The developed algorithms for measurement optimization, AP localization, PPs estimation, and autonomous crowdsourcing are discussed in detail. Background survey service and WiFi positioning service are also investigated and demonstrated, followed by test results and performance analyses.

Chapter 4 deals with the issues of design and implementation of an automatic fingerprinting-based WPS and the comparison of two automatic WPSs. The design, implementation, and performance evaluation of a fingerprinting-based automatic WPS is discussed. In this chapter, background survey service and WiFi positioning service are investigated and analyzed. Algorithms for automatic radio map database generation and improved fingerprinting-based WiFi positioning are demonstrated, and their performances are evaluated through the field tests. The proposed automatic fingerprinting-based WPS is also compared with the automatic trilateration-based WPS.

Chapter 5 focuses on the issues of design and implementation of the WiFi/MEMS integration for indoor navigation. An innovative algorithm, based on the integration of INS and PDR, is proposed for the MEMS-based navigation solution. Two integrated schemes for MEMS and WiFi, LC integration and TC integration, are proposed to improve the accuracy of the indoor navigation solution. The navigation performances of PDR, INS, PDR/INS-integrated MEMS solution, LC integration solution, and TC integration solution are also evaluated and compared in this chapter.

Chapter 6 summarizes the achieved work of this thesis, concludes the results of this research, and gives recommendations for future research to improve the proposed system. Figure 1-1 shows the outline of the thesis and topic classification corresponding to the issues listed in Section 1.3.

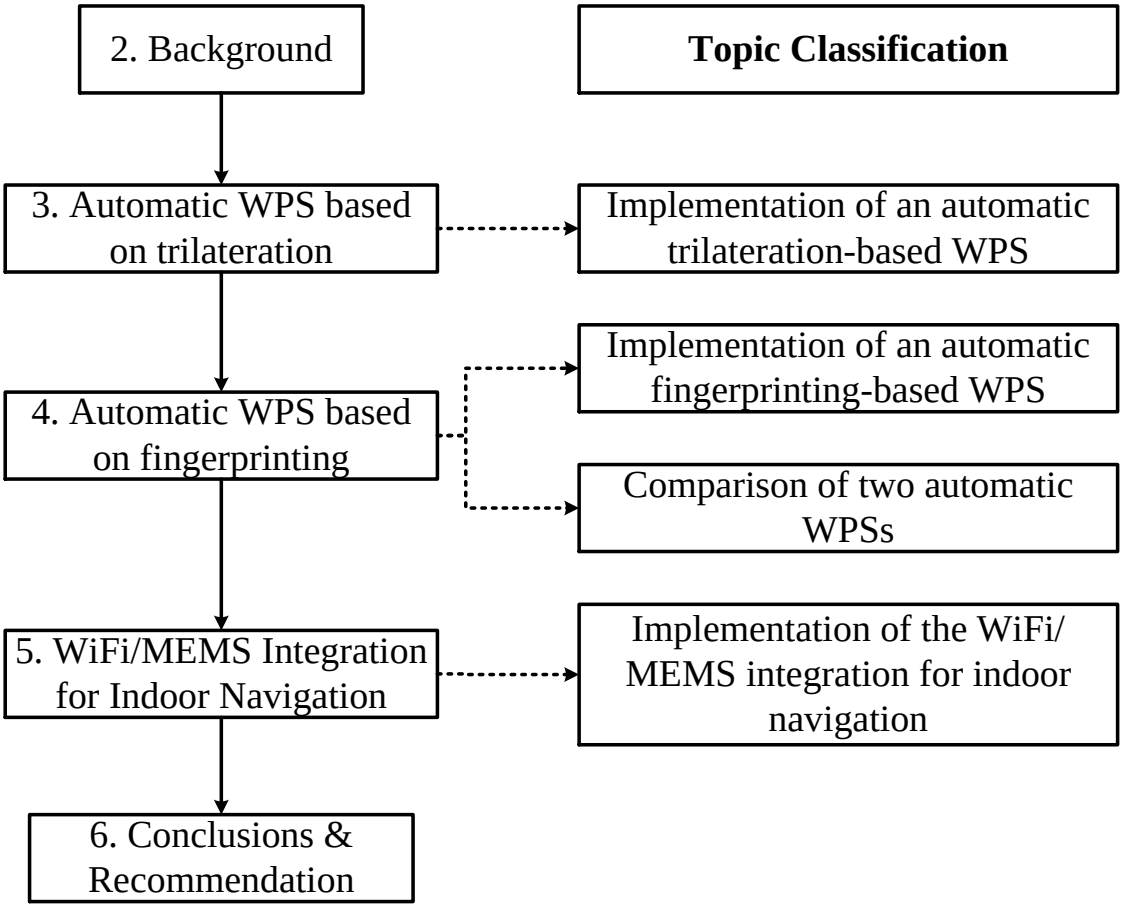


Figure 1-1 Thesis outline.

Chapter Two: **Background**

This chapter will cover the background information for the development and analysis of an automatic indoor navigation system based on the cooperation of MEMS sensors and WiFi. Since MEMS sensors play a significant role in indoor navigation, Section 2.1 summarizes the commonly used processes for implementing the MEMS-based navigation solution, which includes the INS solution, the PDR solution, and the motion constraints. Section 2.1 also describes the problems of current MEMS solutions. Section 2.2 discusses two typical implementations (trilateration and fingerprinting) for the WiFi-based navigation solution along with their limitations. The integrated navigation solutions using MEMS sensors and wireless signals are given in Section 2.3. Finally, Section 2.4 describes three estimation approaches for navigation applications: Kalman filter, extended Kalman filter, and nonlinear least squares.

2.1 MEMS Solution for Indoor Navigation

Currently, there are two different approaches to implement inertial sensors-based pedestrian navigation solution: INS and PDR. In the first approach, raw inertial sensor data is put to the INS mechanization equations to calculate the user's navigation information. INS can provide 3D (Three dimensional) position, velocity, and attitude (PVA) information. However, the navigation error based on this approach increases rapidly with time due to the MEMS errors and the integrations used in the mechanization (Titterton and Weston 2004). On the other hand, PDR has four main procedures: step detection, step/stride length estimation, heading estimation, and 2D (Two dimensional) position calculation. In PDR, navigation solution errors are proportional to the distance traveled, and not to the time (Jahyoung and Hojung 2011). Besides these two approaches, motion constraints are also often used in MEMS-based navigation solutions. This section describes

three motion constraints used for pedestrian navigation: NHC (Non-holonomic constraints), ZUPT (Zero velocity update), and ZARU (Zero angular rate update). In the end, problems of current MEMS-based navigation solutions are discussed.

2.1.1 Reference Frames

Definitions of reference frames, which include navigation frame, body frame (sensor frame), and vehicle frame (pedestrian frame), are given below.

The navigation frame (n-frame, north-east-down NED in this thesis) is a local geodetic frame which has its origin coinciding with that of the sensor frame, its x-axis pointing towards the geodetic north, its z-axis orthogonal to the reference ellipsoid pointing down, and its y-axis completing a right-handed orthogonal frame.

The body frame (b-frame) is the frame in which accelerations and angular rates are generated from the accelerometers and gyroscopes. In this thesis, the sensor frame (s-frame) is the same as the b-frame, and the roll, pitch, and heading are defined for the handheld device (or the IMU), but not for the pedestrian. This is appropriate because the pedestrian usually has a very small roll and pitch when walking or being static, while the handheld device may have a large roll and pitch. However, the pedestrian heading is assumed to be the same as the heading of the handheld device (heading misalignment is zero degree), which is often satisfied when holding the device for navigation. Several researches have been conducted to estimate the heading misalignment when it is not zero degree. However, this is not the focus of this thesis.

The vehicle frame (v-frame) is an orthogonal forward-transversal-down axis set. In this thesis, the vehicle frame can also be called the pedestrian frame because the proposed navigation systems is

used for pedestrians. The frame is required because the b-frame is usually not parallel to the v-frame in the handheld pedestrian navigation applications.

2.1.2 INS Solution

An inertial-sensors-based navigation system usually consists of three accelerometers and three gyroscopes. Accelerometers sense the specific force f^b in the body frame, whereas gyroscopes measure the angular velocity ω_{ib}^b in the body frame, which is the rotation of the body frame with respect to the inertial frame, measured in the body frame. The specific force measurements f^b are used to compute the body acceleration, which is later used in estimating position differences after double integration with respect to time. The angular velocity measurements ω_{ib}^b are used to calculate the angular differences of the body relative to its initial orientation after integration in time (Titterton and Weston 2004). In summary, INS mechanization equations use specific force measurements f^b and angular velocity measurements ω_{ib}^b to compute the PVA information for the object (Titterton and Weston 2004), which is given as follows (Aggarwal et al. 2010).

$$\begin{bmatrix} \dot{r}^n \\ \dot{v}^n \\ \dot{C}_b^n \end{bmatrix} = \begin{bmatrix} D^{-1}v^n \\ C_b^n f^b - (2\Omega_{ie}^n + \Omega_{en}^n)v^n + g^n \\ C_b^n(\Omega_{ib}^b - \Omega_{in}^b) \end{bmatrix} \tag{2-1}$$

where

$$\dot{r}^n = \begin{pmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{pmatrix} = \begin{pmatrix} \frac{1}{M+h} & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} v_N \\ v_E \\ v_D \end{pmatrix} = D^{-1}v^n \tag{2-2}$$

$r^n = \begin{bmatrix} \phi & \lambda & h \end{bmatrix}^T$ is the position vector (latitude, longitude, and height). $v^n = \begin{bmatrix} v_N & v_E & v_D \end{bmatrix}^T$ is the velocity vector in the navigation frame. C_b^n is the transformation matrix from the body frame to the navigation frame as a function of attitude components. g^n is the gravity vector in the navigation frame. $2\Omega_{ie}^n + \Omega_{en}^n$ is the skew-symmetric matrix of the angular velocities $2\omega_{ie}^n + \omega_{en}^n$. ω_{ie}^n is the angular velocity of e-frame with respect to i-frame as measured in the navigation frame and ω_{en}^n is the angular velocity of the navigation frame with respect to the e-frame as measured in the navigation frame. $2\omega_{ie}^n + \omega_{en}^n$ can be calculated as follows.

$$2\omega_{ie}^n + \omega_{en}^n = 2\begin{pmatrix} \omega^e \cos\varphi \\ 0 \\ \omega^e \sin\varphi \end{pmatrix} + \begin{pmatrix} \dfrac{v_E}{N+h} \\ \dfrac{-v_N}{M+h} \\ \dfrac{-v_E \tan\varphi}{N+h} \end{pmatrix} = \begin{pmatrix} \dfrac{v_E}{N+h} + 2\omega^e \cos\varphi \\ \dfrac{-v_N}{M+h} \\ \dfrac{-v_E \tan\varphi}{N+h} + 2\omega^e \sin\varphi \end{pmatrix} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{2-3}$$

where ω^e is the earth rotation rate. Therefore, the $2\Omega_{ie}^n + \Omega_{en}^n$ in (2-1) can be expressed as follows.

$$2\Omega_{ie}^n + \Omega_{en}^n = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \tag{2-4}$$

where Ω_{in}^b is the skew-symmetric matrix of the rotation vector ω_{in}^b, from the navigation frame to the inertial frame measured in the body frame. ω_{in}^b can be given by the following equation.

$$\omega_{in}^{b} = C_{n}^{b}\begin{pmatrix} \frac{v_E}{N+h} + \omega^e \cos\varphi \\ \frac{-v_N}{M+h} \\ \frac{-v_E \tan\varphi}{N+h} + \omega^e \sin\varphi \end{pmatrix} \tag{2-5}$$

The INS mechanization algorithm is summarized in Figure 2-1 (El-Sheimy 2006).

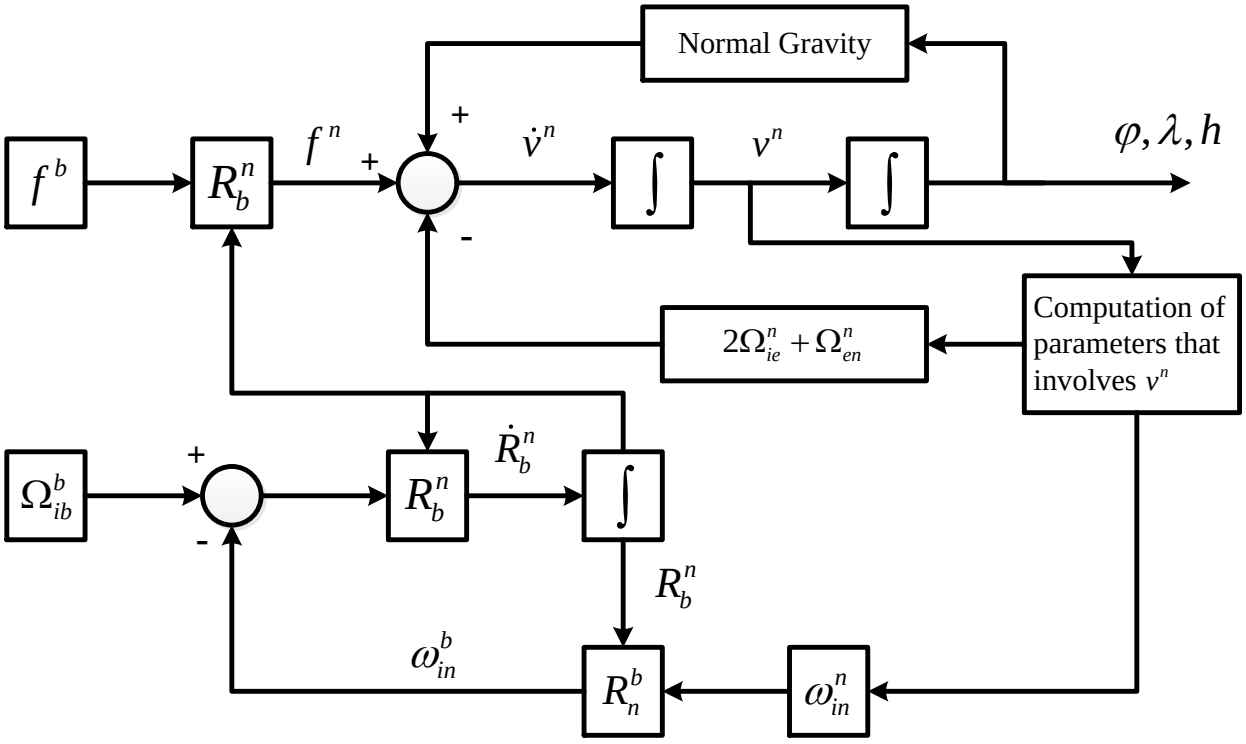


Figure 2-1 INS mechanization algorithm.

2.1.3 PDR Solution

PDR determines the current position of the pedestrian from the knowledge of the previous position and the measurements of the motion direction and traveled distance. The PDR algorithm usually

includes step detection, step length estimation, heading estimation, and PDR mechanization (Zhuang et al. 2013a).

First, steps are usually detected by means of the cycle pattern of the acceleration norm. Currently, peak detection, zero crossing, auto/cross correlation and spectral analysis are typical approaches for the step detection (Harle 2013). Because stride is associated with sharp changes to the vertical acceleration, peak detection can be used to find the strikes. Zero crossing is a simpler way to detect steps by monitoring the acceleration value. Another step detection approach is based on the strong periodicity in the sensor data from the periodic nature of walking. The steps can be extracted by the autocorrelation of a sequence of sensor data. If a sample sequence of sensor data for a step has previously been collected, steps also can be extracted by the cross correlation between the collected sensor data and this sample data. Spectral analysis computes the frequency spectrum of the cyclic data and identifies strong peaks as step frequencies. These approaches are also summarized in Table 2-1. For more details about step detection, please refer to (Harle 2013). In this thesis, the peak detection is used for step detection.

Table 2-1 Step detection algorithms

Algorithms	Basic Idea
Peak Detection	Detect peaks of acceleration norms
Zero Crossings	Detect zero crossings of acceleration norms
Auto/Cross Correlation	Mean-adjusted auto/cross correlation
Spectral Analysis	Identify strong peaks of spectrum as step frequencies

The step length estimation is used to estimate the moving distance of the pedestrian at each step. Different approaches have been proposed for estimating the step length. For the foot-mounted MEMS sensors, INS can provide the information of step length (Alvarez et al. 2006; Jimenez et al. 2009). However, INS solution drifts very fast when using small-size MEMS sensors. To improve the accuracy of step length estimation, ZUPT (Zero Velocity Update) in the stance phase is used to attenuate the bias of the accelerometers. This approach is not suitable for handheld devices because the stance phase cannot be detected in this case. If the device is mounted at the COM (Center of Mass) of the user, an inverted pendulum model can be used to calculate the step length by using the user's leg length and the vertical displacement of the COM during one step (Jahn et al. 2010; Weinberg 2002). The need for a specific mounted place also makes this model unsuitable for handheld devices. Another group of methods estimate the step length by combining the step frequency, acceleration variance, vertical velocity, etc. The combination can be implemented by using difference models (Kappi et al. 2001; Ladetto 2000; Lee et al. 2011; Shin and Park 2011). Empirical models are also efficient approaches to estimate the step length. The models are built from sufficient experimental data (Alvarez et al. 2006; Jahn et al. 2010; Kim et al. 2004). In this thesis, the model proposed in (Weinberg 2002) is used for step length estimation, which assumes the step length is proportional to the vertical movement of the human hip. The largest difference of the vertical acceleration at each step is used to calculate vertical movement. The equation for step length estimation is expressed as:

$$SL = \sqrt[4]{a_{z\max} - a_{z\min}} \cdot K \tag{2-6}$$

where $a_{z\max}$ is the maximum value of the vertical acceleration a_z, $a_{z\min}$ is the minimum value of a_z, and K is a calibrated constant parameter. When using (2-6) to estimate the step length of a user, the device is assumed to be levelled. Therefore, the vertical direction is the z-axis of the body frame of the device.

There are two main ways to estimate the moving direction of a person: using gyroscopes and magnetometers. Gyroscopes provide a relative heading. Therefore, an initial heading should be derived from GPS velocity or provided by the user. It is accurate only for short term due to the accumulated error as a function of time. However, compared to magnetometers which can be easily disturbed by the environment, it will not suffer from sudden changes in the heading estimation. The magnetometers provide long term absolute heading. However, its main problem is the effect of external disturbance. Gyroscopes can be used to detect external disturbance using Equation (2-7) (Ladetto et al. 2001)

$$\begin{aligned} &|\omega_G - \omega_C| > th \\ &\omega_C = \frac{\psi(t_{k+1}) - \psi(t_k)}{t_{k+1} - t_k} \end{aligned} \tag{2-7}$$

where ω_G is the derived angle rate from magnetometer measurements, ω_C is the angle rate of the gyroscopes, th is the threshold selected at the calibration process, and ψ is the magnetic heading from magnetometer measurements. The details of these two approaches are depicted in Table 2-2.

Table 2-2 Summary of two approaches for heading estimation

Heading	Mag-Based	Gyro-Based
Theory	$H_M = \text{atan}_2(\frac{H_y}{H_x})$	$H_{G(k+1)} = H_{G(k)} + \int_{t_k}^{t_{k+1}} \omega(t)dt$
Pros	Absolute heading Long term accuracy	Less disturbance Short term accuracy
Cons	Unpredictable External disturbance	Drift Relative heading
Calibration	Hard	Easy
Cost	Low	High

Even if using Equation (2-7), the magnetic heading does not work well indoors due to complex indoor environments. Therefore, gyroscopes serve as the main source for pedestrian heading estimation in this research. In Table 2-2, the heading estimation equations for magnetometers and gyroscopes are based on the assumption that the handheld device is levelled. This assumption is right when the user holds the device in compass mode. If this assumption is not valid in some cases, the device needs to be leveled down, and heading will be re-estimated. With the assumption that the handheld device is level (roll and pitch are zero degrees), the pedestrian's moving direction is estimated by the integration of the vertical gyroscope. Finally, the PDR mechanization is given by

$$\begin{cases} E_k = E_{k-1} + \hat{s}_{(k-1,k)} \cdot \sin(\hat{H}_k) \\ N_k = N_{k-1} + \hat{s}_{(k-1,k)} \cdot \cos(\hat{H}_k) \end{cases} \tag{2-8}$$

where (E_{k-1}, N_{k-1}) and (E_k, N_k) are positions at epoch $k-1$ and epoch k, $\hat{s}_{(k-1,k)}$ and $\hat{H}_k$ are estimated step length and heading at epoch k. A simple description for PDR is shown in Figure 2-2.

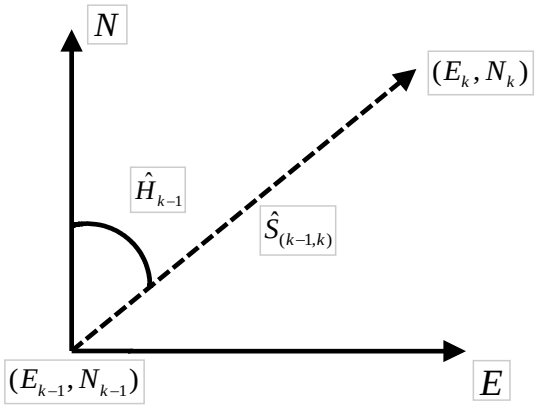


Figure 2-2 PDR algorithm.

2.1.4 Motion Constraints

A MEMS-based navigation solution can also be improved by using several motion constraints, such as NHC, ZUPT, and ZARU (Zero Angular Rate Update). NHC (Syed et al. 2008) uses the fact that a land vehicle cannot move sideways or vertically. It can work as a velocity update to improve the MEMS solution. NHC is also suitable for normal pedestrian walking. ZUPT uses zero velocity as the velocity update to limit velocity error if the pedestrian is static. ZARU considers the fact that the heading remains unchanged to limit the attitude error if the pedestrian is static. With these motion constraints, a MEMS-based navigation solution can perform better than before.

NHC, also known as velocity constraints, can be used to improve the performance of MEMS-based navigation solutions, especially when there are no other wireless signals. NHC uses the fact that a land vehicle cannot suddenly move sideways or vertically. Therefore, these two velocity

components should be close to zero. This velocity constraint also can be used for typical pedestrian walking to constrain the lateral and vertical speeds of the pedestrian. The NHC equations for the NED implementation of the navigation frame are as follows.

$$\begin{cases} v_y^b \approx 0 \\ v_z^b \approx 0 \end{cases} \tag{2-9}$$

where y represents the lateral component of the velocity, z represents the vertical component of the velocity, and b represents the pedestrian body frame.

If a static interval is detected, ZUPT and ZARU are used as motion constraints for the INS to limit the navigation error. The ZUPT-based zero velocity vector in the body frame is given by

$$v_{ZUPT}^b = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \tag{2-10}$$

If the pedestrian is detected as static, the pedestrian heading is unchanging based on ZARU, which is given by

$$\psi_{INS} = \psi_{pre-stored} \tag{2-11}$$

where ψ_{INS} is the INS-based heading and $\psi_{pre-stored}$ is the pre-stored heading of the first epoch after the static is detected.

2.1.5 Limitations of MEMS-Based Solution

MEMS sensors are widely used in many applications; they can be found in various handheld products such as smartphones, tablets, and personal digital assistants (PDAs). However, measurements of low-cost MEMS sensors are usually contaminated by different types of error

sources: bias, bias variations, and scale factor, etc. Therefore, MEMS sensors cannot be used to provide relatively long-term accurate solutions without external aiding sources, especially by using the INS mechanization equations. The navigation error based on this approach increases rapidly with time due to the MEMS errors and the integrations used in the mechanization (Titterton and Weston 2004). PDR reduces the accumulated speed of the navigation error by decreasing the use of integrations. PDR used in the handheld devices usually assumes that the handheld device is leveled (roll and pitch are zero degrees). However, this assumption is not always valid. In these cases, the PDR-based heading, calculated by the direct integration of the vertical gyroscope is inaccurate. The heading estimation error will finally affect the positioning accuracy. To compensate for the limitations of INS and PDR, we propose a MEMS solution on handheld devices for indoor navigation, based on the use of PDR/INS integration. The proposed PDR/INS-integrated MEMS solution combines the advantages of both schemes. In this algorithm, step detection and step length estimation are kept the same as the traditional PDR algorithm. The estimated step length is used to calculate the forward speed, which works as the velocity update for the INS to limit the velocity error, and further limit the position error and attitude error. Therefore, the PDR/INS-integrated MEMS solution is superior to the INS solution. The heading from the PDR/INS integration also performs better when compared with PDR because it considers the effect of the roll and pitch. Furthermore, motion constraints are also used to improve the MEMS-based navigation solution. Even using these algorithms for MEMS sensors, the navigation solution still slowly drift with time. Therefore, wireless signals are usually used to aid the MEMS sensors to limit their drifts. WiFi is the main wireless signal in indoor environments, and typical WiFi positioning systems are discussed in Section 2.2. Two integration approaches (LC integration and TC integration) are also discussed in Section 5 for MEMS sensors and WiFi.

2.2 WiFi Solution for Indoor Navigation

WiFi based positioning is a candidate technology for indoor navigation because it provides location information using pre-existing WiFi infrastructures. Currently, most public buildings, such as universities, colleges, airports, shopping malls, and office buildings, already have well established WiFi infrastructures. WiFi localization error does not accumulate with time which makes it a potential aiding source for the standalone navigation solution based on MEMS sensors.

2.2.1 Trilateration

2.2.1.1 Radio Propagation Model

The relationship between transmitter power and receiver power is described by simplified path loss method (Goldsmith 2005), and is shown as follows.

$$P_r\left(dBm\right) = P_t\left(dBm\right) + K\left(dB\right) - 10n\log_{10}\left(\frac{d}{d_0}\right) \tag{2-12}$$

where P_r is the RSS value received at the WiFi receiver in dBm at a distance d from the transmitter, P_t is the transmitted signal strength of the AP. K is a unitless constant depends on the antenna characteristics and the average channel attenuation, d_0 is a reference distance for the antenna far-field, and n is the path loss exponent which depends on the propagation environment. d_0 is typically assumed to 1-10m indoors and 10-100m outdoors. Typical values of this parameter are $n = 2$ for free space and $2 \le n \le 6$ for an office building with multiple floors (Goldsmith 2005). The value of K is sometimes set to the free space path loss at the distance d_0.

$$K\left(dB\right) = -20\log_{10}\left(4\pi d_0 / \lambda\right) + 10\log_{10}\left(G_t\right) + 10\log_{10}\left(G_r\right) \tag{2-13}$$

Where λ is WiFi signal wavelength, G_t is the gain of the transmitting antenna, and G_r is the gain of the receiving antenna. Shadowing of channels should be carefully considered in indoor environment. The effects of shadowing are modeled statistically in the uncertain and changing indoor environment. The assumption of log-normal random process with zero mean is applied for shadowing in (Goldsmith 2005). One term $-\psi(dB)$ indicating the shadows is added to Equation (2-12) for this. Combining Equations (2-12) and (2-13), a new propagation model is formulated as follows.

$$P_r(dBm) = -10n\log_{10}\left(\frac{d}{d_0}\right) + P_0(dBm) - \psi \tag{2-14}$$

where $P_0(dBm) = P_t(dBm) + K(dB)$ is the received signal strength at distance d_0. Equation (2-14) is simplified to Equation (2-15) through averaging and assuming $d_0 = 1$, and is given as follows.

$$RSS = -10n\log_{10}(d) - A \tag{2-15}$$

where $A = -mean(P_0(d = 1m)) - mean(\psi)$. Another approach for deriving (2-15) based on MLE (Maximum Likelihood Estimation) is given in (Mazuelas et al. 2009). The typical range for A is 0 ~ 100. Equation (2-15) is the simplified propagation model used in this research.

2.2.1.2 Trilateration-Based Position Estimation

Trilateration-based WiFi positioning consists of two steps: range estimation and position estimation. First, distances (ranges) between the user and APs are estimated from the RSS values by using the propagation model. Second, the user's position is calculated by applying estimation

techniques for the ranges and APs' locations. Equation (2-16) is given to calculate the distance d by rewriting Equation (2-15).

$$d = 10^{\frac{RSS+A}{-10n}} \tag{2-16}$$

Propagation parameters (A and n) and RSS are needed to calculate the distance d . We can easily obtain RSS values from the WiFi receivers in handheld devices. Typical values of A and n can be set for Equation (2-16). However, typical values are not suitable for a specific indoor environment. A and n values can be obtained from pre-surveys, and they are stored in the database. AP locations are additional necessary information for trilateration-based WiFi positioning, and they are usually obtained by pre-surveys or uploads from users. With known AP locations and ranges, typical estimation techniques, such as non-linear iterative LSQ, are utilized to estimate user's positions. The details about non-linear iterative LSQ are given in Section 2.4.2.

2.2.2 Fingerprinting

Fingerprinting based WiFi positioning includes two phases: pre-survey and real-time positioning (Bahl and Padmanabhan 2000). The pre-survey is to build the radio map databases by measuring and storing the positions and corresponding RSS values at measurement points. Real-time positioning uses several approaches to determine the user's position by comparing current RSS values with radio map databases. Building the radio map databases will be discussed in the next section. Then, the approaches for estimating user's positions by using radio map databases will be discussed in Section 2.2.2.2.

2.2.2.1 Pre-Survey Phase

Pre-surveys are usually required to build the radio map databases which contain a location fingerprint F labeling with a location information L. The location fingerprint is based on some RF characteristics such as RSS, which is the basis for representing a unique location. The location information L is defined to differentiate a special location from other locations. The radio map databases are used to estimate the user's position during the real-time positioning phase. The location and fingerprint are usually denoted as a tuple of (L, F) (Zhang et al. 2011).

The location information L is usually stored in radio map databases in the form of a tuple of coordinates. For 3D systems, three dimension space and two orientation of variables make up the tuple of coordinates. For 2D systems, the tuple of coordinates consists of two dimension space and one orientation variables, which is given by

$$L = \left\{ (x, y, d) \mid x, y \in R^2, d \in \{North, East, South, West\} \right\} \tag{2-17}$$

where x, y represent 2D coordinates, and d represents the heading.

RSS is the most effective RF signature for location fingerprints in WiFi positioning systems (Outemzabet and Nerguizian 2008) (Bahl and Padmanabhan 2000). RSS values are more dependent on locations than SNR (Signal to noise ratio) values because the noise in SNR is random in nature. However, RSS has one main drawback, which is that it fluctuates over time even at the same location for the same AP. The environment change is the main reason for this fluctuation, which can be caused by the moving of the passerby. Typically, the mean of RSS values at each measurement point is calculated and recorded as an element ρ_i for the location fingerprint. For a

measurement point that can obtain RSS values from N APs, the location fingerprint can be given by

$$F = \left(\rho_1, \rho_2, ..., \rho_N\right)^T \tag{2-18}$$

where ρ_i is an average RSS element. Radio map databases with tuples of $\left(L, F\right)$ are built by combining L and F at each measurement point.

An alternate approach for building the radio map database is given in (Roos et al. 2002) which calculates probability distributions of all measurement points as location fingerprints. Different from the average of RSS values, the location fingerprint in this approach is the probability distribution $f(r \mid p)$, where r represents the RSS vector and p represents the location of the measurement point. The conditional probability $f(r \mid p)$ is usually called the likelihood function because it represents the probability of occurrence of r when given p . Furthermore, the probability approaches are utilized for real-time positioning corresponding to this type of location fingerprint.

Two different location fingerprints define two frames for fingerprint-based WiFi positioning systems: deterministic frame and probability frame. Real-time positioning approaches corresponding to these two location fingerprints are also different. The details about the real-time positioning approaches are discussed in Section 2.2.2.2.

2.2.2.2 Real-Time Positioning Phase

If the radio map database has been successfully set up, real-time positioning can use several approaches for RSS values and radio map databases to determine the user's position. These

approaches can be classified into two schemes: deterministic frame and probability frame. They are given in detail as follows.

- Deterministic Frame

In this frame, the user's position is calculated as the weighted average of selected measurement points' positions by some criteria. Weights are determined by the inverse of the norm of the RSS innovation (Honkavirta et al. 2009). The details are given in Equation (2-19) and (2-20).

$$\hat{p} = \sum_{i=1}^{M} \frac{\omega_i}{\sum_{j=1}^{M} \omega_j} p_i \tag{2-19}$$

$$\omega_i = \frac{1}{\|r - r_i\|} \tag{2-20}$$

where $\hat{p}$ is the estimated user's position, p_i is the position of the i^{th} measurement point, ω_i is the weight corresponding to the i^{th} measurement point, r is the measured RSS vector of current position, r_i is the RSS vector in the i^{th} measurement points, and M is the total number of observable APs. The norm $\|\cdot\|$ is arbitrary, and the Euclidean norm (2-norm) is widely used and given as follows.

$$2-norm: \|x\|_2 = \sqrt{\sum_{i=1}^{Nx} |x_i|^2} \tag{2-21}$$

The weighted K-nearest neighbour (Li et al. 2006a) is another popular method, which keeps the largest K weights and sets others to zeroes. The K-nearest neighbour (KNN) is a special type of WKNN, in which K neighbours have equal weights.

- Probability Frame

In the probability frame, determination of the user position can be considered as a probability problem. The aim of this probability problem is to estimate the optimal solution for the user's position from the probability density functions. Three optimality criteria have been widely used for positioning: (1) maximization of the likelihood density (Gelb 1974), (2) minimization of the mean square error (Maybeck 1982), and (3) maximization of the posteriori density (Maybeck 1982). Their corresponding optimal estimators are as follows: (1) maximum likelihood (ML) estimator (Gelb 1974), (2) minimum mean square error (MMSE) estimator (Maybeck 1982), and maximum a posteriori (MAP) estimator (Maybeck 1982). The ML estimator finds the user position estimate by maximizing the likelihood density function, shown as:

$$\hat{p}_{ML} = \arg\max_{p} f\left(r \mid p\right) \tag{2-22}$$

where $f\left(r \mid p\right)$ is the likelihood density. The ML estimator chooses one measurement point with the maximum likelihood density as the estimate for the user's position. If the measurement points are sparsely distributed, the positioning accuracy is limited by only choosing one measurement point as the position estimate. To improve the position accuracy, we calculate the position estimate by averaging (or weighted averaging) K measurement points with largest likelihood densities. This method is also known as KNN (or WKNN). This KNN method is different from the KNN in the deterministic frame because the K neighbours here are chosen by the ML estimator, whereas the K neighbours are selected by the norms of the RSS innovations in the deterministic frame. The MMSE estimator calculates the user's position by minimizing the mean square of positioning errors, and the equation is given as follows.

$$\hat{p}_{MMSE} = \arg\min_{\hat{p}} \left\{ \left(p - \hat{p} \right)^T \left(p - \hat{p} \right) | r \right\} \tag{2-23}$$

where $\hat{p}$ is the estimated user's position. Last, the MAP estimator is formulated as follows.

$$\hat{p}_{MAP} = \arg\max_{p} f\left(r | p \right) f\left(p \right) \tag{2-24}$$

where $f\left(p \right)$ is a priori density of p. The ML estimator can be thought of as a special case of the MAP estimator without a priori information.

2.2.3 Limitations of WiFi-Based Solution

WiFi-based positioning is a potential aiding source for the standalone navigation solution based on MEMS sensors. However, both trilateration and fingerprinting require special databases to estimate the user position. AP locations and PPs are necessary for trilateration-based WiFi positioning. Fingerprinting estimates the user position by finding the closest fingerprints within the radio map database. In traditional approaches, professional surveyors are hired to build and maintain the databases. A radio map database requires intensive surveys of the areas where the RSS of available APs are mapped with respect to absolute positions. Pre-survey is also needed to build the database of PPs and AP locations for trilateration-based WiFi positioning. Pre-survey, a labor-intensive and time-consuming process, is one of the limitations in most current WiFi positioning systems. In addition, if an indoor environment is changed due to the removal or addition of WiFi routers, this survey must be redone to maintain the database.

One purpose of this research is to design an automatic indoor WiFi positioning system (WPS), with virtually no pre-survey, through crowdsourcing. In order to achieve this aim, two different

automatic WPSs are proposed based on trilateration and fingerprinting. T-PN is a commercial software that converts inertial sensors into navigation solution that can be used on any smartphone operating system (e.g. Android). This software is used to automatically build the databases. In both schemes, a background survey service runs on the operating system of handheld devices to build databases automatically. Another positioning service can also be activated to provide a positioning solution for the user. In the trilateration scheme, a background survey service estimates AP locations and PPs automatically. These values are estimated by using nonlinear iterative least squares (LSQ) and recorded in the database when some pairs of the T-PN solution and corresponding RSS values meet the pre-set requirements. The estimated accuracy of AP locations is also stored in the database for the future use of WiFi positioning. Autonomous crowdsourcing is used to update the AP information in the database and keep data accurate. The database update happens automatically in the background, without any restriction on the user, thus making the crowdsourcing completely autonomous. The positioning service is mainly based on trilateration and positioning result optimization through the use of the automatically surveyed database. In the fingerprinting scheme, the background pre-survey builds the radio map database automatically. In the crowdsourcing model, fingerprints are generated automatically, whether the user is walking or static, as long as the service is running in the background. The accuracy of the database will be improved when more fingerprints are generated to update the database through autonomous crowdsourcing. Because the system does not guarantee that the radio map database contains all the fingerprints in the building, an improved positioning algorithm is designed in the proposed system.

2.3 Integrated Navigation Solutions

Our proposed indoor pedestrian navigator is based on the cooperation of MEMS sensors and WiFi. The key idea of this system is that the MEMS-based navigation solution is used to build the database for WiFi through crowdsourcing when it is accurate, whereas the WiFi solution is used to reduce the drift of MEMS sensors when its database is successfully built. The basic question as to "How to automatically build the WiFi database based on crowdsourcing using the MEMS-based navigation solution?" has been discussed in the last section. In this section, some background knowledge will be given on "How to use wireless signals to reduce the drifts of MEMS sensors." GPS/INS integration is used as an example to introduce the background of integrated navigation solutions because it is the most common. The methodology and result of WiFi and MEMS integrated systems are discussed in Chapter 5.

The GPS/INS integrated system has several advantages. INS can fill the gap of GPS signal outages to implement a seamless navigation solution. On the other hand, GPS signals can be used to aid the INS to reduce the drifts. Another advantage of the GPS/INS integrated system is that it can provide redundant measurements and improve the reliability of the navigation system. Usually, there are two schemes for GPS/INS integrated systems: LC integration and TC integration.

2.3.1 Loosely-Coupled Integration

The most popular integration for GPS and INS is the loosely-coupled integration. In this integration, these systems operate independently and provide two navigation solutions. Usually, GPS-based position and velocity as well as the INS solution are fed to a KF. The error states consist of position errors, velocity errors, and attitude errors, as well as INS errors. The KF can estimate these errors by using the difference between GPS and INS solutions and the error model. To further

improve the accuracy of the navigation solution, the estimated INS errors are fed back into the INS mechanization. The INS solution is also corrected for these errors to produce an improved integrated navigation solution. A block diagram of the LC GPS/INS integration is depicted in Figure 2-3. Note that "GPS Kalman Filter" usually exists in the GPS part, however, it is not mandatory.

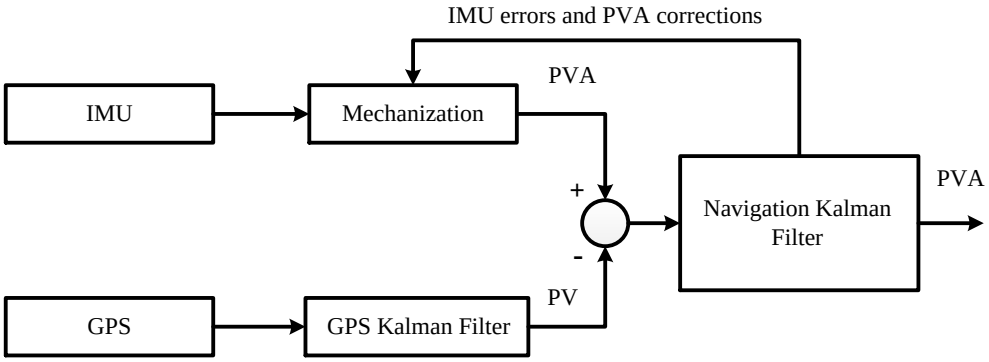


Figure 2-3 Loosely-coupled GPS/INS integration.

The main advantage of the LC integration is that it is simple to implement and robust, i.e. a smaller size of KF states is used in this integration when compared to the tightly-coupled integration. It provides three navigation solutions: GPS, INS, and GPS/INS-integrated solutions. The main disadvantage of LC integration is that it cannot provide a GPS solution to aid the INS when there are less than four satellites available. Another advantage is that this integration has two KFs, which introduce more process noise and decrease the signal-to-noise ratio.

2.3.2 Tightly-Coupled Integration

TC integration is also known as centralized integration which only uses a single common filter. The difference between the pseudo-range and pseudo-range rate measurements from GPS and INS are fed to the KF to estimate the navigation errors, GPS receiver clock errors, and INS errors. INS errors are fed back into the INS mechanization to correct the integrated navigation solution.

Usually, GPS receiver clock errors are also fed back into the GPS receiver to improve the GPS-based pseudo-ranges and pseudo-range rates. A block diagram of the TC GPS/INS integration is depicted in Figure 2-4.

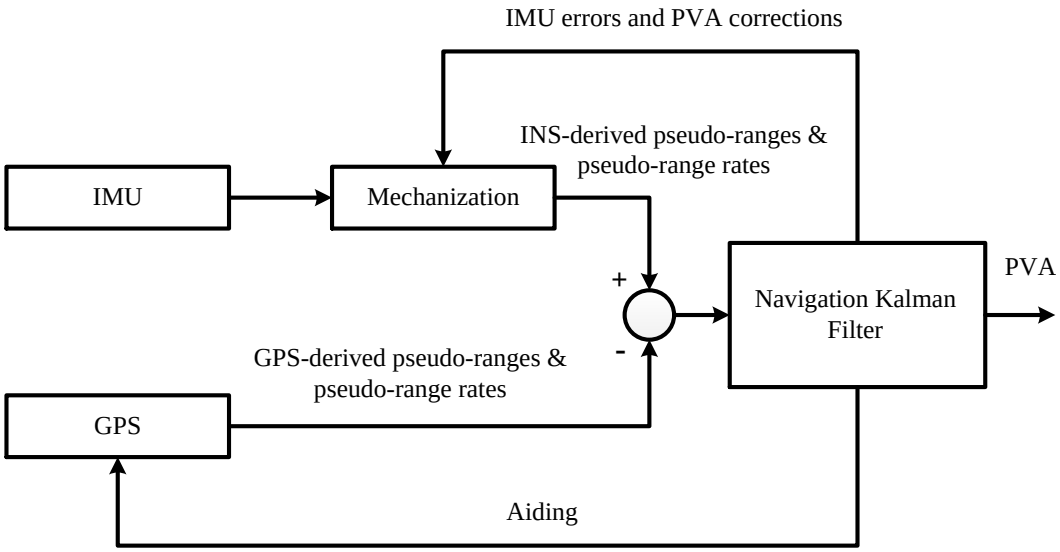


Figure 2-4 Tightly-coupled GPS/INS integration.

The main advantage of TC integration is that it can provide a GPS update for INS even when there are less than four satellites available. This advantage makes the TC integration work in challenging environments, such as urban canyons, where the number of available satellites are less than four. However, this integration is more complex to implement, as the algorithm involves processing GPS pseudo-ranges and pseudo-range rates. Another disadvantage of this integration is that there is no stand-alone GPS solution. Typically, the TC integration can provide a more accurate navigation solution when compared to the LC integration.

2.4 Estimation for Navigation

Many navigation applications need estimation theory to determine the parameters and their covariance from redundant measurements. Least squares is the most commonly used approach to convert redundant measurements to parameters. Dynamics also can be combined with the redundant measurements to achieve the optimal estimates if the system has them. The KF is a popular algorithm for estimating states from measurements and dynamics.

2.4.1 Kalman Filtering

The Kalman filter is a well-known optimal filter based on minimizing the variance estimation of system dynamics and measurements. It is usually used to fuse multiple navigation solutions. The KF has two models: the system model and the measurement model. Both models consist of a deterministic and a stochastic part. The general KF operates in two steps: a prediction and an update step. The prediction step uses system dynamics to predict the next state vector and the state covariance matrix while the update step combines the measurements and the prediction to give the final estimates and their covariance matrix. The KF has the capability to recursively estimate the current state vector based on previous steps and current measurements.

Given the fact that measurements usually occur at discrete times, the KF works in the discrete mode for navigation applications. Therefore, the system dynamic model must be converted to the discrete format, which is given in the following equation.

$$x_k = \Phi_{k-1,k} x_{k-1} + w_{k-1} \tag{2-25}$$

where x_k and x_{k-1} represent the state vectors at epoch k and $k-1$, $\Phi_{k-1,k}$ represents the state transition matrix from epoch $k-1$ to epoch k, and w_{k-1} represents the process noise. The measurement model in the discrete form is given as follows.

$$z_k = H_k x_k + v_k \tag{2-26}$$

where z_k represents the measurement vector at epoch k, H_k represents the design matrix at epoch k, and v_k represents the measurement noise.

The KF algorithm is made up of two parts: prediction and update. The prediction part is responsible for predicting the state vector from epoch to epoch by using the transition function based on the system dynamics. The prediction equations are formulated as follows.

$$\hat{x}_k^- = \Phi_{k-1,k}\hat{x}_{k-1}^+ \tag{2-27}$$

$$P_k^- = \Phi_{k-1,k}P_{k-1}^+\Phi_{k-1,k}^T + Q_{k-1} \tag{2-28}$$

where ($\hat{}$) denotes estimation, (-) denotes the estimated value after prediction, and (+) denotes the estimated value after update. x represents the navigation state vector, P represents the covariance matrix of the state vector, $\Phi_{k-1,k}$ represents the transition matrix from epoch $k-1$ to epoch k, and Q_{k-1} is the system noise matrix. The update equations are given by

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} \tag{2-29}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k\left(z_k - H_k\hat{x}_k^-\right) \tag{2-30}$$

$$P_k^+ = \left(I - K_k H_k^T\right) P_k^- \tag{2-31}$$

where K_k is the Kalman gain, R_k is the measurement covariance matrix, and H_k is the measurement design matrix. The general process of the discrete time KF is shown in Figure 2-5.

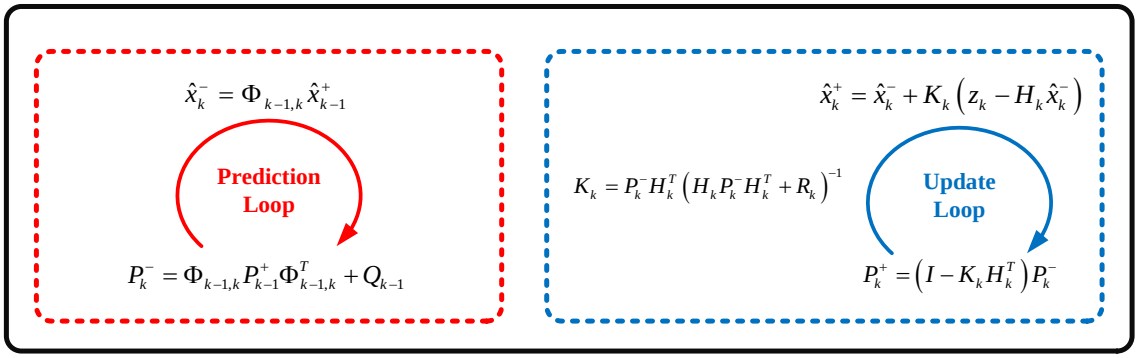


Figure 2-5 General process of the discrete time KF

2.4.2 Extended Kalman Filter

KF assumes that the system model and measurement model are linear. However, this assumption is not always satisfied for all the applications, such as the GPS/INS integration system. In this case, the non-linearity is mainly derived by estimating the user's position, velocity, and attitude from the mechanization equations. There are two approaches to process the non-linear systems. First, the system is linearized based on a nominal or approximate trajectory during the design of the KF. Second, the system is linearized about the actual trajectory, which is done by linearizing the process around the current state. The second approach is commonly known as the Extended Kalman Filter (EKF). The EKF is usually used to fuse other information such as position and velocity from GNSS to reduce the drift characteristics of the MEMS sensors. When the EKF is used to fuse other information for INS, the state vector is determined first as follows:

$$x_s = \left[\delta r_{1\times3} \quad \delta v_{1\times3} \quad \varepsilon_{1\times3} \quad d_{1\times3} \quad b_{1\times3}\right]^T \tag{2-32}$$

where δr , δv , and ε represent errors of position, velocity and attitude. d and b represent gyroscope drift and accelerometer bias, which are estimated and fed back to the INS mechanization. The discrete-time EKF system model and observation model can be expressed as

$$\begin{cases} \delta x_k = \Phi_{k-1,k}\delta x_{k-1} + w_k \\ \delta z_k = H_k \delta x_k + v_k \end{cases} \tag{2-33}$$

where δx_k represents the state vector at epoch k ; $\Phi_{k-1,k}$ represents the state transition matrix from epoch $k-1$ to epoch k ; and w_k represents the process noise. δz_k represents the observation misclosure vector at epoch k ; H_k represents the design matrix at epoch k ; and v_k represents the observation noise. For more knowledge about the EKF for the integrated navigation system, please refer to (Gelb 1974).

2.4.3 Nonlinear Least Squares

The method of least squares is the standard approach to obtain unique values for parameters from related redundant measurements through a known observation model. The typical observation model for the LSQ is given in Equation (2-34) (Petovello 2012).

$$\mathbf{z} = h(\mathbf{x}) + \mathbf{v} \tag{2-34}$$

where $\mathbf{z}$ is the measurement vector, and $h(\mathbf{x})$ is a function of the state vector $\mathbf{x}$. A Taylor series is then used to linearize the nonlinear measurement vector by expanding the terms around the current estimated state, $\hat{\mathbf{x}}$, as shown in Equation (2-35). Only the first order term is used in the linearization.

$$
\begin{aligned}
\mathbf{z} &= h(\mathbf{x}) + \mathbf{v} \\
&= h(\hat{\mathbf{x}}) + \left.\frac{dh(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} (\mathbf{x} - \hat{\mathbf{x}}) + \cdots + \mathbf{v} \\
&\approx h(\hat{\mathbf{x}}) + \left.\frac{dh(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} (\mathbf{x} - \hat{\mathbf{x}}) + \mathbf{v} \\
\mathbf{z} &= h(\hat{\mathbf{x}}) + \mathbf{H}\delta\mathbf{x} + \mathbf{v}
\end{aligned}
\tag{2-35}
$$

where $\delta\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}}$ represents the "error" in the state vector and $\mathbf{H} = \frac{dh(\mathbf{x})}{d\mathbf{x}}$ is the design matrix. Rearranging Equation (2-35) will give a measurement misclosure vector ($\delta\mathbf{z}$) as shown in Equation (2-36). Equation (2-36) is a linear observation model.

$$
\begin{aligned}
\mathbf{z} - h(\hat{\mathbf{x}}) &= \mathbf{H}\delta\mathbf{x} + \mathbf{v} \\
\delta\mathbf{z} &= \mathbf{H}\delta\mathbf{x} + \mathbf{v}
\end{aligned}
\tag{2-36}
$$

The solution, $\delta\hat{\mathbf{x}}$, and its covariance matrix, $\mathbf{C}_{\delta\hat{\mathbf{x}}}$, are given in (Petovello 2012) and provided below in Equation (2-37) as

$$
\begin{aligned}
\delta\hat{\mathbf{x}} &= (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\delta\mathbf{z} \\
\mathbf{C}_{\delta\hat{\mathbf{x}}} &= (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}
\end{aligned}
\tag{2-37}
$$

where $\mathbf{R}$ is the covariance matrix of observations. The new state vector is calculated as

$$
\hat{\mathbf{x}}_{updated} = \hat{\mathbf{x}} + \delta\hat{\mathbf{x}} \tag{2-38}
$$

and, the observation model is expanded at the new state vector, $\hat{\mathbf{x}}_{updated}$. It is an iterative process until $|\delta\hat{\mathbf{x}}| < threshold$. Equation (2-39) provides the residual and covariance equations as follows.

$$\begin{aligned} \mathbf{r} &= \mathbf{z} - h(\hat{\mathbf{x}}) \\ \mathbf{C}_{\mathbf{r}} &= \mathbf{R} - \mathbf{H}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T \end{aligned} \tag{2-39}$$

where $\mathbf{r}$ is the residual vector of LSQ and $\mathbf{C}_{\mathbf{r}}$. $\mathbf{C}_{\mathbf{r}}$ is its covariance matrix. The measurement covariance matrix can be written as

$$\mathbf{R} = \sigma_0^2 \mathbf{Q}_{\mathbf{R}} \tag{2-40}$$

where σ_0^2 is the a-priori variance factor, and $\mathbf{Q}_{\mathbf{R}}$ is the cofactor matrix of $\mathbf{R}$. The solution of the nonlinear LSQ is given by (Petovello 2012).

$$\begin{aligned} \delta\hat{\mathbf{x}} &= (\mathbf{H}^T\mathbf{Q}_{\mathbf{R}}{}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Q}_{\mathbf{R}}{}^{-1}\delta\mathbf{z} \\ \mathbf{C}_{\delta\hat{\mathbf{x}}} &= \sigma_0^2(\mathbf{H}^T\mathbf{Q}_{\mathbf{R}}{}^{-1}\mathbf{H})^{-1} \\ \mathbf{r} &= \mathbf{z} - h(\hat{\mathbf{x}}) \\ \mathbf{C}_{\mathbf{r}} &= \sigma_0^2\left(\mathbf{Q}_{\mathbf{R}} - \mathbf{H}(\mathbf{H}^T\mathbf{Q}_{\mathbf{R}}{}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\right) \end{aligned} \tag{2-41}$$

Note that the estimations of $\delta\hat{\mathbf{x}}$ and $\mathbf{r}$ are independent of σ_0^2. However, σ_0^2 scales $\mathbf{C}_{\delta\hat{\mathbf{x}}}$ and $\mathbf{C}_{\mathbf{r}}$ directly, as shown in Equation (2-41). On the other hand, $\mathbf{Q}_{\mathbf{R}}$ affects $\hat{\mathbf{x}}$, $\mathbf{r}$, $\mathbf{C}_{\delta\hat{\mathbf{x}}}$, and $\mathbf{C}_{\mathbf{r}}$. To use nonlinear LSQ for estimation problems, the key step is to determine the observation model and state vector, including $\mathbf{x}$, $\mathbf{z}$, $\mathbf{R}$, $\mathbf{H}$, and $\hat{\mathbf{x}}$.

Chapter Three: **Automatic WPS Based on Trilateration**

This chapter and the next chapter mainly focus on two automatic WPSs on handheld devices: trilateration-based and fingerprinting-based. In these two chapters, two systems are carefully discussed, evaluated, and compared. The results are used to select an appropriate WPS to integrate with MEMS sensors. In this chapter, an efficient and practical trilateration-based WPS is proposed in order to overcome the extensive surveying needed by traditional systems. The main purpose of this research is to reduce the labor needed for the survey of WiFi databases. Currently, most WPSs based on trilateration assume that AP locations and PPs are available from pre-surveys (Yim et al. 2008).

Most public buildings, such as universities, colleges, airports, shopping malls, and office buildings already have well established WiFi infrastructure. WiFi positioning solutions do not drift as compared to standalone inertial navigation solutions using MEMS sensors. However, current WiFi positioning systems (WPSs) usually require pre-survey to provide AP locations, PPs, or radio maps (Bahl and Padmanabhan 2000; Hui et al. 2007; Swangmuang and Krishnamurthy 2008). The pre-survey is time and labor consuming, which makes most current WPSs not practical. In fact, even if this information is available, it may not be suitable for real-time WiFi positioning due to the changing environment. Changes in the environment could be caused by the following situations:

- Removal or addition of WiFi routers;
- Temporary loss of signals from one or more routers; or
- Changes in the obstruction pattern from survey time to data collection time.

Consequently, the automatic estimation for AP locations and PPs is an effective way to ensure accurate WiFi positioning. An autonomous system will also reduce the labor and time costs for

surveys to maintain the databases because crowdsourcing will be updating the databases in the background. Unfortunately, most current methods cannot automatically estimate AP locations and PPs, adapting to the changes in the environment.

In order to implement an automatic and practical WPS, a novel algorithm for the background survey service is proposed by using a MEMS-based navigation solution, such as the T-PN (Trusted Positioning Inc. Portable Navigator). The algorithm includes AP localization, PPs estimation and autonomous crowdsourcing. T-PN is highly customizable software that converts any quality and grade of inertial sensors into navigation capable sensors that can be used on many smartphone operating system (e.g. Android). In this research, T-PN is used as an example of the navigation solution provider, and other providers can also be used in our proposed system. T-PN provides the user position information and position accuracy as observations to build the database. Therefore, the accuracy of the automatically built database relies on the accuracy of the navigation solution. However, the T-PN solution can be improved if the WiFi positioning solution is estimated by using the automatically built database. AP locations and PPs are estimated using nonlinear LSQ and the corresponding information is recorded in the database when some pairs of the T-PN solution and corresponding RSS values meet the pre-set requirements. Additionally, the estimation accuracy of the AP localization data is also stored in the database to be used for WiFi positioning in the future. The function of autonomous crowdsourcing is to update the AP information in the database to ensure the accuracy of the database. The database update happens automatically in the background, without any restriction on the user; thus, making the crowdsourcing completely autonomous.

The WiFi positioning service contains two steps. First, RSS values are converted to ranges using the propagation model based on PPs from an automatically surveyed database. Next, user position

is estimated based on nonlinear LSQ and positioning result optimization. The main contributions of this research are as follows:

- A convenient and practical WPS on smartphones is proposed to reduce the labor of pre-surveying and to improve the positioning accuracy.
- Novel algorithms for the background survey service, which includes estimating AP locations and PPs in the propagation model and autonomous crowdsourcing, are proposed.
- The proposed system is implemented on smartphones and evaluated by both simulations and real-world experiments.

The remainder of this chapter is organized as follows. Section 3.1 describes the overview of the proposed system. Section 3.2 presents the T-PN solution. Section 3.3 presents the optimization of measurements. Section 3.4 describes the background survey service, including the algorithms for AP localization, PPs estimation and autonomous crowdsourcing. Section 3.5 describes the proposed WiFi positioning service, and is followed by test results and the performance analysis. Finally, Section 3.7 gives the summary of this chapter.

3.1 System Overview

In this chapter, a WPS based on autonomous crowdsourcing is proposed for handheld devices with the support of a MEMS solution (e.g. T-PN). Figure 3-1 shows the structure for the proposed system. In the proposed system, background survey service and WiFi positioning service are two significant services running on a handheld device. RSS values and position information from the MEMS solution (e.g. T-PN) are inputs for the background survey service. This service outputs the AP information (AP locations and PPs) to the WiFi database. Background survey service is mainly based on crowdsourcing, and reduces the labor consumption for the survey process. WiFi

positioning service provides a WiFi position solution through improved algorithms based on trilateration with the help of the database. WiFi solution also can be used as an aiding source for the MEMS solution (e.g. T-PN) to improve its performance. Details about this system are described in Section 3.3 and Section 3.4.

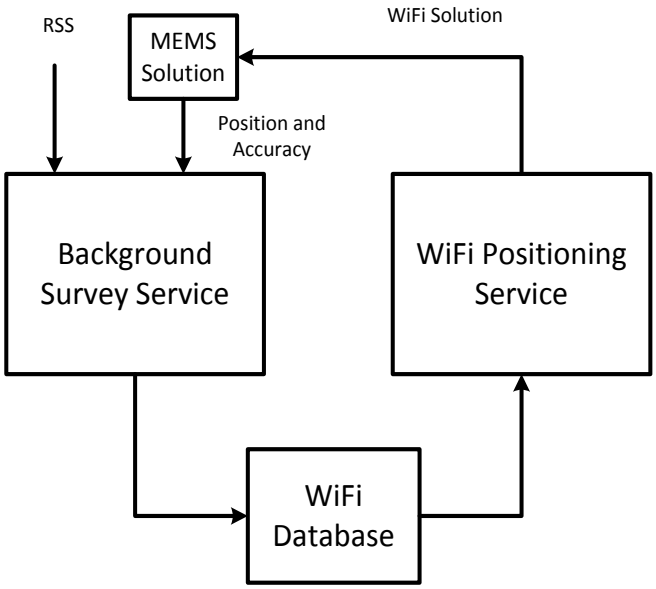


Figure 3-1 System overview of the proposed automatic WPS.

3.2 T-PN Solution

T-PN is a highly customizable software that provides an inertial sensors based navigation solution and can be used on many of available smartphone/tablet operating systems such as Android (Zhuang et al. 2013b). This engine improves the navigation results by taking any available absolute measurements as filter updates. GPS is the most common type of external update that provides absolute position and velocity information to the inertial engine and limits the drift errors.

Physical movements of the user, such as pedestrian dead reckoning, zero velocity updates and non-holonomic constraints, are used as constraints to improve the navigation solution. The constraints

are also tailored to user transit mode (such as walking, cycling, taking a bus, etc.) to ensure the most robust navigation solution for the user. The mode of transit is automatically detected on a continuous basis. If additional sensors such as a magnetometer and barometer are present and properly calibrated, their readings can be used as optional updates by the T-PN. Most of the handheld devices today usually have both magnetometers and barometers and in this case the T-PN provides a 3D PVA of the system.

Two examples of the T-PN navigation solution in different scenarios are shown in Figure 3-2. The experiments were executed by using a Samsung Galaxy SIII. T-PN navigation solution is derived from the integration of 3-axis accelerometers, 3-axis gyroscopes, 3-axis magnetometers and a barometer in Samsung Galaxy SIII. In the experiments, the user held the smartphone, and walked normally. The experiments were carried out in building E and the west part of building M. Building E is the building of Energy, Environment, and Experiential learning (EEEL), University of Calgary, which is about 120m × 40m. Building M is the MacEwan Student Center (MSC), University of Calgary, with a west part about 90m × 70m. These two trajectories lasted 2 minutes and 3 minutes, respectively. Note that reference trajectory is provided by using several pre-set markers' positions from the floor plan of the building. The initial position and heading are given manually by using the floor plan. The results show that the maximum position errors of the T-PN solution are less than 5 meters in these two trajectories when comparing to the reference trajectories. Therefore, T-PN has been adopted in this thesis as a reliable position provider for WiFi database generation through autonomous crowdsourcing. Note that T-PN also provides an indicator for the accuracy of its navigation solution. This accuracy indicator is a significant factor in the proposed automatic database generation algorithms.

(a)

(b)

Figure 3-2 Examples of the navigation solution from the T-PN with respect to reference: (a) building E and (b) the west part of building M.

3.3 Measurements Optimization

To evaluate the stability of AP signals for WiFi-based positioning system, a test was conducted by recording the signals from a number of APs in the building E, University of Calgary. The test included 36 APs and was conducted using a Samsung Galaxy SIII in walking and static mode. The

test revealed that some APs with weak signals are not always observable even when the handheld device is static at the same place. Therefore, the response rate is introduced to evaluate the stability of AP signals. Preliminary results show that APs with RSS values greater than -75dBm provided a response rate of over 90%; APs with RSS values between -75dBm and -85dBm provided a response rate of about 70%; and APs with RSS values less than -85dBm provided a response rate of about 30%. If the user stands at a specific location for a long time, the response rate can be used to determine the quantity and quality of the recorded RSS information. However, in this research, measurements are collected by the background service on handheld devices. Sometimes, only one sample is collected at a measurement point when the mobile user is walking. In this case, a high response rate is used by setting the threshold to -85dBm to ensure the response rate of RSS signals, and to potentially increase the reliability of the database.

The fluctuation of RSS values also needs to be considered beyond the AP response rate. A three-point average is used to reduce the noise of RSS values. The current RSS value is re-determined by averaging the previous, current, and next RSS values. Of course, the average can improve the accuracy of the measured RSS value if the user is static. If the user is walking, the previous RSS and the next RSS are measured at points different from the current RSS value. However, the previous RSS and the next RSS are close to the current RSS because they only have one sample's difference. Furthermore, the previous and next measurement points are usually located at two opposite sides of the current measurement point, and thus these RSS values are usually complementary. This is helpful as WiFi RSS measurements are highly noisy. Therefore, no matter whether the user is static or moving, the average of three sample's RSS values will improve the

accuracy of the RSS value for building the database. Note that RSS sample rates in current handheld devices (smartphones and tablets) are usually in the range of 0.5 ~ 2 Hz.

The position and RSS information are collected as pairs to build the database in the background survey service of a handheld device. The position information from the T-PN solution includes geodetic coordinates - latitude, longitude, and height (LLH), as well as their accuracies. The geodetic coordinates LLH can be converted to coordinates in the local east-north-up (ENU) coordinate system. RSS values are read from the operating system running on the handheld device. To optimize the measurements, algorithms are designed to detect and solve the RSS ambiguity problem, in which the RSS values of two pairs are totally different, while the LLH coordinates are almost the same. This ambiguity problem is mainly caused by the fluctuation of RSS values and the navigation errors of T-PN. The RSS ambiguity problem is detected by using (3-1).

$$\begin{cases} \left[\left(horizontal\_dis(\mathrm{LLH}_1, \mathrm{LLH}_2) < hor\_th \right) \textbf{ and } \left(height\_dis(\mathrm{LLH}_1, \mathrm{LLH}_2) < floor\_th \right) \right] \\ \textbf{or} \\ \left[\left(horizontal\_dis(\mathrm{ENU}_1, \mathrm{ENU}_2) < hor\_th \right) \textbf{ and } \left(height\_dis(\mathrm{ENU}_1, \mathrm{ENU}_2) < floor\_th \right) \right] \\ \sigma_{h,1} < acc\_th \textbf{ and } \sigma_{h,2} < acc\_th \\ \sigma_{a,1} < floor\_th \textbf{ and } \sigma_{a,2} < floor\_th \\ \mathrm{E}\left[\mathrm{abs}\left(S_1 - S_2 \right) \right] > RSS\_th \end{cases} \quad (3\text{-}1)$$

where LLH_1 and LLH_2 represent the geodetic coordinates of two pairs; ENU_1 and ENU_2 represent the ENU coordinates of two pairs; $horizontal\_dis$ and $height\_dis$ represent the calculations of horizontal distance and height distance for two pairs; hor_th and $floor\_th$ represent the horizontal and floor thresholds for determining whether coordinates of two pairs are almost the same; $\sigma_{h,1}$ and $\sigma_{h,2}$ represent horizontal accuracies of two pairs; $\sigma_{a,1}$ and $\sigma_{a,2}$

represent the altitude accuracies of two pairs; acc_th represents the threshold of horizontal accuracy; RSS_th represents the RSS threshold; and S_1 and S_2 represent the RSS vectors of available APs. These thresholds will affect the performance of WPSs. Before discussing the setting of these thresholds, how to appropriately set the grid spacing for the WPSs will be discussed. If it is set too large, it decreases the accuracy of WiFi positioning. If it is set too small, it needs more data to build the database and uses more memory. In this thesis, the grid spacing is set to a balanced value of 3 meters, which is determined by experimentation. hor_th is set to the same as the grid spacing, and $floor\_th$ is set to a typical floor height (3 meters). acc_th is set to 5 meters, which is larger than the hor_th. It was not set to a smaller value since more useful data can be used for building the databases through crowdsourcing. Also, it is not set to a larger value, in which case T-PN is not accurate enough to provide navigation solutions. RSS_th is set to 5 dBm, which is the standard deviation of RSS values in the static field tests. If this ambiguity is detected, these two pairs will be replaced by a new pair given in

$$T_{new} = E\left[T_1, T_2\right] \tag{3-2}$$

where $T_i = \left\{\mathrm{ENU_i} \quad \mathrm{S_i}\right\}$ represents the measurement pair including the ENU coordinates $\mathrm{ENU_i}$ and the RSS values $\mathrm{S_i}$, and $E\left[\bullet\right]$ represents the expectation. The detection and solution of the ambiguity problem improves the reliability of measurements.

3.4 Background Survey Service

3.4.1 System Flow Chart

A flow chart and general description of the proposed algorithm for AP localization, PPs estimation and autonomous crowdsourcing is given in Figure 3-3 and summarized here. To prove the concept, the algorithm was implemented as a background service for Android-based handheld devices. The RSS values and position solution from the T-PN are automatically collected as pairs if they satisfy the requirements. The position information is converted from geodetic coordinates LLH to ENU coordinates, and paired with the corresponding RSS values. The pairs are checked for RSS ambiguity problem. If the ambiguity is detected, the method provided in Section 3.3 is utilized to fix this problem and improve the accuracy of the pairs in the database. Nonlinear iterative LSQ is used for estimation of the AP location, PPs, and their accuracies if multiple measurements from the same APs are collected. Dilution of Precision (DOP) (Langley 1999), which is an efficient indictor for evaluating the geometric distribution of measurements, is also calculated after the LSQ. The efficiency of DOP for performance evaluation will be shown in Section 3.6.1.1. If the computed results pass the verification of the criteria (such as range check for the path loss exponent) which will be discussed in Section 3.4.2.3, and no information about this AP is found in the database, this AP information from the LSQ results is recorded in the database. The details about the verification is given in Section 3.4.2.3. If the AP information is already present in the database, the computed results are used to update AP information in the database. This update process is a significant part of autonomous crowdsourcing.

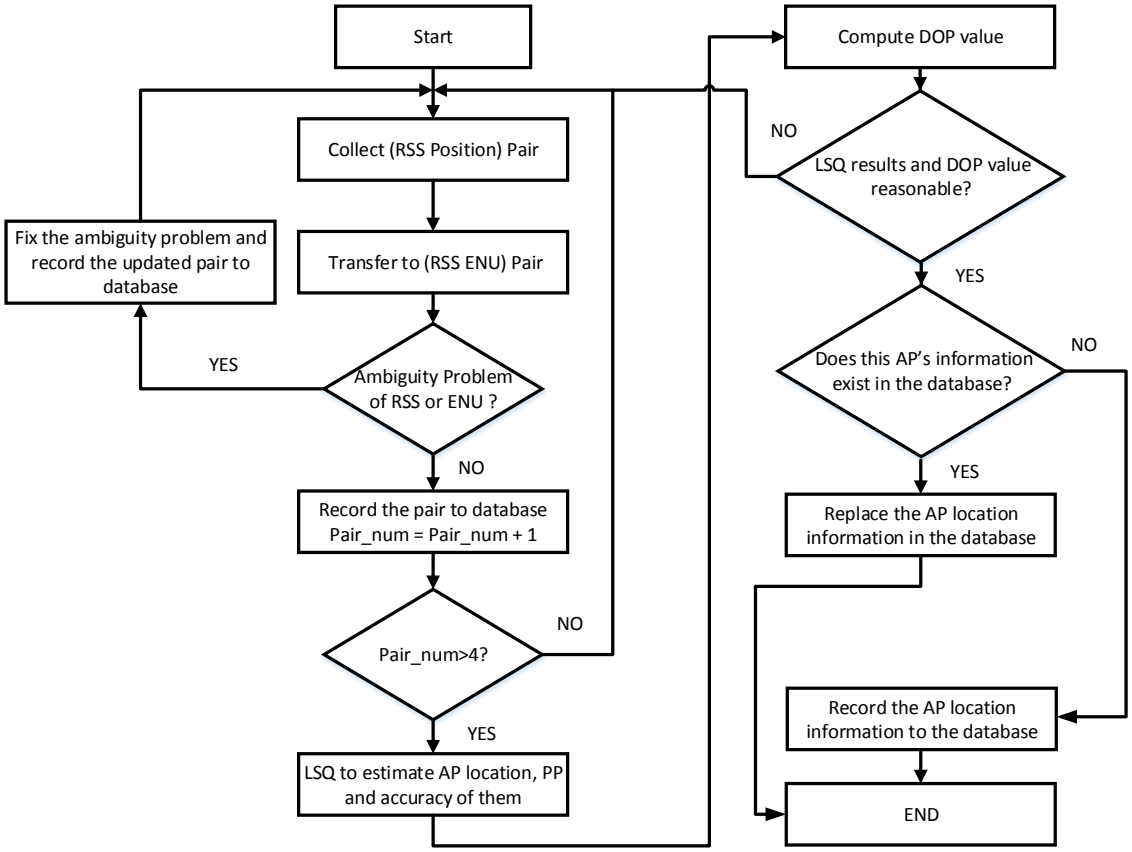


Figure 3-3 Flow chart of background survey service in the trilateration-based WPS.

3.4.2 AP Localization and PPs Estimations

This section details the algorithm of AP localization and PPs estimation and is divided into three subsections: a propagation model, the LSQ-based estimation for AP locations and PPs, and the LSQ results assessment.

3.4.2.1 Propagation Model

The typical path loss model follows the distance power law:

$$P_r = P_0 - 10n\log_{10}(d / l_0) + X_\sigma \tag{3-3}$$

where P_r is the RSS value received at the receiver in dBm at a distance d from the transmitter; P_0 is the RSS value with distance l_0 from the transmitter; n is the path loss exponent with typical values in the range of 2.0 – 6.0 indoors; and X_σ represents the shadow noise which is modeled as a Gaussian random variable with zero mean. Equation (3-3) can be simplified by averaging as follows:

$$RSS = -10\, n\, log_{10}(d) - A \tag{3-4}$$

where $A = -mean(P_0(l_0 = 1m))$, and the distance between the AP located at (x_0, y_0) and the measurement point (x_i, y_i) is defined as

$$d_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2} \tag{3-5}$$

Note that there are other propagation models that consider the effect of walls and floors (Bahl and Padmanabhan 2000) (Lott and Forkel 2001). However, they are not suitable for real-time AP localizations because a priori information of walls and floors are usually unavailable. The walls and floors can affect the estimation of PPs. Unfortunately, floor plans are not always available. For example, the floor plans of many older buildings can be unreliable or, in some cases, even unknown. Further, individuals at times cannot download the floor plan of a building quickly due to some technical problems. In this research, we design the system to provide a general and accurate positioning solution without depending on much additional information, such as a floor plan. The advantage of this system, when compared with other floor-plan-depended systems, is that it can work well without a floor plan.

3.4.2.2 LSQ-Based Estimation for AP Locations and PPs

3.4.2.2.1 First Design for LSQ

The goal in this subsection is to estimate AP locations and PPs by using observations (RSS values) with the position information from MEMS-based navigation solution (e.g. T-PN). In many cases, MEMS-based navigation solution cannot provide the accuracy information about the position solution. In this case, MEMS-based position solutions can only be considered error-free information, and not to be included in the observations. In the designed LSQ of this subsection, only RSS values are used in the observation vector, which can work for all MEMS-based navigation solutions. For convenience, this designed LSQ is called LSQ1. In the next subsection, both RSS values and MEMS-based derived positions are included in the observation vector for the LSQ, which can be used when the position accuracy is also provided by the MEMS-based navigation solution and will be called LSQ2.

In LSQ1, the state vector to estimate AP locations (x_0 and y_0) and PPs (n and A) is $\mathbf{x} = [x_0, y_0, n, A]^T$, while the observation vector is $\mathbf{z} = \mathbf{RSS}$. The nonlinear observation model using LSQ is provided in Equation (3-6), which combines Equation (3-4) and Equation (3-5) and adds measurement error vector $\mathbf{v}$.

$$\mathbf{RSS} = -10n\log_{10}\left(\sqrt{(x_0 - \mathbf{x_u})^2 + (y_0 - \mathbf{y_u})^2}\right) - A + \mathbf{v} \tag{3-6}$$

where $\mathbf{RSS} = [RSS_1, RSS_2, ..., RSS_k]^T$ is an RSS vector for k measurement points, $\mathbf{x_u} = [x_1, x_2, ..., x_k]^T$, and $\mathbf{y_u} = [y_1, y_2, ..., y_k]^T$. The initial $\mathbf{x} = [mean(\mathbf{x_u}), mean(\mathbf{y_u}), 3, 35]^T$ with 3 and 35 as the typical values for n and A in an indoor environment. Coordinates (x_i, y_i) of the

measurement points are provided by the T-PN solution. The equation of the design matrix can be obtained by comparing Equation (3-6) with the LSQ observation model, and is shown as follows:

$$h(\mathbf{x}) = -10nlog_{10}\left(\sqrt{(x_0 - \mathbf{x_u})^2 + (y_0 - \mathbf{y_u})^2}\right) - A \tag{3-7}$$

The derivative of Equation (3-7) is the design matrix and is provided below

$$\mathbf{H} = \frac{dh(\mathbf{x})}{d\mathbf{x}} = \begin{pmatrix} \frac{-10n(x_0 - x_{u_1})}{d_1^2 ln10} & \cdots & \frac{-10n(x_0 - x_{u_k})}{d_k^2 ln10} \\ \frac{-10n(y_0 - y_{u_1})}{d_1^2 ln10} & \cdots & \frac{-10n(y_0 - y_{u_k})}{d_k^2 ln10} \\ -10log_{10}(d_1) & \cdots & -10log_{10}(d_k) \\ -1 & \cdots & -1 \end{pmatrix} \tag{3-8}$$

As discussed in Chapter 2, the measurement covariance matrix can be written as:

$$\mathbf{R} = \sigma_0^2 \mathbf{Q_R} \tag{3-9}$$

where σ_0^2 is a-priori variance factor, and $\mathbf{Q_R}$ is the cofactor matrix of $\mathbf{R}$. $\mathbf{Q_R}$ is a diagonal matrix because the RSS values are independent for all the measurements, and is given by

$$\mathbf{Q_R} = diag(Q_{R,11}, Q_{R,22}, ..., Q_{R,kk})^T \tag{3-10}$$

where $Q_{R,11}, Q_{R,22}, ..., Q_{R,kk}$ are the diagonal elements of $\mathbf{Q_R}$. Note that σ_0^2 is often not provided or, if provided, it is unreliable. Therefore, one empirical value is pre-set for σ_0^2 at first. $\mathbf{Q_R}$ is an identity matrix if the weights are equal and the algorithm is a simple LSQ. On the other hand, if

the weights are not equal, the algorithm is called a weighed LSQ. In this case, RSS values can be used as weights for the measurement variances as given in Equation (3-11).

$$Q_{R,ii} = \frac{RSS_i}{sum(RSS_i)}, i = 1, 2, ...k \tag{3-11}$$

After the parameters are set for the LSQ estimation, the LSQ results are calculated by using the equations discussed in Chapter 2.

3.4.2.2.2 Second Design for LSQ

In the previous subsection, LSQ1 is designed by only using RSS values in the observation vector. For the LSQ2 in this subsection, both the RSS values and the MEMS derived positions are considered in the observation vector. LSQ2 can be used if the MEMS-based navigation solution provides the position accuracy along with the position. In LSQ2, the observation model can be obtained by rewriting Equation (3-6), and is given by

$$(x_0 - \mathbf{x_u})^2 + (y_0 - \mathbf{y_u})^2 - 10^{-\frac{\mathbf{RSS}+A}{5n}} = 0 \tag{3-12}$$

where $\mathbf{RSS} = [RSS_1, RSS_2, ..., RSS_k]^T$ is the observed RSS vector collected by handheld devices for k measurement points; $\mathbf{x_u} = [x_1, x_2, ..., x_k]^T$ and $\mathbf{y_u} = [y_1, y_2, ..., y_k]^T$ are the position coordinates from the MEMS-based navigation solution (e.g. T-PN). Therefore, the observed vector is rewritten as $\mathbf{L} = \left[\mathbf{x_u}^T, \quad \mathbf{y_u}^T, \quad \mathbf{RSS}^T\right]^T$. The state vector is the same as before, which is $\mathbf{x} = [x_0, y_0, n, A]^T$. Equation (3-12) is a combined (implicit) LSQ model of the form:

$$f(\mathbf{x}, \mathbf{L}) = 0 \tag{3-13}$$

The vector function f represents r equations relating n observations and u unknowns.

If the vector function f is nonlinear, Taylor expansion is used to approximately linearize these functions. The expanded point is the initial approximation to the state vector ($\mathbf{x}^0$), and the measured values of the observation vector ($\mathbf{L}_{obs}$) with the covariance matrix $\mathbf{C}_{\mathbf{l}_{obs}}$. The linearized model is given as (El-Sheimy 2000):

$$f(\mathbf{x},\mathbf{L}) = f(\mathbf{x}^0,\mathbf{L}_{obs}) + \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}^0,\mathbf{L}_{obs}} \boldsymbol{\delta} + \left.\frac{\partial f}{\partial \mathbf{L}}\right|_{\mathbf{x}^0,\mathbf{L}_{obs}} \mathbf{v} = 0 \tag{3-14}$$

or

$$\mathbf{w} + \mathbf{A}\boldsymbol{\delta} + \mathbf{B}\mathbf{v} = 0 \tag{3-15}$$

where $\mathbf{w} = f(\mathbf{x}^0,\mathbf{L}_{obs})$ is called the misclosure vector; and $\mathbf{A} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}^0,\mathbf{L}_{obs}}$ and $\mathbf{B} = \left.\frac{\partial f}{\partial \mathbf{L}}\right|_{\mathbf{x}^0,\mathbf{L}_{obs}}$ are called the design matrices. Lagrange's method is used to solve for Equation (3-15), and the result is given by

$$\begin{aligned}
\hat{\boldsymbol{\delta}} &= -\left(\mathbf{A}^{\mathbf{T}}(\mathbf{B}\mathbf{P}^{-1}\mathbf{B}^{\mathbf{T}})^{-1}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathbf{T}}\left(\mathbf{B}\mathbf{P}^{-1}\mathbf{B}^{\mathbf{T}}\right)^{-1}\mathbf{w} \\
\hat{\mathbf{k}} &= \left(\mathbf{B}\mathbf{P}^{-1}\mathbf{B}^{\mathbf{T}}\right)^{-1}\left(\mathbf{A}\hat{\mathbf{x}} + \mathbf{w}\right) \\
\hat{\mathbf{v}} &= -\mathbf{P}^{-1}\mathbf{B}^{\mathbf{T}}\hat{\mathbf{k}}
\end{aligned} \tag{3-16}$$

where $\mathbf{P} = \mathbf{C}_{\mathbf{l}_{obs}}^{-1}$, and $\hat{\mathbf{k}}$ is the Lagrange multiplier. The covariance matrices for $\mathbf{w}$, $\hat{\boldsymbol{\delta}}$, and $\hat{\mathbf{v}}$ are expressed as:

$$\mathbf{C_w} = \left(\mathbf{BP^{-1}B^T}\right)^{-1}$$
$$\mathbf{C_{\hat{\delta}}} = \left(\mathbf{A^T(BP^{-1}B^T)^{-1}A}\right)^{-1} \tag{3-17}$$
$$\mathbf{C_{\hat{v}}} = \mathbf{P^{-1}B^T}\left(\mathbf{BP^{-1}B^T}\right)^{-1}\left[\mathbf{BP^{-1}} - \mathbf{A}\left(\mathbf{A^T(BP^{-1}B^T)^{-1}A}\right)^{-1}\mathbf{A^T}\left(\mathbf{BP^{-1}B^T}\right)^{-1}\mathbf{BP^{-1}}\right]$$

The adjusted quantities are given by

$$\hat{\mathbf{x}} = \mathbf{x}^0 + \hat{\boldsymbol{\delta}}$$
$$\hat{\mathbf{L}} = \mathbf{L} + \mathbf{L}_{obs} \tag{3-18}$$

Usually, the estimation for the state vector is an iterative process. In this case, the expanded point in Equation (3-14) is changed to the previous estimate of state vector ($\hat{\mathbf{x}}_{(i-1)}$), and $\mathbf{w}$ in Equation (3-15) is changed to

$$\mathbf{w} = f\left(\hat{\mathbf{x}}_{(i-1)}, \mathbf{L}_{obs}\right) + \mathbf{B}\left(\mathbf{L}_{obs} - \hat{\mathbf{L}}_{(i-1)}\right) \tag{3-19}$$

where $\hat{\mathbf{L}}_{(i-1)}$ is the previous estimate of observation vector. Equations (3-14) ~ (3-18) are repeated until $\hat{\boldsymbol{\delta}}_{(i+1)} - \hat{\boldsymbol{\delta}}_{(i)}$ approaches 0. For more details about the solution of implicit LSQ model, see (El-Sheimy 2000).

- Parameter Determination

By comparing (3-12) with (3-13), the functional model can be written as:

$$f\left(\mathbf{x}, \mathbf{L}\right) = (x_0 - \mathbf{x_u})^2 + (y_0 - \mathbf{y_u})^2 - 10^{-\frac{\mathbf{RSS}+A}{5n}} \tag{3-20}$$

where $\mathbf{x} = [x_0, y_0, n, A]^T$ and $\mathbf{L} = \begin{bmatrix} \mathbf{x_u}^T & \mathbf{y_u}^T & \mathbf{RSS}^T \end{bmatrix}^T$. The vector function f represents k equations relating $3k$ observations and 4 unknowns. k represents the number of measurement points. If $k > 4$, the number of degrees of freedom is greater than 0, Equation (3-13) can be solved by using LSQ. The covariance matrix of $\mathbf{L}$ is given by

$$\mathbf{C_L} = diag\left(\begin{bmatrix} \sigma_{x_1}^2 & \sigma_{y_1}^2 & \sigma_{RSS_1}^2 & \cdots & \sigma_{x_k}^2 & \sigma_{y_k}^2 & \sigma_{RSS_k}^2 \end{bmatrix}\right) \tag{3-21}$$

where $\sigma_{x_i}^2$, $\sigma_{y_i}^2$, and $\sigma_{RSS_i}^2$ are the variances of x_i, y_i, and RSS_i. The accuracy of these variances will affect the final estimation result. If a MEMS-based navigation solution cannot provide an appropriate estimate of the position variances, the result of this method may be worse than the one only using RSS values as observations. The design matrices $\mathbf{A}$ and $\mathbf{B}$ are given by

$$\mathbf{A} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}^0, \mathbf{L}_{obs}} = \begin{bmatrix} 2\left(x_0^0 - x_1\right) & 2\left(y_0^0 - y_1\right) & -10^{-\frac{RSS_1 + A^0}{5n^0}} \cdot \frac{\ln(10)\left(RSS_1 + A^0\right)}{5\left(n^0\right)^2} & 10^{-\frac{RSS_1 + A^0}{5n^0}} \cdot \frac{\ln(10)}{5n^0} \\ \cdots & \cdots & \cdots & \cdots \\ 2\left(x_0^0 - x_k\right) & 2\left(y_0^0 - y_k\right) & -10^{-\frac{RSS_k + A^0}{5n^0}} \cdot \frac{\ln(10)\left(RSS_k + A^0\right)}{5\left(n^0\right)^2} & 10^{-\frac{RSS_k + A^0}{5n^0}} \cdot \frac{\ln(10)}{5n^0} \end{bmatrix}_{k \times 4} \tag{3-22}$$

and

$$\mathbf{B} = \left.\frac{\partial f}{\partial \mathbf{L}}\right|_{\mathbf{x}^0, \mathbf{L}_{obs}} = \begin{bmatrix} 2\left(x_1 - x_0^0\right) & 2\left(y_1 - y_0^0\right) & 10^{-\frac{RSS_1 + A^0}{5n^0}} \cdot \frac{\ln(10)}{5n^0} & & \\ & & & \ddots & \\ & & & & 2\left(x_k - x_0^0\right) \quad 2\left(y_k - y_0^0\right) \quad 10^{-\frac{RSS_k + A^0}{5n^0}} \cdot \frac{\ln(10)}{5n^0} \end{bmatrix}_{k \times 3k} \tag{3-23}$$

Then, Equations (3-14) ~ (3-18) are repeated until $\hat{\boldsymbol{\delta}}_{(i+1)} - \hat{\boldsymbol{\delta}}_{(i)}$ approaches 0.

3.4.2.3 LSQ Results Assessment

To improve the estimation performance for AP locations and PPs, it is important to ensure that the algorithm is converged and that the terms, listed below, are checked.

- Path loss exponent n in Equation (3-4)
- Constant value A in Equation (3-4)
- Reasonable AP location
- DOP value

The typical ranges of the path loss exponent n and the constant A in Equation (3-4) are 2.0 - 6.0 and 0 – 100, respectively. The estimation result is ignored if it is not located within these typical ranges. According to the typical propagation model and field tests, an AP always stays within 200 meters of the WiFi measurement points. Therefore, the estimation results are ignored and deemed unreliable if the estimated AP location is far away from the measurement points. The last value that needs to be evaluated is the DOP value of the measurements. For the designed LSQ, the horizontal DOP value (Petovello 2012) is given in

$$\text{DOP} = \sqrt{(\mathbf{Q_P})_{11} + (\mathbf{Q_P})_{22}} \tag{3-24}$$

where $(\bullet)_{ii}$ represents the element in the i^{th} row, i^{th} column of a matrix and $\mathbf{Q_P}$ is calculated by

$$\mathbf{Q_P} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{Q_R}^{-1}\mathbf{H}\right)^{-1} \tag{3-25}$$

where $\mathbf{H}$ and $\mathbf{Q}_{\mathbf{R}}$ are the design matrix and the cofactor matrix of $\mathbf{R}$, respectively. For details of the DOP calculation and application, please refer to (Langley 1999) (Petovello 2012). The similar applications of DOPs for WiFi navigation are discussed in (Yu 2012) (Zirari et al. 2010). The estimated results for the AP locations and PPs are used only when the DOP values are less than the pre-set threshold of 4.0.

3.4.3 Autonomous Crowdsourcing

The proposed system is a natural crowdsourcing system, and ensures the creation and maintenance of the database automatically and efficiently. In traditional methods (Cheng et al. 2005), trained professionals are employed to survey an area to obtain a robust and precise database of AP locations. After the initial creation, the database needs sporadic maintenance due to changes in the environment. Furthermore, both survey and maintenance of the database cost time and labor, especially for a large area. The autonomous crowdsourcing-based approach is developed to reduce the cost of building and maintaining the database of the AP locations and PPs. Regular smartphone users can collect RSS values and corresponding positioning solutions from T-PN as measurements during their normal daily use of their mobile devices. When enough measurements are collected, they are used automatically to estimate the AP locations and PPs. The estimation results are, then, updated to the database by autonomous crowdsourcing without additional operations. The estimation result is recalculated and the database is updated as more measurements of the AP become available. The aim of crowdsourcing is to maintain the accuracy of the AP information in the database (locations and PPs) for future positioning usage. The crowdsourcing-based systems usually face some problems such as: (1) hardware differences of various devices and (2) different mounting places of devices. For the problem (1), if the uploaded data is large enough, final

estimated database can achieve the best performance by using some algorithms to process the large data. Also, WiFi RSS biases in different devices are also estimated in the tightly-coupled integration of WiFi and MEMS sensors as will be discussed in Chapter 5. The estimated RSS biases can also be used to solve the problem of hardware differences. For the problem (2), T-PN can provide navigation solutions in various modes/mounting places, therefore, different mounting places do not affect the proposed crowdsourcing-based systems. However, mounting places may affect the system if the navigation solution provider cannot provide solutions in various mounting places. In this case, mode detection is required before using the navigation solution to update the database.

3.5 WiFi Positioning Service

The flow chart of the WiFi positioning service based on trilateration is shown in Figure 3-4. In the trilateration-based system, iterative nonlinear LSQ is used for WiFi positioning if the AP number is larger than the threshold "AP_th". AP locations and ranges between the user and APs are necessary information for the user position estimation. AP locations are obtained from the background survey service, as discussed in Section 3.4. The ranges are calculated by substituting the real-time collected RSS values to the propagation model (Subsection 3.4.2.1), whose parameters are from the automatically generated database. To estimate user position (x_u and y_u), the state vector is set to $\mathbf{x}=[x_u, y_u]^T$. The height is not considered in the state vector, because it cannot be accurately estimated only using WiFi RSS values. In the design, the measurement vector $\mathbf{z}$ is the range between user and AP ($\mathbf{z}=\mathbf{range}$), which is calculated from RSS values by using the propagation model.

The nonlinear observation model using LSQ is provided as follows:

$$\mathbf{range} = \sqrt{(x_{user} - \mathbf{x}_{AP})^2 + (y_{user} - \mathbf{y}_{AP})^2} + \mathbf{v} \tag{3-26}$$

where $\mathbf{range} = [range_1, range_2, ..., range_k]^T$ is a range vector for k measurement points, and $\mathbf{x}_{AP} = [x_{AP1}, x_{AP2}, ..., x_{APk}]^T$ and $\mathbf{y}_{AP} = [y_{AP1}, y_{AP2}, ..., y_{APk}]^T$ are 2D coordinate vectors from the ENU coordinates of the AP locations. $\hat{\mathbf{x}} = [mean(\mathbf{x}_{AP}), mean(\mathbf{y}_{AP})]^T$ is set as the initial values for the iterative LSQ.

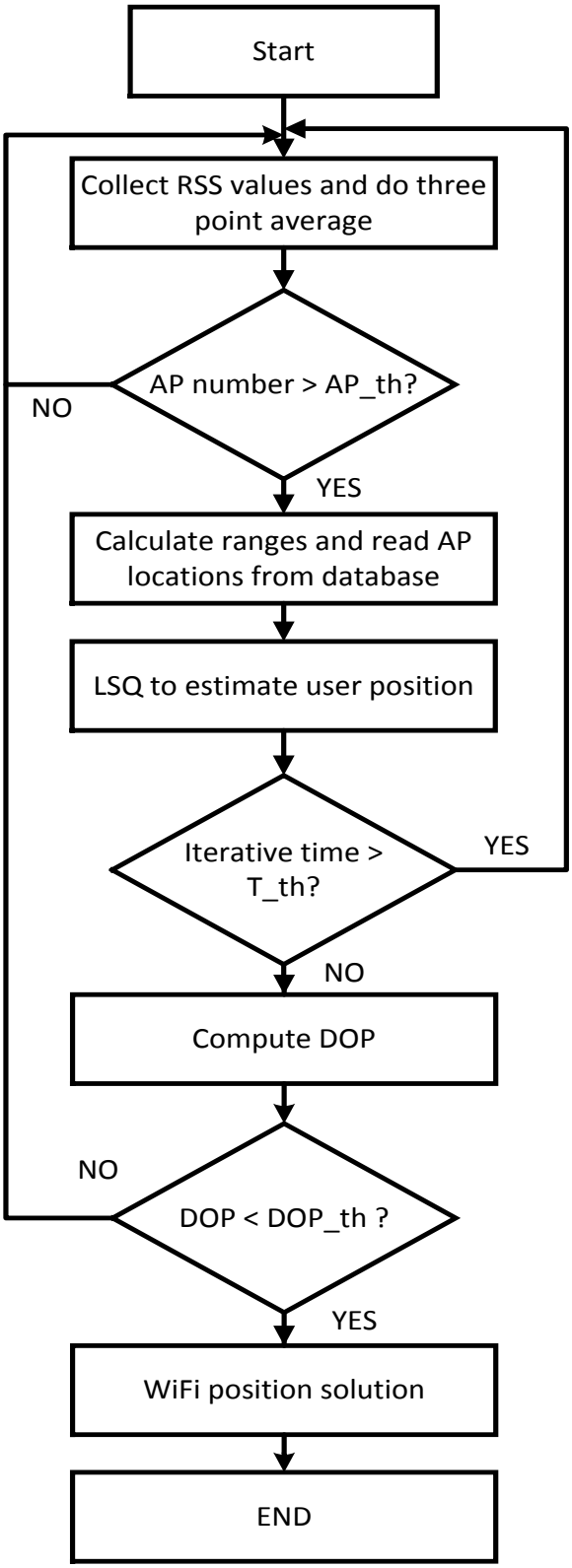


Figure 3-4 Flow chart of WiFi positioning service in the trilateration-based WPS.

The equation of the design matrix can be obtained by comparing Equation (3-26) with the LSQ observation model as follows:

$$h(\mathbf{x}) = \sqrt{(x_{user} - \mathbf{x}_{AP})^2 + (y_{user} - \mathbf{y}_{AP})^2} \tag{3-27}$$

Therefore, the design matrix is given by

$$\mathbf{H} = \frac{dh(\mathbf{x})}{d\mathbf{x}} = \begin{pmatrix} \frac{-(x_{user} - x_{AP1})}{RANGE_1} & \frac{-(x_{user} - x_{AP2})}{RANGE_2} & \cdots & \frac{-(x_{user} - x_{APk})}{RANGE_k} \\ \frac{-(y_{user} - y_{AP1})}{RANGE_1} & \frac{-(y_{user} - y_{AP2})}{RANGE_2} & \cdots & \frac{-(y_{user} - y_{APk})}{RANGE_k} \end{pmatrix} \tag{3-28}$$

where $RANGE_1$, $RANGE_2$, …, $RANGE_k$ represent the k elements about the range information in the vector $\mathbf{RANGE} = [RANGE_1, RANGE_2, ..., RANGE_k]^T$, which is given by

$$\mathbf{RANGE} = \sqrt{(x_{user} - \mathbf{x}_{AP})^2 + (y_{user} - \mathbf{y}_{AP})^2} \tag{3-29}$$

In this LSQ, $\mathbf{Q_R}$ is a diagonal matrix because the ranges, which are calculated from RSS values, do not depend on each other, and is given by

$$\mathbf{Q_R} = diag(Q_{R,11}, Q_{R,22}, ..., Q_{R,kk})^T \tag{3-30}$$

where $Q_{R,11}, Q_{R,22}, ..., Q_{R,kk}$ are the diagonal elements of $\mathbf{Q_R}$. The setting of $\mathbf{Q_R}$ is from the estimated accuracies of AP locations in the database. After the parameters are set for the LSQ estimation, LSQ results are calculated by using the equations discussed in Chapter 2.

As mentioned earlier, there are some criteria that should be met to ensure the performance of the improved WiFi positioning algorithm. First of all, the number of observed APs must be over a minimum number to ensure the accuracy of WPS. The next criterion is related to the DOP value, which should be less than a threshold to make sure the distribution of the measurements are appropriate. Finally, if the iteration time goes beyond a pre-set threshold, the algorithm will stop the LSQ for this epoch, and process the data for the next epoch. All of the thresholds stated here are determined by the experimental tests.

3.6 Test Results and Performance Analysis

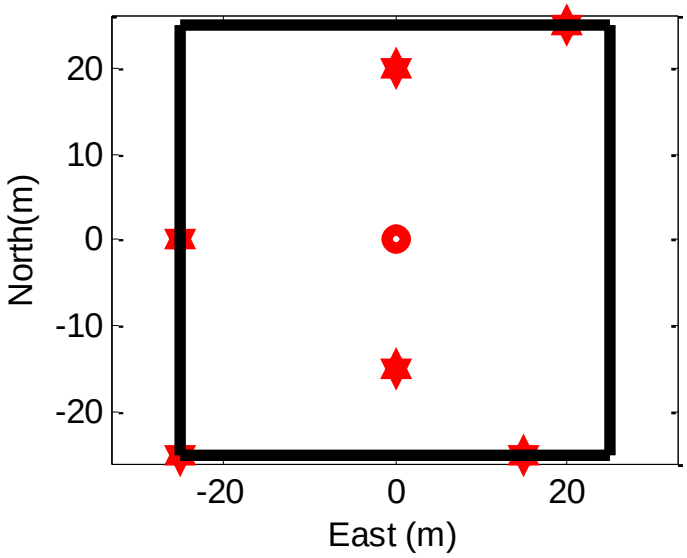
3.6.1 Performance of AP Localization and PPs Estimation

3.6.1.1 Simulations

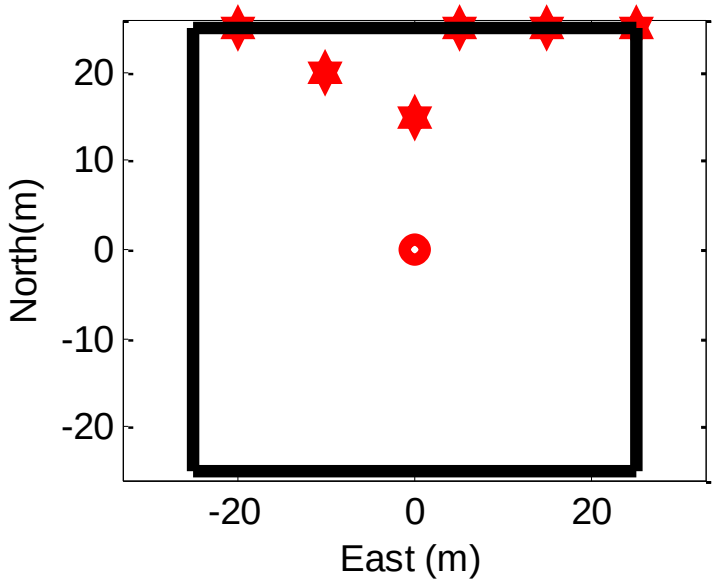
A simulation, in a 50 m × 50 m square, is conducted in this section to evaluate the performance of the proposed algorithm of AP localization and PPs estimation. In this subsection, the first design of LSQ is evaluated as an example. Two different geometrical distributions of measurements are simulated as shown in Figure 3-5. Configuration as depicted in Figure 3-5(a) has a smaller DOP value because it has better distributed measurements. Simulated RSS values are generated by using the propagation model in Equation (3-3) with l_0 set to 1m, A set to 30dBm, and n set to 3. The Gaussian random variable X_σ in Equation (3-3) is simulated as a statistical variable, which has a mean value of 0, and a standard deviation of 2.

The simulated results for estimating AP locations and PPs are shown in Table 3-1. In the case of Figure 3-5(a), the estimation error of the AP location is about 3.6 meters and the relative error of PPs is about 20%. Due to the larger DOP value, Figure 3-5(b) has a poorer estimation performance

than Figure 3-5(a). This example shows that DOP is a significant indicator for the accuracy of AP localization. For the rest of the simulations, only the case in Figure 3-5(a) is discussed.



(a)



(b)

Figure 3-5 Simulation area.

Table 3-1 Simulated results of estimating AP locations and PPs

	East (m)	North (m)	Estimated n	Estimated A	AP Localization Error (m)	n Estimated Error	A Estimated Error
(a)	3.51	-0.73	2.57	37.15	3.59	14.33%	23.83%
(b)	-2.27	16.84	1.27	58.73	16.99	57.67%	95.77%
True Value	0	0	3	30	N/A	N/A	N/A

To compare the proposed algorithm with other methods, several methods are also implemented in this project. Table 3-2 shows the AP localization results of several methods as follows: (a) M1: average method in (Cheng et al. 2005); (b) M2: weighted average method in (Cheng et al. 2005); (c) M3: method in (Jahyoung and Hojung 2011); (d) M4: method in (Yu 2012); and (e) M5: the proposed method. The proposed method clearly show better performance than the other methods.

Table 3-2 AP localization results using different methods

Method	East (m)	North (m)	Error (m)
M1	-2.50	-3.33	4.17
M2	-2.73	-3.42	4.38
M3	5.31	-3.39	6.30
M4	4.31	0.37	4.32
M5	3.51	-0.73	3.59

In order to evaluate the performance of PPs estimation, several simulations are conducted with different PPs as shown in Table 3-3. To simulate different environments, the PP "n" is set to 2, 3, and 4; and "A" is set to 30 and 40. In different environments, the results show that the proposed method can actually estimate the PPs. It also illustrates that the proposed method can successfully cope with changes in the dynamic environment.

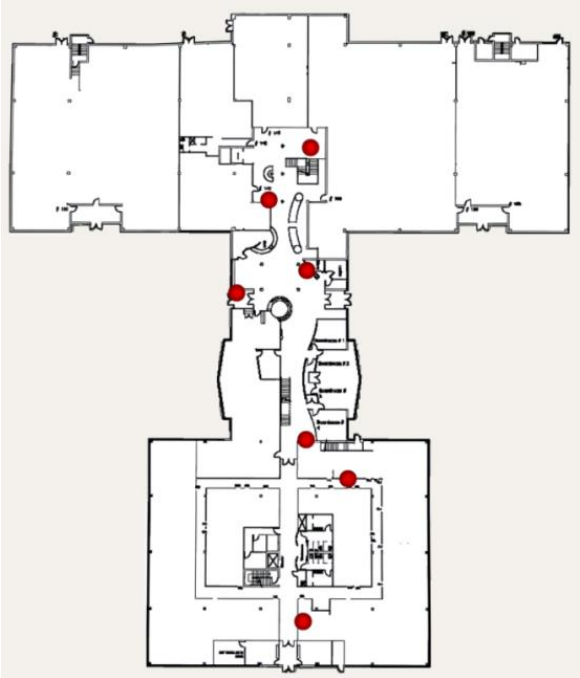
Table 3-3 Simulated results in different indoor environments

Set n; A	East (m)	North (m)	Estimated n	Estimated A	AP Localization Error (m)	n Estimated Error	A Estimated Error
2; 30	-2.04	0.55	2.16	28.04	2.11	8.00%	6.53%
3; 30	3.51	-0.73	2.57	37.15	3.59	14.33%	23.83%
4; 30	0.59	-0.63	3.78	32.83	0.86	5.50%	9.43%
2; 40	-2.78	0.10	1.77	43.78	2.78	11.50%	9.45%
3; 40	2.00	-0.26	3.30	35.99	2.02	10.00%	10.02%
4; 40	-0.38	0.10	3.72	44.47	0.39	7.00%	11.17%

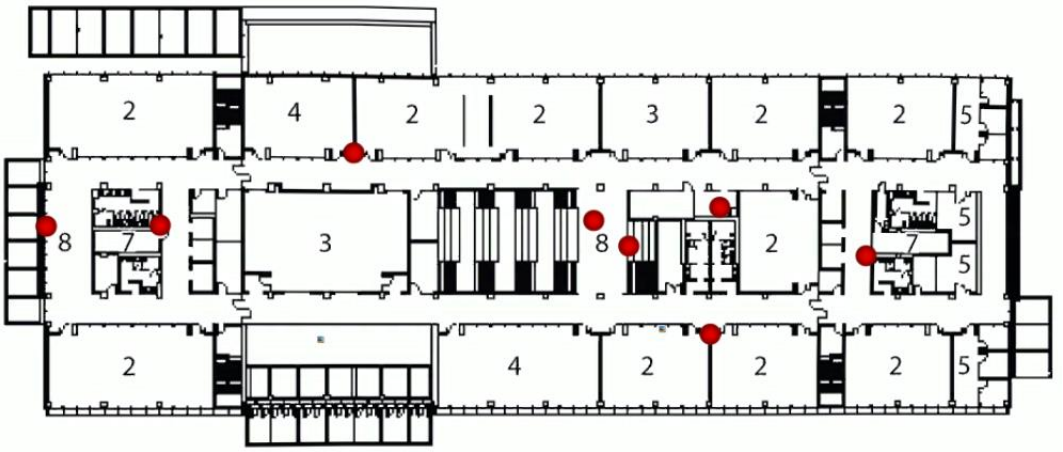
3.6.1.2 Field Experiments

This section discusses the setup, results, and analysis of the field experiments to evaluate the performance of the proposed algorithms. First, the design and setup of the experiments are explained. Then, several preliminary results of the real-world scenarios are tested and analyzed. Two proposed systems using LSQ1 and LSQ2 are evaluated and compared by field tests.

To evaluate the performance of the proposed systems in field environments, we implemented the algorithm as the background survey service on three Android-based Samsung Galaxy S III smartphones. Two evaluation sites were selected for the experiments which are shown in Figure 3-6. The first experimental site was building A (Alastair Ross Technology Centre, Calgary, about 100m × 70m), with seven location-known APs as shown in Figure 3-6(a). Building E (about 120m × 40m) with eight location-known APs is chosen as the second test site, as shown in Figure 3-6(b). Note that there were more APs in these two buildings, but they were not used for assessing the performance of AP localization. However, their locations and PPs are also estimated and recorded in the database for the use of WiFi positioning.



(a)

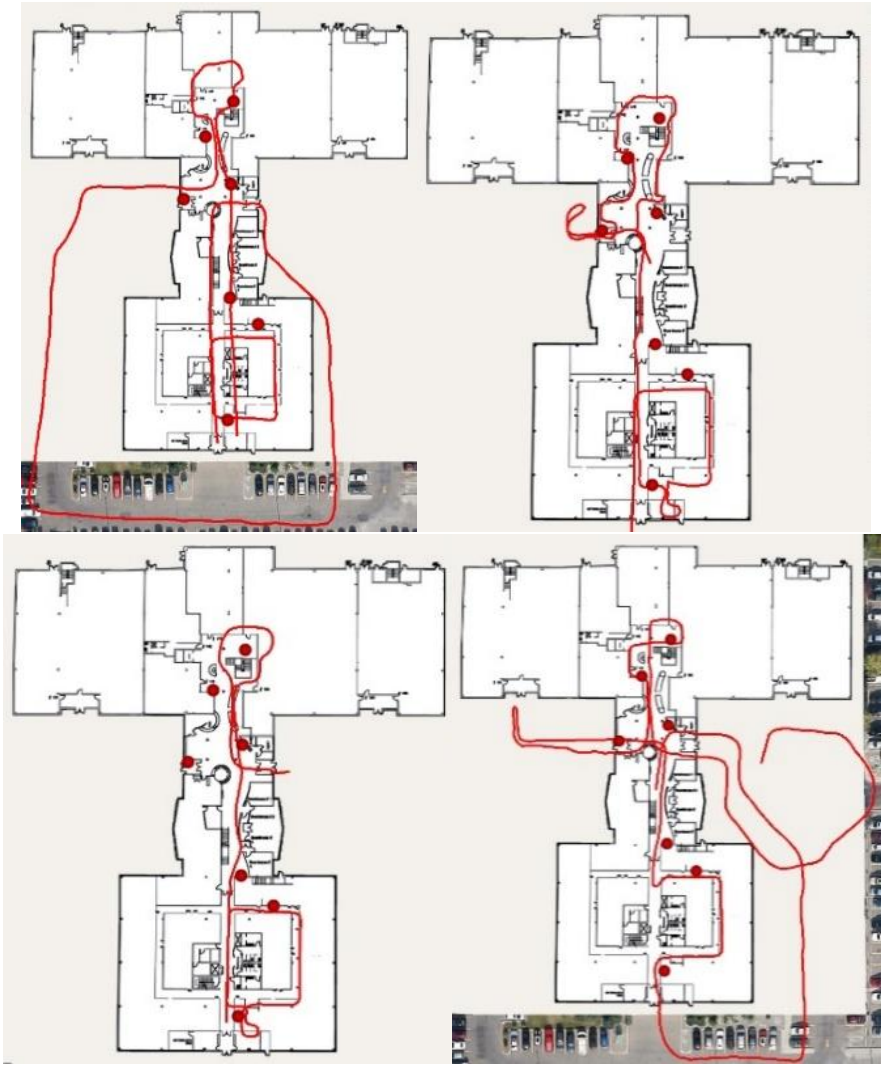


(b)

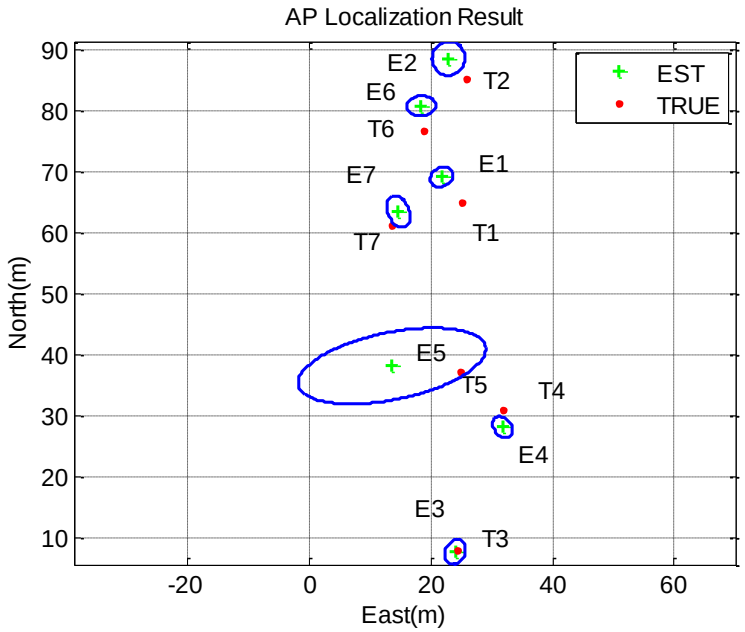
Figure 3-6 Experimental area (red circles = APs): (a) building A and (b) building E.

The experimental results of AP localization and PPs estimation in building A using LSQ1 are shown in Figure 3-7. In Figure 3-7(a), the red trajectories in four sub-figures are automatically generated by T-PN, which represents the paths taken by the user in building A. Figure 3-7(b), (c), and (d) show the final estimation results by using all four trajectories. The estimated and true locations of APs are shown in Figure 3-7(b). The blue ellipses in Figure 3-7(b) represents the standard confidence ellipses. For the 2D case, the standard confidence ellipse has a probability of 39.4% associated with it (Petovello 2012). In other words, only 39.4% of the points fall within the standard confidence ellipse. There are 3/7 APs located in the standard confidence ellipses, which is close to the reference. The estimation result is calculated by nonlinear LSQ, and its accuracy mainly depends on the fluctuation of RSS signals, the accuracy of T-PN solutions, and the geometrical distribution of measurement pairs. Figure 3-7(b) clearly shows that the estimated AP locations are close to the true values, which illustrates the efficiency of the proposed system. In Figure 3-7(c), the estimated path loss exponent “n” and constant “A” are located in typical ranges. The true values of PPs cannot be shown here because they are unknown in this environment.

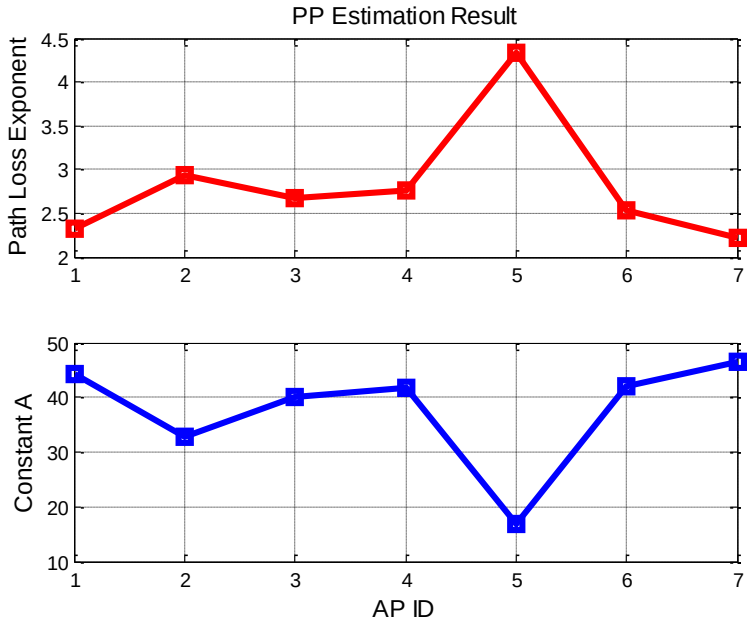
However, the efficiency of PPs estimation has been demonstrated in the simulation (Subsection 3.6.1.1). In Figure 3-7(d), the estimated AP localization error is close to the true value at most times, and the maximum difference between them is about 4 meters. Therefore, the estimated AP localization error is an efficient parameter to indicate the performance of AP localization. It is recorded in the database, and used as an indicator for the accuracy of AP locations.



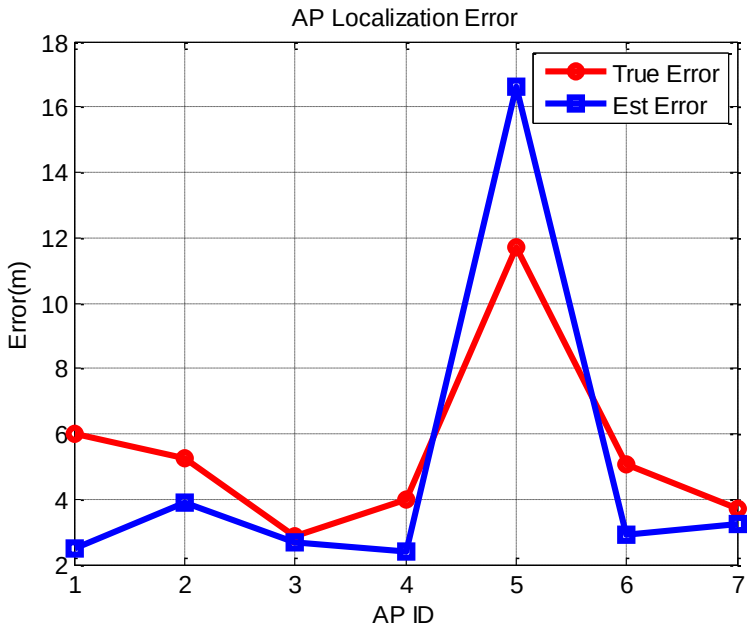
(a)


AP Localization Result
EST
TRUE
E2
T2
E6
T6
E1
E7
T1
T7
E5
T5
T4
E4
E3
T3
North(m)
East(m)
90
80
70
60
50
40
30
20
10
-20
0
20
40
60

(b)


PP Estimation Result
Path Loss Exponent
4.5
4
3.5
3
2.5
2
1
2
3
4
5
6
7
Constant A
50
40
30
20
10
AP ID

(c)



(d)

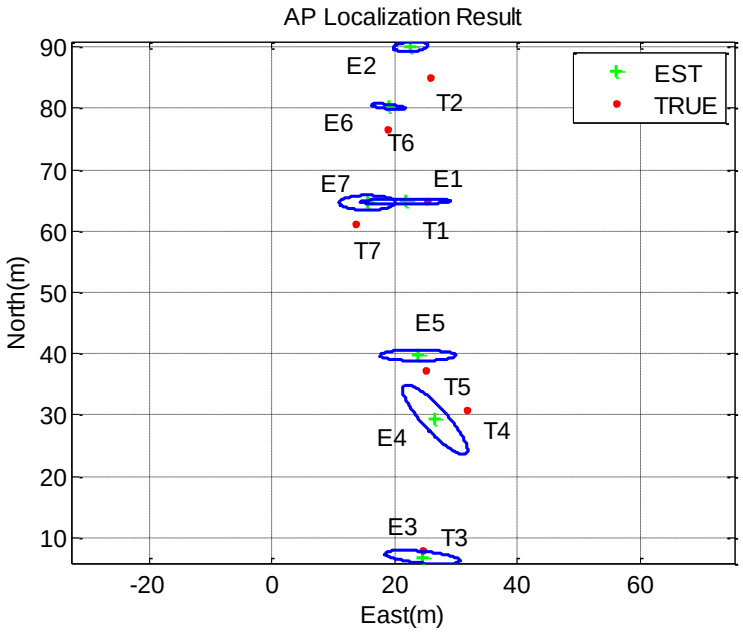
Figure 3-7 Results of AP localization and PPs estimation in building A using LSQ1: (a) four T-PN trajectories used for estimation, (b) the result of AP localization, (c) the result of PPs estimation, and (d) estimated and true 2D errors of AP localization.

Table 3-4 clearly depicts the trend where the increase in RSS and T-PN pairs improves the accuracy of AP localization. In Table 3-4, "AP Localization Error" represents the difference between the estimated AP location and the true value. "Accuracy Estimation Error" equals the difference between the estimated and true AP localization error, and is used to determine whether estimated AP localization error is an efficient indicator for the accuracy of AP localization. Note that both "AP Localization Error" and "Accuracy Estimation Error" are calculated in 2D space. Table 3-4 shows that "AP Localization Error" and "Accuracy Estimation Error" decrease as the number of trajectories increases. However, this does not apply if the measurement error of a trajectory is large, as was the case for trajectory 4 in which "AP Localization Error" increased.

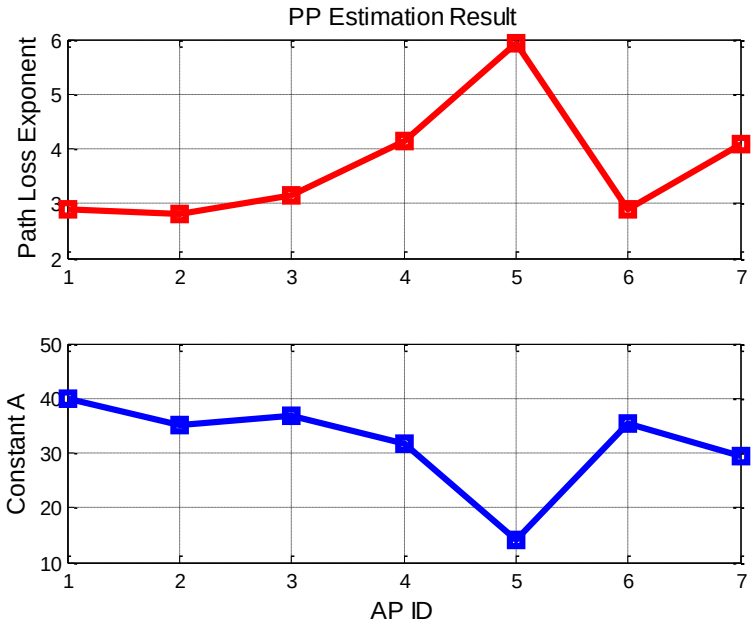
Table 3-4 AP localization results using LSQ1 in building A

Trajectory Number	Number of Estimated APs	AP Localization Error		Accuracy Estimation Error	
		MEAN (m)	RMS (m)	MEAN (m)	RMS (m)
1	7	6.34	6.65	3.80	5.82
2	7	5.72	5.89	2.56	3.44
3	7	5.27	5.47	2.82	3.46
4	7	5.51	6.14	2.03	2.55

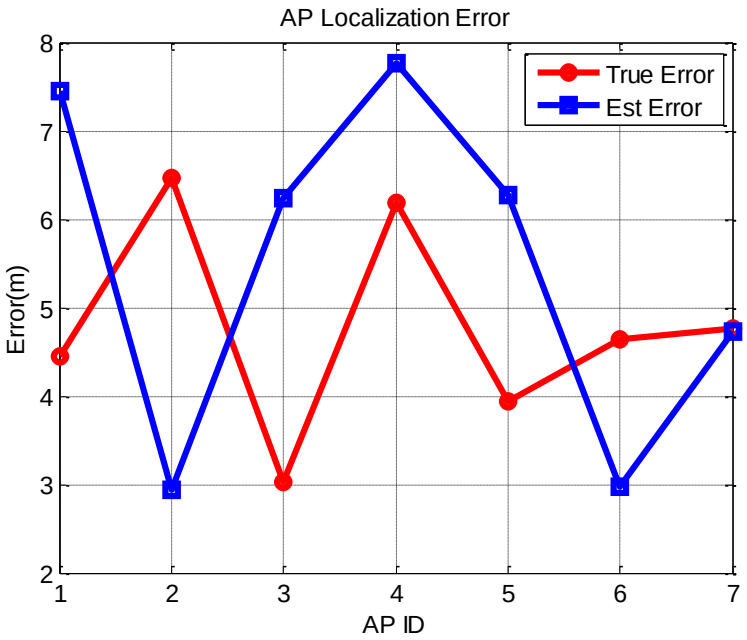
The experimental results of AP localization and PPs estimation in building A using LSQ2 are shown in Figure 3-8. The used trajectories as the same as the Figure 3-7 (a). Figure 3-8 (a), (b), and (c) show the final estimation results based on LSQ2 by using all four trajectories. The estimated and true locations of APs are shown in Figure 3-8 (a). Figure 3-8 (a) clearly shows that the estimated AP locations are close to the true values. The blue ellipses in Figure 3-8 (a) represents the standard confidence ellipses. There are 2/7 APs located in the standard confidence ellipses. In Figure 3-8 (b), the estimated path loss exponent "n" and constant "A" are located in typical ranges. In Figure 3-8 (c), the estimated AP localization error is not always close to the true value, however, it can be considered a rough estimate.


AP Localization Result
EST
TRUE
E2
T2
E6
T6
E7
E1
T1
T7
E5
T5
T4
E4
E3
T3
North(m)
East(m)
90
80
70
60
50
40
30
20
10
-20
0
20
40
60

(a)


PP Estimation Result
Path Loss Exponent
Constant A
AP ID
6
5
4
3
2
50
40
30
20
10
1
2
3
4
5
6
7

(b)



(c)

Figure 3-8 Results of AP localization and PPs estimation in building A using LSQ2: (a) the result of AP localization, (b) the result of PPs estimation; and (c) estimated and true 2D errors of AP localization.

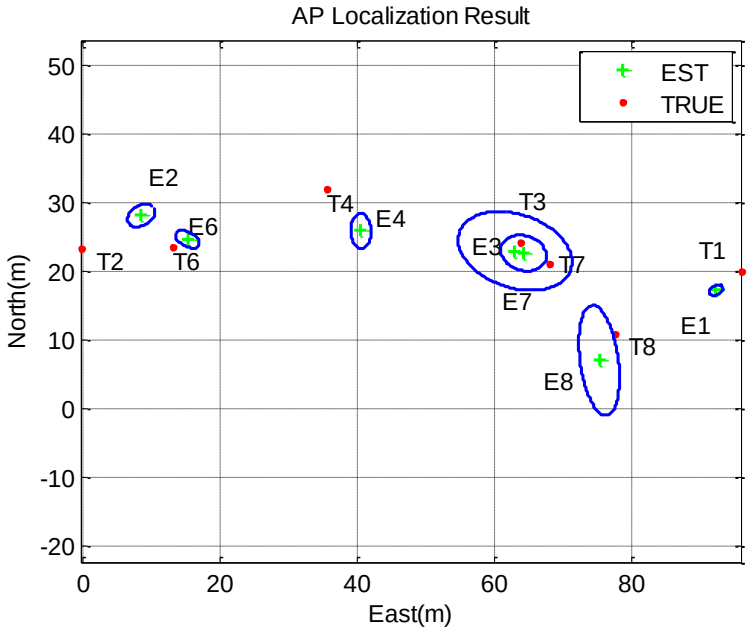
Table 3-5 summarizes the results of AP localizations by using different numbers of trajectories in building A. In Table 3-5, "AP Localization Error" and "Accuracy Estimation Error" decrease as the trajectories increase. Note that both "AP Localization Error" and "Accuracy Estimation Error" are calculated in the 2D space.

Table 3-5 AP localization results using LSQ2 in building A

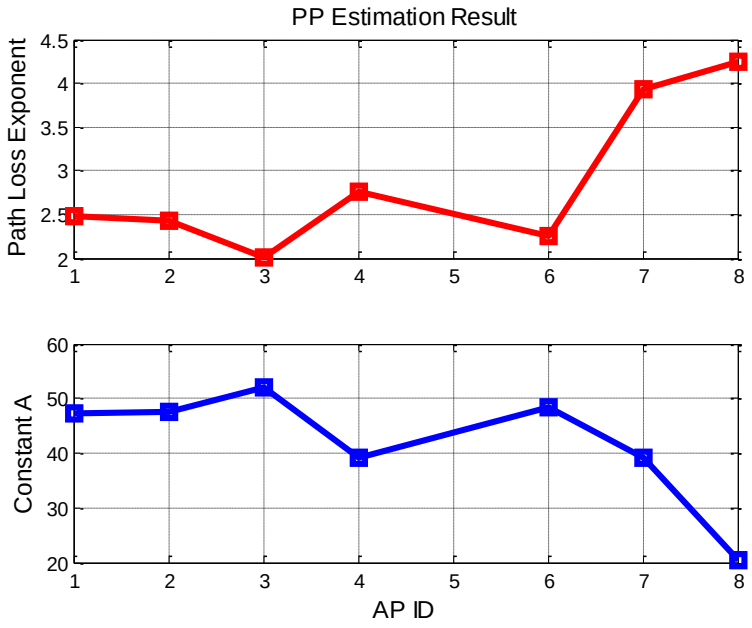
Trajectory Number	Number of Estimated APs	AP Localization Error		Accuracy Estimation Error	
		MEAN (m)	RMS (m)	MEAN (m)	RMS (m)
1	5	7.90	8.34	8.16	11.89
2	7	6.81	7.10	2.64	3.60
3	7	6.23	6.64	3.47	4.24
4	7	4.78	4.90	2.19	2.46

The second test for evaluating the performance of AP localization by using LSQ1 was conducted in building E, as shown in Figure 3-9. Six red trajectories were generated from the T-PN solution, used for AP localization as shown in Figure 3-9 (a). The results in Figure 3-9 (b), (c), and (d) were estimated by using all six trajectories. Figure 3-9 (b) and Figure 3-9 (c) demonstrate the efficiency of AP localizations and PPs estimation. Estimated AP localization error is not always a perfect indicator of the true values as shown in Figure 3-9 (d). However, since it is the only available value for the accuracy of APs, it can be considered a rough estimate.

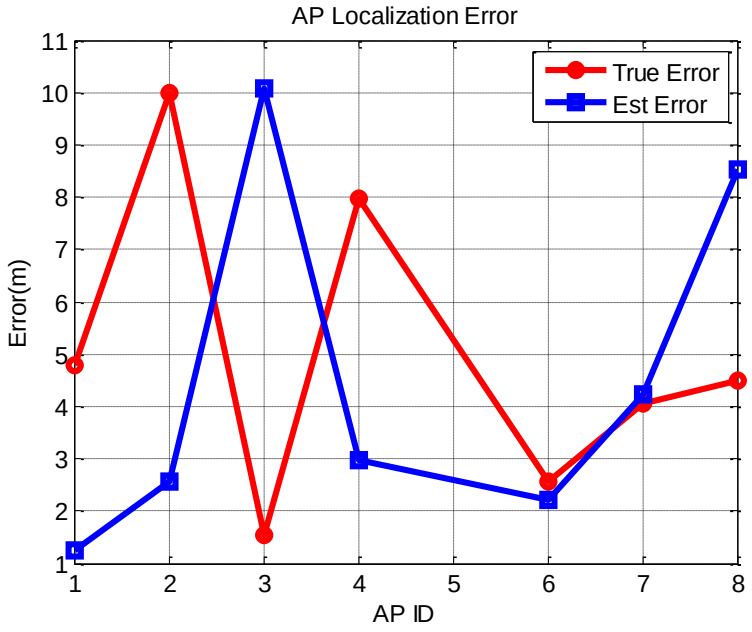
(a)


AP Localization Result
EST
TRUE
E1
E2
E3
E4
E6
E7
E8
T1
T2
T3
T4
T6
T7
T8
North(m)
East(m)
50
40
30
20
10
0
-10
-20
0
20
40
60
80

(b)



(c)



(d)

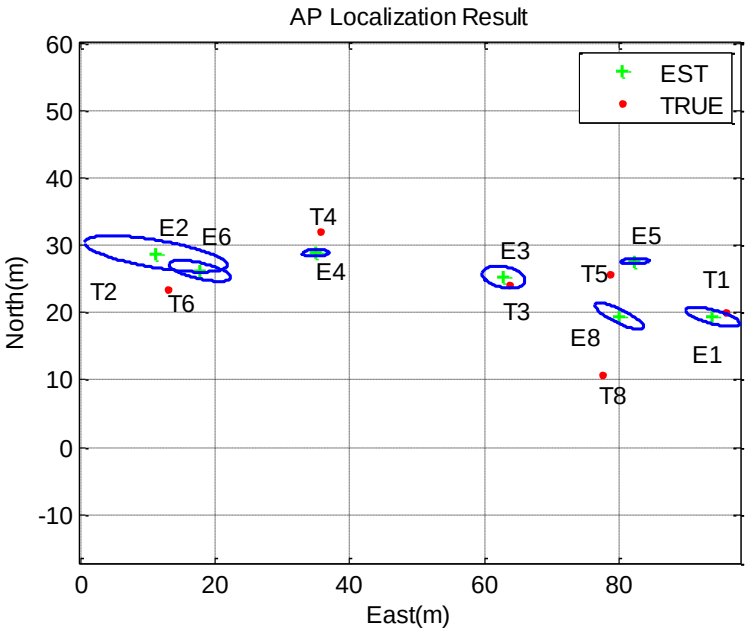
Figure 3-9 Results of AP localization and PPs estimation in building E using LSQ1: (a) six T-PN trajectories used for estimation, (b) the result of AP localization, (c) the result of PPs estimation, and (d) estimated and true 2D errors of AP localization.

Table 3-6 clearly depicts the trend where the increase in RSS and T-PN pairs improves the accuracy of AP localization. The rule that "AP Localization Error" and "Accuracy Estimation Error" decrease if the number of trajectories increases as discussed for Table 3-4 is validated in Table 3-6 as well. Note that both "AP Localization Error" and "Accuracy Estimation Error" are calculated in the 2D space.

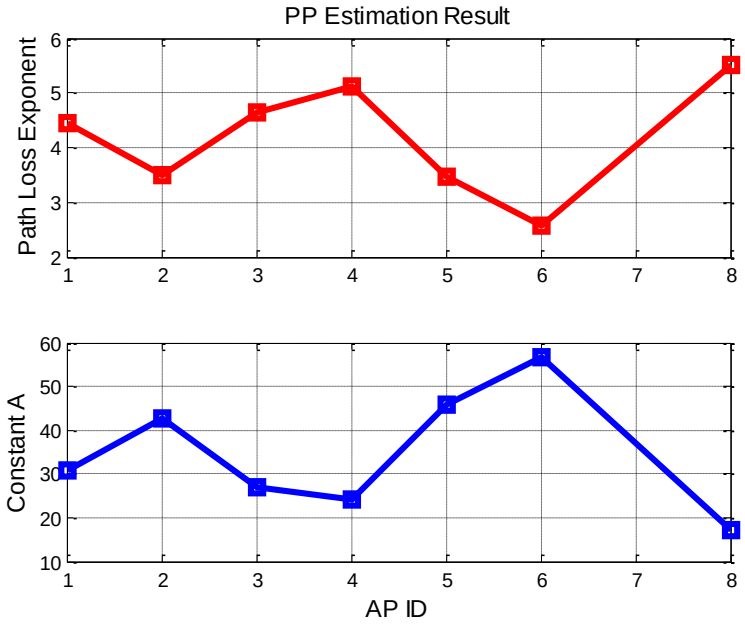
Table 3-6 AP localization results using LSQ1 in building E

Trajectory Number	**Number of Estimated APs**	**AP Localization Error**		**Accuracy Estimation Error**	
		MEAN (m)	RMS (m)	MEAN (m)	RMS (m)
1	5	9.77	10.62	4.54	6.49
2	7	6.39	7.52	3.94	5.32
3	7	5.06	5.96	3.79	5.04
4	7	4.91	6.05	3.62	5.12
5	7	5.22	5.94	4.32	5.20
6	7	5.05	5.75	4.15	5.09

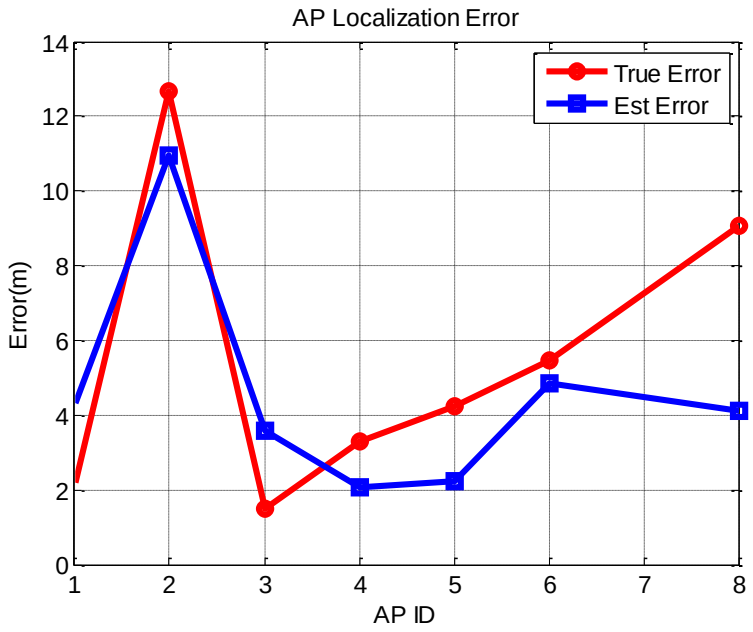
The evaluation of AP localization using LSQ2 was conducted in building E, as shown in Figure 3-10. The results in Figure 3-10 (a), (b), and (c) were estimated by using all six trajectories which are the same as Figure 3-9 (a). Figure 3-10 (a) and Figure 3-10 (b) illustrate the efficiency of AP localizations and PPs estimation by using LSQ2. In Figure 3-10 (c), estimated AP localization error is close to the true value.


AP Localization Result
EST
TRUE
E2
E6
T4
E4
E3
E5
T5
T1
T2
T6
T3
E8
E1
T8
North(m)
East(m)
60
50
40
30
20
10
0
-10
0
20
40
60
80

(a)


PP Estimation Result
Path Loss Exponent
Constant A
AP ID
1
2
3
4
5
6
7
8
10
20
30
40
50
60

(b)



(c)

Figure 3-10 Results of AP localization and PPs estimation in building E using LSQ2: (a) the result of AP localization, (b) the result of PPs estimation, and (c) estimated and true 2D errors of AP localization.

Table 3-7 also clearly depicts the trend where the increase in RSS and T-PN pairs improves the accuracy of AP localization. The rule that "AP Localization Error" and "Accuracy Estimation Error" decrease if the number of trajectories increases as discussed for Table 3-6 is validated in Table 3-7 as well. Trajectory 5 in which "AP Localization Error" increased is not the case since the observation error of this trajectory is large. Note that both "AP Localization Error" and "Accuracy Estimation Error" are calculated in 2D space.

Table 3-7 AP localization results using LSQ2 in building E

Trajectory Number	Number of Estimated APs	AP Localization Error		Accuracy Estimation Error	
		MEAN (m)	RMS (m)	MEAN (m)	RMS (m)
1	4	12.72	14.69	8.68	8.89
2	6	10.40	11.28	5.12	6.02
3	7	6.60	7.77	3.49	4.08
4	7	5.87	7.37	3.14	3.85
5	7	6.37	7.63	2.39	3.35
6	7	5.48	6.63	2.09	2.44

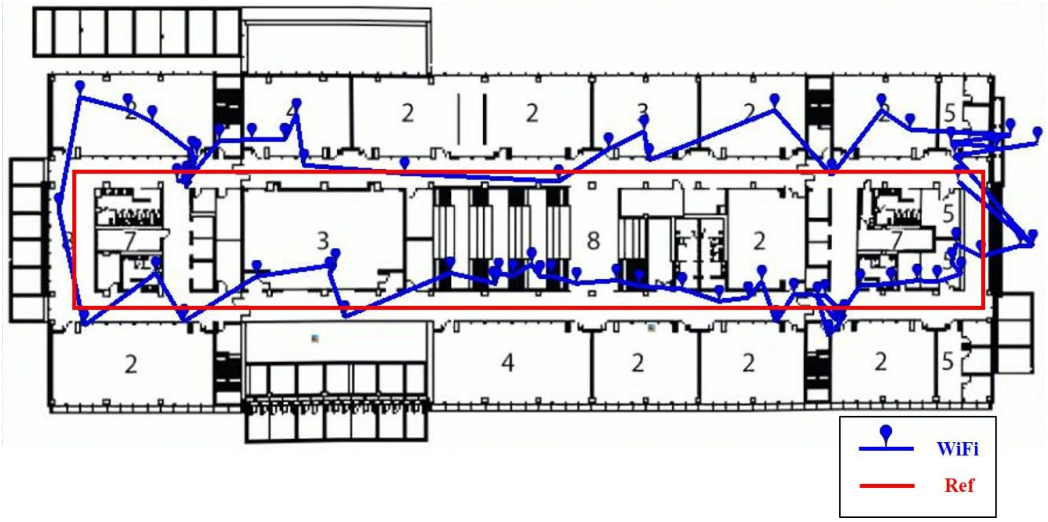
The compared results of the two proposed systems based on LSQ1 and LSQ2, respectively, are summarized in Table 3-8. These two methods use RSS and RSS+T-PN positions as observations respectively. As shown in Table 3-8, LSQ2 has a better performance than the case of LSQ1 in building A. However, it has a worse result than LSQ1 in building E. It is likely caused by the different accuracies of the provided position variances from T-PN in these two scenarios. The provided position variances in building A should be more accurate than the case of building E. The results illustrate that the performance of LSQ2 depends on the output position accuracy of T-PN. Considering that many MEMS-based navigation solutions cannot output the accuracy along with the position solution, LSQ1 which only adopts RSS values as observations is used in the rest of this thesis.

Table 3-8 Compared results of AP localization using the designed two LSQs.

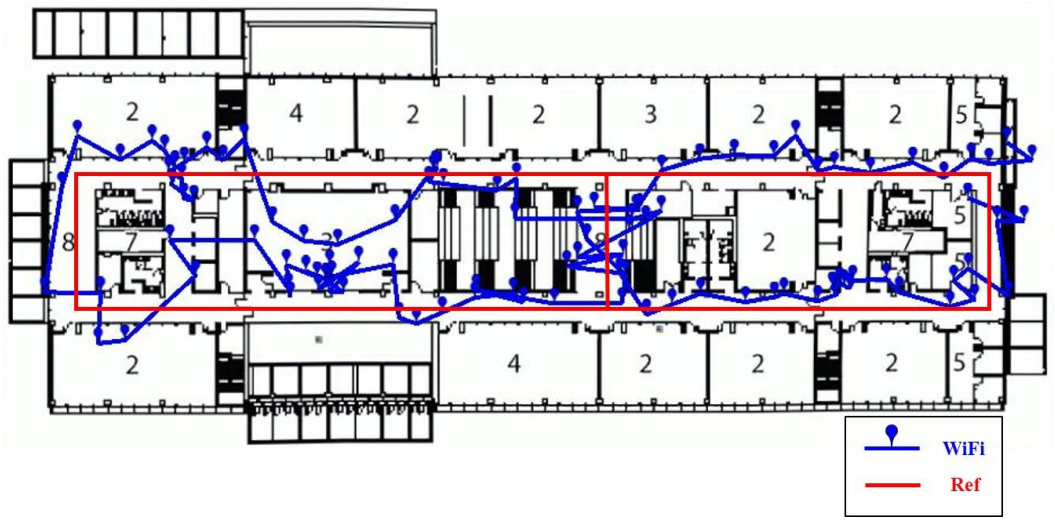
Observations for LSQ	Building A		Building E	
	MEAN (m)	RMS (m)	MEAN (m)	RMS (m)
RSS	5.51	6.14	5.05	5.75
RSS+T-PN Positions	4.78	4.90	5.48	6.63

3.6.2 Performance of WiFi Positioning Service

Three Samsung Galaxy S III smartphones were used in the two buildings to evaluate the performance of the proposed WiFi positioning service. Two Samsung Galaxy S III smartphones (Unit1 and Unit2) were used in building E, whereas two smartphones (Unit1 and Unit3) were used in building A. Note that the WiFi database, including AP locations, PPs, and the accuracy of AP locations, was automatically built using Unit1 in Subsection 3.6.1.2. As shown in Figure 3-11, WiFi positioning solutions of the two trajectories are close to the reference, which demonstrate the performance of the proposed WiFi positioning service.



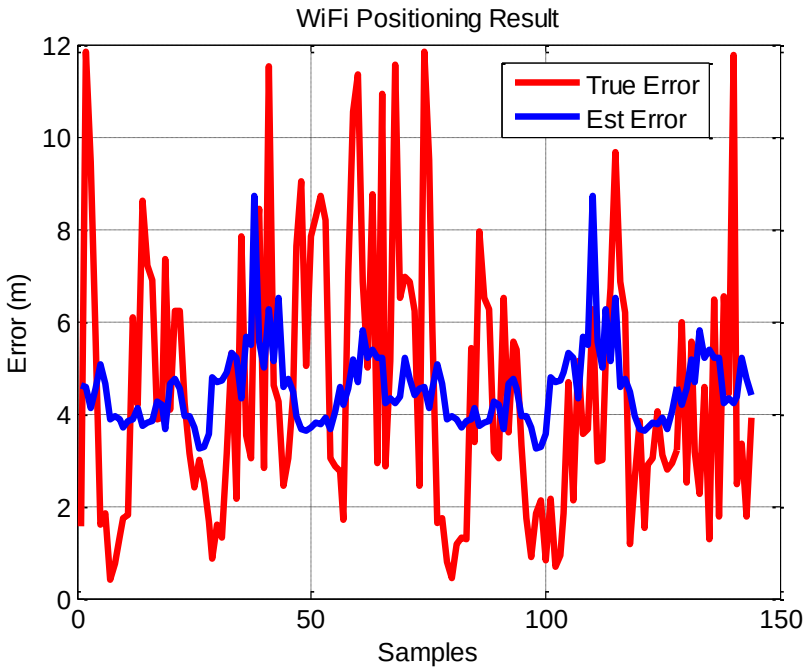
(a)



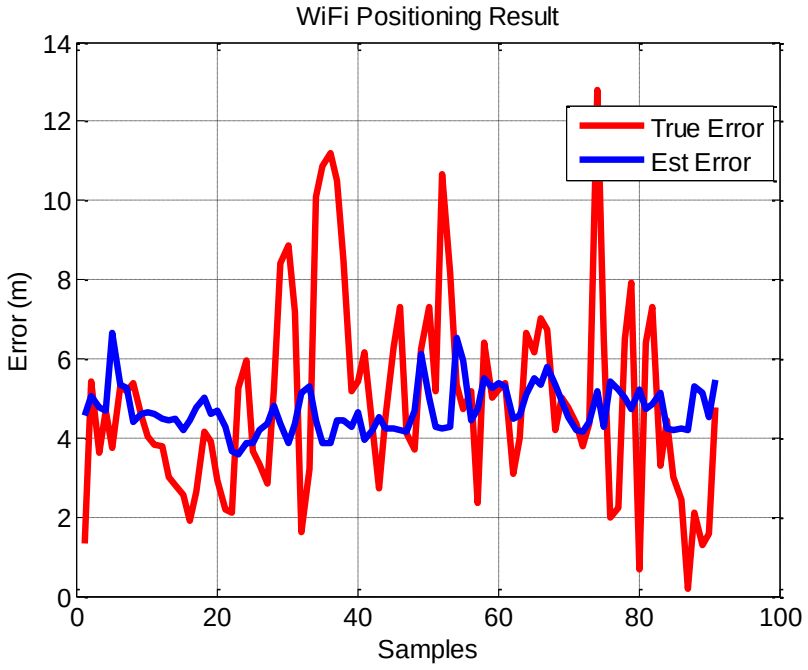
(b)

Figure 3-11 Result of WiFi positioning service in building E using Unit 1: (a) Trajectory I and (b) Trajectory II.

Figure 3-12 illustrates WiFi positioning errors in these two trajectories. The red lines represent true positioning errors, derived by comparing the proposed WiFi solution with the reference, while the blue lines are estimated positioning errors from the LSQ results. Note that reference trajectory is provided by using several pre-set markers' positions from the floor plan of the building. The results show that the estimated positioning error can be used as an approximate indicator for WiFi positioning accuracy. The number of APs used in estimating the user's position is shown in Figure 3-13. The AP number varies from 15 to 40 in both these two trajectories. Figure 3-12 and Figure 3-13 show that WiFi positioning error is related to the number of available APs for positioning. WiFi positioning accuracy is improved with the increase in the number of visible APs.

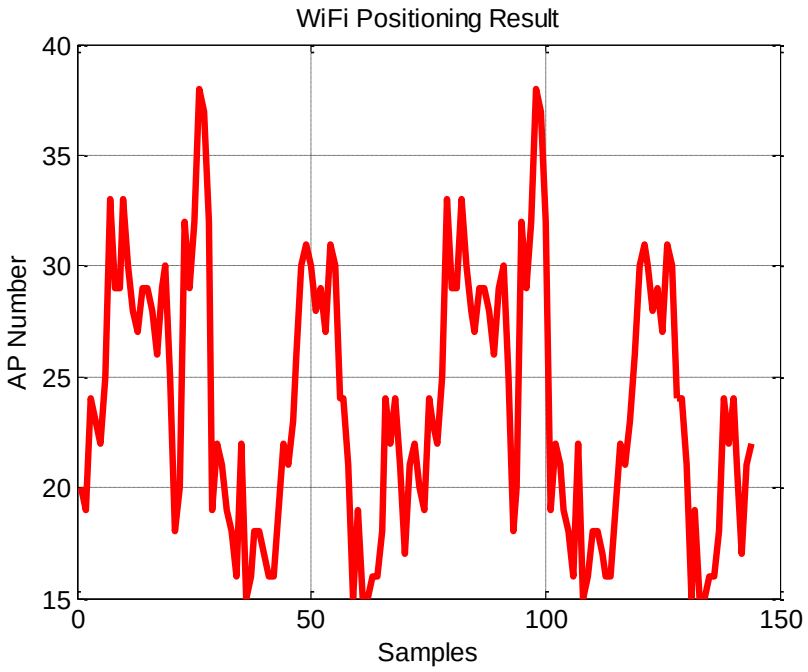


(a)

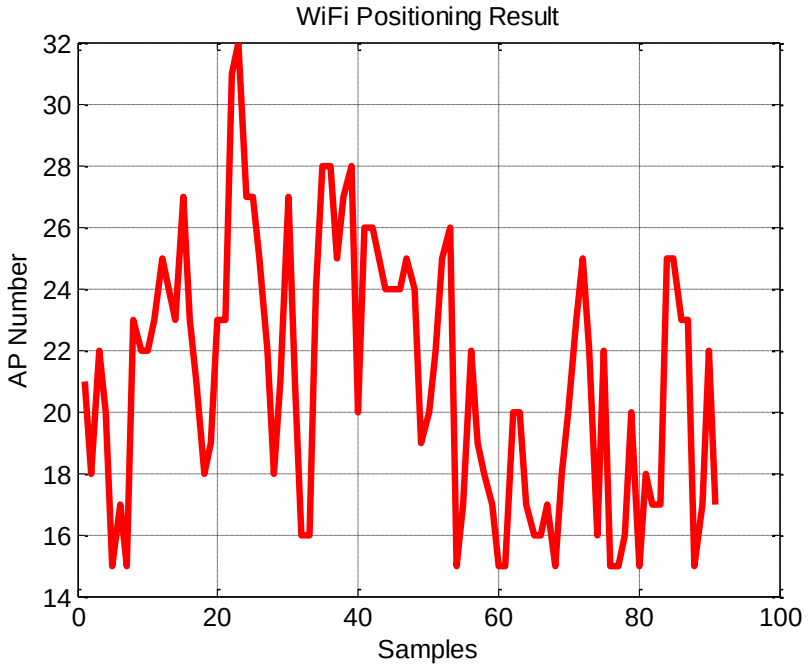


(b)

Figure 3-12 WiFi positioning error in building E using Unit1: (a) Trajectory I and (b) Trajectory II.



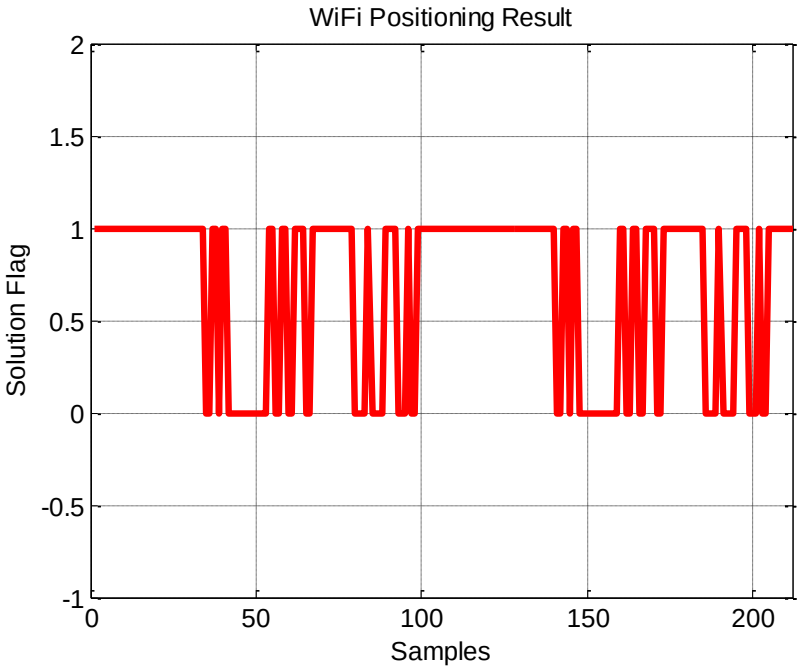
(a)



(b)

Figure 3-13 Observed AP number for WiFi positioning in building E using Unit1: (a) Trajectory I and (b) Trajectory II.

The availability of the WiFi positioning solution in building E using Samsung Galaxy S III Unit 1 is shown in Figure 3-14. In this figure, the solution flag equals 1 if WiFi solution is available, and is 0 when unavailable. Based on the criteria for convergence, the solution is only available when it is reliable and, hence, Figure 3-14 can also be used to determine whether the WiFi solution is reliable or not. The main reason for the unavailability of some WiFi solutions is that they cannot pass the verification of the criteria used for evaluating the performance of WiFi positioning in Section 3.5. The aim of the algorithm is to provide a reliable and accurate WiFi positioning solution which results in some unavailable periods. Other techniques such as inertial navigation can fill in the gaps when the solution is unavailable and, thus, a hybrid positioning system should be a part of an always-available type of positioning solution.

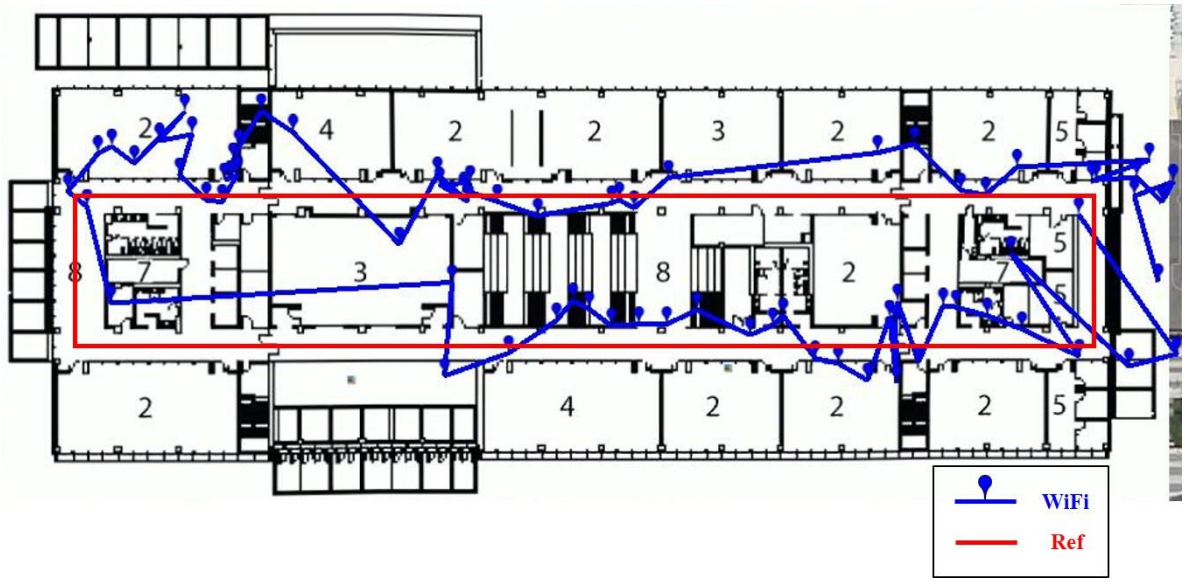


(a)

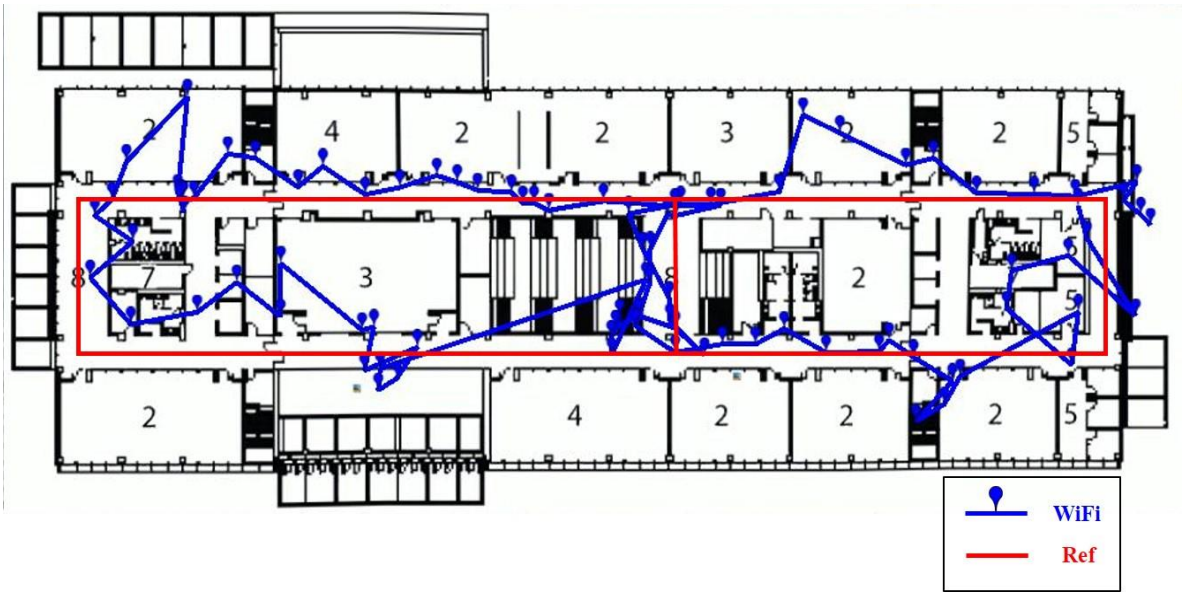
(b)

Figure 3-14 Availability of WiFi positioning in building E using Unit 1: (a) Trajectory I and (b) Trajectory II.

In Figure 3-15, Unit 2 provides the results of the WiFi positioning service while the database was built using Unit 1. The experiment is used to evaluate whether the WiFi database developed by a specific device can be used for other devices. In Figure 3-15, two trajectories in building E show that the WiFi positioning solutions are also close to the references, even when using a different device for positioning.



(a)



(b)

Figure 3-15 Result of WiFi positioning service in building E using Unit 2: (a) Trajectory I and (b) Trajectory II.

The performance of WiFi positioning in building E is summarized in Table 3-9. Note that both "WiFi Positioning Error" and "Accuracy Estimation Error" are calculated in 2D space. The positioning performance of Unit 2 is slightly worse than Unit 1 which may be due to the hardware

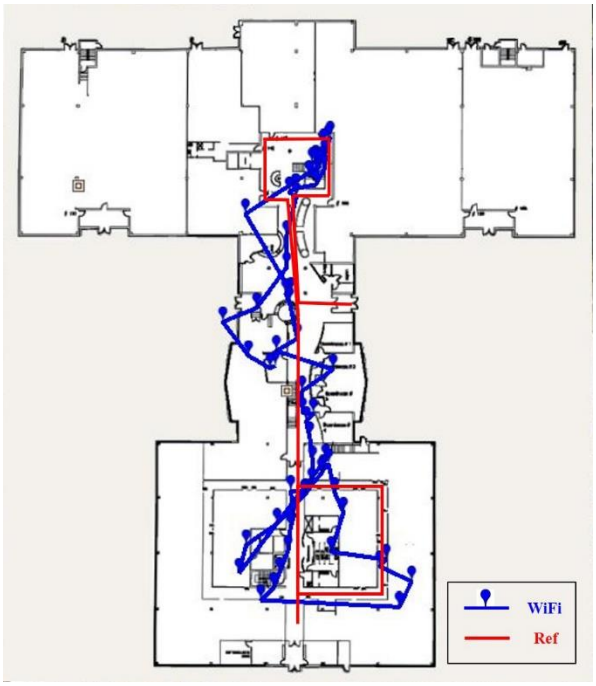
difference of the devices. In Table 3-9, average true positioning errors, which are the differences between estimated positions and the references, in different trajectories through the use of different devices are all less than 5.7 meters. "Accuracy Estimation Error" in Table 3-9 equals the difference between the estimated and true WiFi positioning errors, which is used to determine whether the estimated WiFi positioning error is an efficient indicator for the accuracy of WiFi positioning. Values of "Accuracy Estimation Error" in different trajectories are all less than 2.9 meters as shown in Table 3-9. Therefore, the estimated positioning accuracy is representative of the positioning performance of the proposed system.

Table 3-9 WiFi positioning result in building E

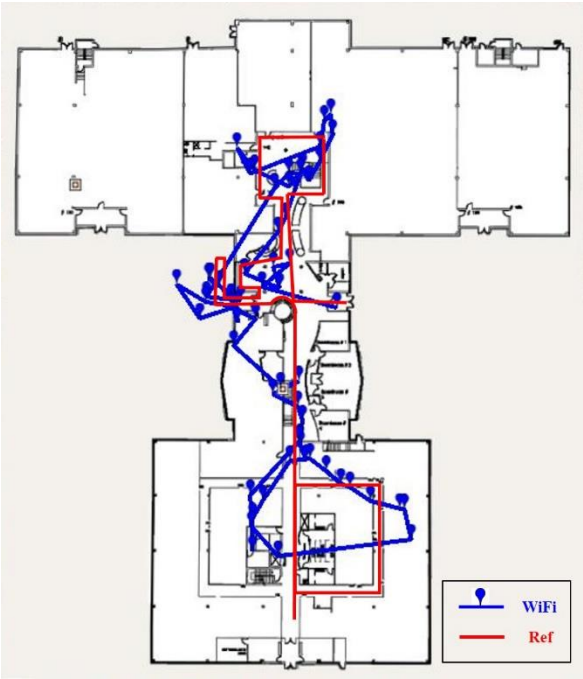
Device Unit	**Trajectory**	**Average AP number**	**Solution Availability**	**WiFi Positioning Error**		**Accuracy Estimation Error**	
				MEAN (m)	RMS (m)	MEAN (m)	RMS (m)
1	I	24	67.9%	4.49	5.32	2.32	2.84
1	II	21	83.4%	4.92	5.50	1.89	2.58
2	I	22	67.2%	5.66	6.66	2.60	3.46
2	II	22	82.6%	5.42	6.69	2.87	3.99

To evaluate the performance of the proposed algorithms in different environments, experiments were carried out in building A. WiFi positioning performance in building A is depicted in Figure 3-16 and Figure 3-17. The results in building A are summarized in Table 3-10. Note that both "WiFi Positioning Error" and "Accuracy Estimation Error" are calculated in 2D space. Similar to

Table 3-9, Unit 3 performs worse than Unit 1 which was used to build the database. In Table 3-10, average positioning errors in different trajectories through the use of different devices are less than 6.5 meters. The performance in building A is slightly worse than building E, which may be related to how building A had fewer APs. “Accuracy Estimation Error” stayed less than 3.1 meters as shown in Table 3-10.

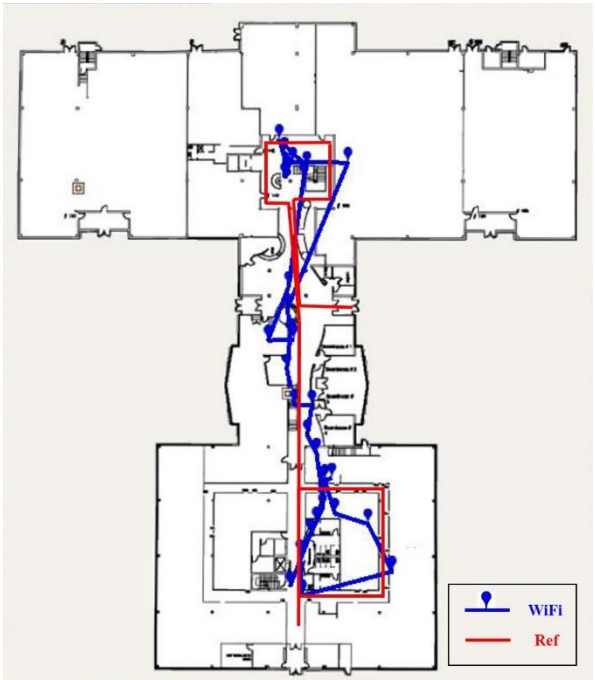


(a)

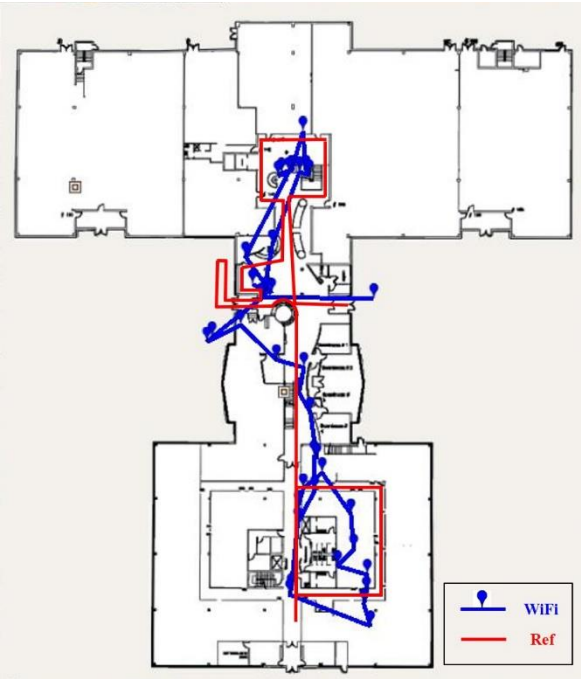


(b)

Figure 3-16 Result of WiFi positioning service in building A using Unit1: (a) Trajectory I and (b) Trajectory II.



(a)



(b)

Figure 3-17 Result of WiFi positioning service in building A using Unit 3: (a) Trajectory I and (b) Trajectory II.

Table 3-10 WiFi positioning result in building A

Device Unit	Trajectory	Average AP number	Solution Availability	WiFi Positioning Error		Accuracy Estimation Error	
				MEAN (m)	RMS (m)	MEAN (m)	RMS (m)
1	I	19	72.0%	5.73	7.39	3.02	4.72
1	II	19	56.5%	5.34	5.98	2.24	2.78
3	I	19	55.4%	6.43	7.22	2.93	3.53
3	II	18	52.0%	5.45	6.02	2.33	2.62

3.7 Summary

In this chapter, available WiFi positioning systems were discussed along with their shortcomings and then the proposed autonomous WiFi positioning system based on handheld devices was introduced. The main contribution of the work is to provide an automatic trilateration-based WiFi positioning solution, with virtually no cost to build and to maintain a WiFi database. This chapter mainly discussed two parts of the proposed automatic WPS: background survey service and WiFi positioning service.

In the first part, a background survey service based on crowdsourcing for automatic AP localization and PPs estimation through the use of the inertial navigation solution from the T-PN was introduced. When the requirements for estimating propagation parameters and AP locations were satisfied, the estimation results were recorded in the database for future positioning use. The method is user-friendly, easy to implement, and robust in changing indoor environments since the database can be continuously updated using crowdsourcing without any survey costs. The performance of the algorithms is evaluated by both simulations and experiments. Results prove the efficiency for building and maintaining the database by using the background survey service. Experimental results of two real-world scenarios show that the average estimation errors of AP locations are less than 6.0 meters. As shown in the results, "AP Localization Error" and "Accuracy Estimation Error" decrease with the increase of trajectories.

In the second part, WiFi positioning service through the use of the automatically generated database was discussed in details. The positioning algorithms included two parts: LSQ estimation for user positioning and the result optimization. Different smartphones were used to evaluate the performance of WiFi positioning in two scenarios. The results showed that average positioning

errors in different situations were all less than 6.5 meters. It was observed that the hardware difference between the devices used for database generation and user positioning could cause slight changes in positioning performance. Also, WiFi positioning performance is enhanced if more APs are available for user positioning.

Chapter Four: **Automatic WPS Based on Fingerprinting**

Fingerprinting-based WiFi positioning generally contains two phases: pre-survey and real-time positioning. In the pre-survey phase, a group of fingerprints including RSS values from available APs and corresponding position information are collected and stored in the radio map database. In the real-time positioning phase, the user's position is estimated based on the comparison between the observed RSS values and the fingerprints in the pre-built radio map database, which was obtained during the pre-survey phase.

Typically, the pre-survey phase is classified into two categories: expert surveyor model and crowdsourcing model. In the first category, trained professionals are employed to survey an area to obtain a robust and precise radio map database. Radio map databases need sporadic maintenance after they are built. Both the pre-survey and maintenance of radio map databases cost time and labor, especially for large areas. Recently, a crowdsourcing model was developed to reduce the cost for the building and maintaining of radio maps. In this model, regular users collect fingerprints during their daily routines to contribute to radio map databases in the pre-survey phase. However, this process introduces some new issues.

This research particularly discusses two major technical issues in the crowd-sourcing-based WiFi positioning system. The first issue is how to obtain accurate positioning information while automatically building the radio map database. As discussed in Chapter 3, T-PN, highly customizable software, can be used on many of the available smartphone operating systems (OS), including the Android OS. The results shown in Chapter 3 already illustrate the performance of the T-PN navigation solution. Therefore, the T-PN running on Android handheld devices is used as the position provider for the automatic fingerprints generation. Second, in the expert surveyor

model, professionals must spend a long time at each measurement point to build a precise radio map database. However, in the crowdsourcing model, fingerprints are generated automatically, whether the user is walking or static, as long as the software is running in the background of the devices. Thus, another issue for the crowdsourcing-based pre-survey is how to ensure the accuracy of the stored RSS measurements while the user is walking. In this case, only few fingerprints are collected at each measurement point. Approaches for measurement optimization, which were discussed in Chapter 3, are also adopted to improve the accuracy of measurements. In this chapter, several scenarios are used to evaluate the performance of the proposed WPS based on the crowdsourcing model. This chapter include three important research contribution:

- An autonomous background survey service based on the crowdsourcing method for radio map database generation is introduced using the T-PN software on handheld devices.
- A WiFi positioning service based on handheld devices using the automatically generated database is developed.
- In several field tests, performance of the automatic fingerprint-based WPS is compared with the automatic trilateration-based WPS, which was proposed in Chapter 3.

4.1 Background Survey Service

The flow chart of the background survey service is shown in Figure 4-1. After the optimization of measurements, RSS/Position-based fingerprints are recorded in the database, or used to update the database. The grid space is a significant parameter when designing the background survey service, affecting the performance of WiFi positioning. If the grid space is too small, the user stays in the same grid for a short time, which results in the insufficient collection of RSS samples. On the other

hand, a large grid space results in poor positioning accuracy. Considering the normal walking speed of a mobile user, an empirical grid space of three meters was selected in this research.

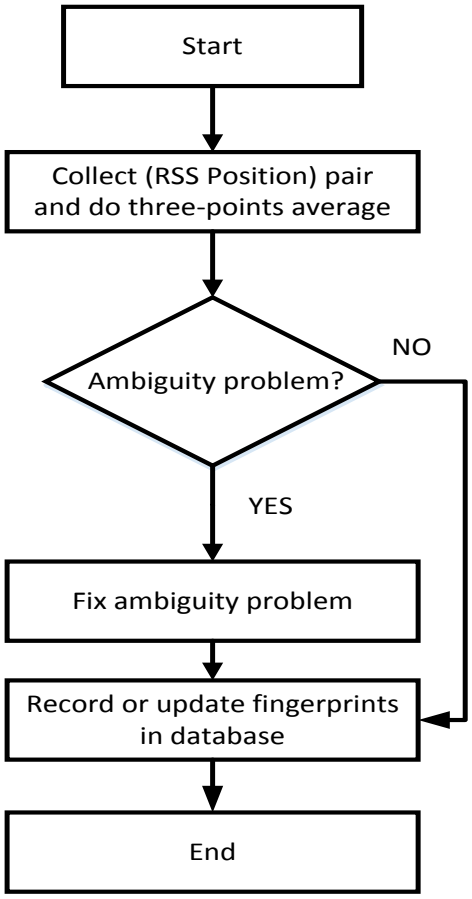


Figure 4-1 Flow chart of the background survey service of fingerprinting-based WPS.

The fingerprint tuple stored at each grid has the following form:

$$T = \{\text{LLH}, \sigma_h, \sigma_a, S\} \tag{4-1}$$

where LLH represents the geodetic coordinates (latitude, longitude, and height) of the grid, σ_h and σ_a represent the horizontal and vertical positioning accuracy, and S is the WiFi RSS set

received from the observable APs. LLH, σ_h, and σ_a are all provided by the T-PN solution. The WiFi RSS set, S, is stored as

$$S = \{(SSID_1, MAC_1, RSS_1), \ldots, (SSID_n, MAC_n, RSS_n)\} \quad (4\text{-}2)$$

where $SSID$ and MAC present the SSID name and MAC address of each AP, RSS represents the RSS vector after the three-point average, and n represents the number of APs.

In the proposed system, fingerprint tuples are automatically saved, and if needed can be uploaded to the database through the background survey service. The system requires no active participation by the user, which is a substantial improvement over the expert survey systems. Autonomous crowdsourcing is a significant part of the background survey service, which is used to efficiently ensure the creation and maintenance of the radio map database. RSS values and corresponding positioning solutions from the T-PN are automatically collected as measurement pairs by the background service during users' daily routines in the proposed system. An example for the radio map database generation from user's daily trajectories is shown in Figure 4-2. When more and more measurements are collected and updated to the radio map database, the database becomes increasingly accurate through autonomous crowdsourcing without additional operations.

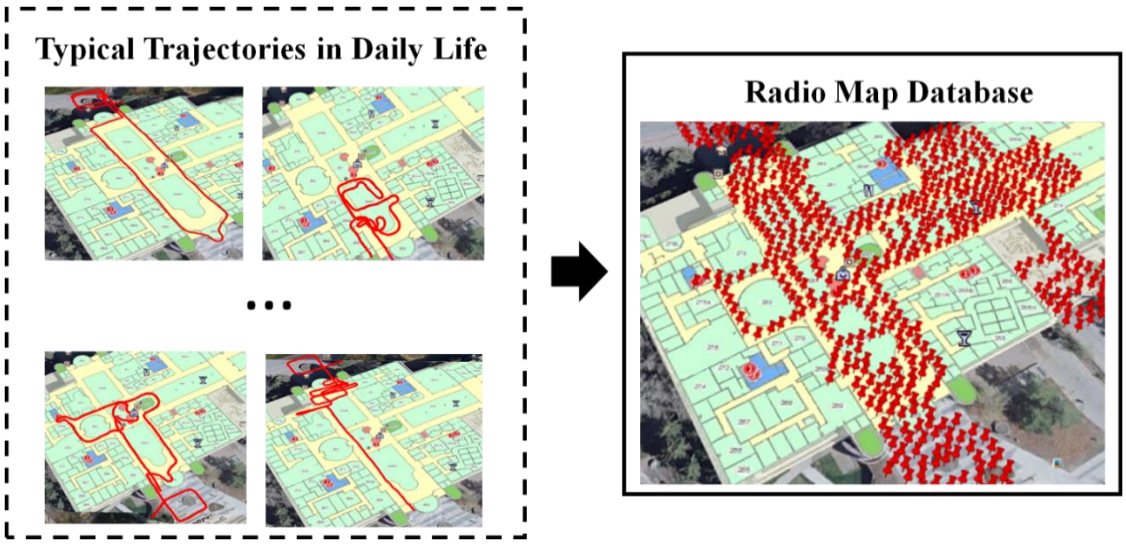


Figure 4-2 An example of radio map database generation from several trajectories.

4.2 WiFi Positioning Service

In the proposed system, a user's position is calculated by using the automatically generated radio map database. The flow chart of the WiFi positioning service is described in Figure 4-3. The position of the mobile user is adjusted to the average position of the closest K fingerprints in the radio map database with the minimum Euler distances, as shown in Figure 4-3. Because the system does not guarantee that the radio map database contains all the fingerprints in the building, the smallest Euler distance should be less than the threshold, "dis_th", to ensure the WiFi positioning result is reliable. When the WiFi position is calculated, it can be used to aid the MEMS solution and improve the absolute accuracy of the mobile user's position estimation, which will be discussed in Chapter 5.

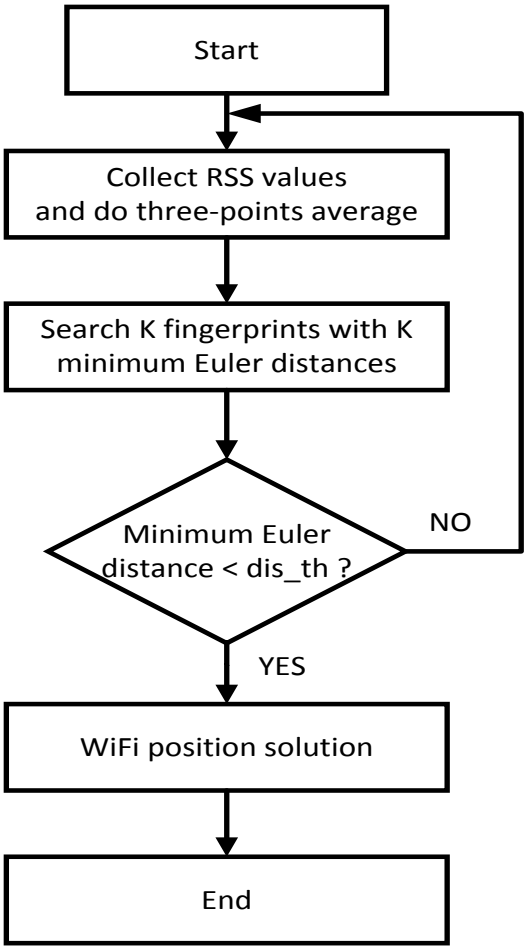


Figure 4-3 Flow chart of the WiFi positioning service in the fingerprinting-based WPS.

4.3 Comparison of Fingerprinting-Based WPS and Trilateration-Based WPS

Two proposed automatic systems, fingerprinting-based WPS and trilateration-based WPS, are compared in this subsection. There are several differences between these two systems. First, the accuracy of the trilateration-based WPS relies on the propagation model, which is affected by several factors such as multipath and fading characteristics. Fingerprinting-based WPS calculates the user's position based on the fingerprint matching algorithms, which is less affected by multipath and fading characteristics. Second, a radio map database typically has more data than a database for trilateration in the same area, which means that a radio map database uses more stored

memory. Third, the pre-survey and maintenance of a radio map database usually consume more time and labor than a database for trilateration. Lastly, the trilateration-based automatic WPS usually has a poorer positioning performance compared to the fingerprinting-based automatic WPS.

4.4 Test Results and Performance Analysis

4.4.1 Automatic WPS Based on Fingerprinting

To evaluate the performance of the proposed system in field tests, we implemented the background survey service and WiFi positioning service on handheld devices with Android OS, such as: Google Nexus 7 and Samsung Galaxy S III. These Android devices are equipped with accelerometers, gyroscopes, magnetometers and a WiFi receiver. For the performance evaluation, we selected two test sites which had different environments. As shown in Figure 4-4 (a), the first experimental site was a large open area in the west section (about 90m × 70m) of building M. Another site was in building E (about 120m × 40m), which has some corridors as shown in Figure 4-4 (b). The performance T-PN results is discussed in Section 3.2 (Figure 3-2).

(a)

(b)

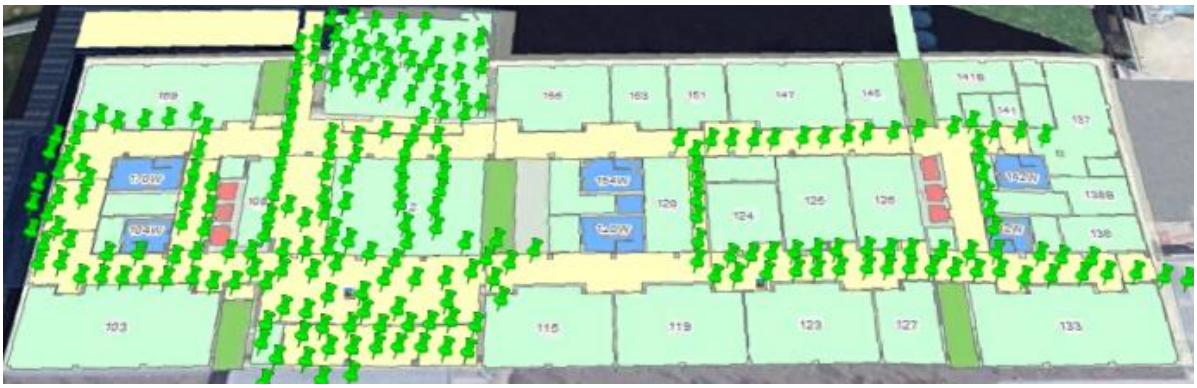
Figure 4-4 Test scenarios: (a) building M and (b) building E.

Figure 4-5 and Figure 4-6 show the automatically generated radio map databases in two scenarios (building M and building E) by using different numbers of trajectories from a typical pedestrian everyday usage of mobile devices. Figure 4-5 (a), (b), and (c), and Figure 4-6 (a), (b), and (c) show the radio map databases in building M and building E which are automatically generated from 6, 12, and 16 trajectories. We expect that the automatically generated databases will become accurate

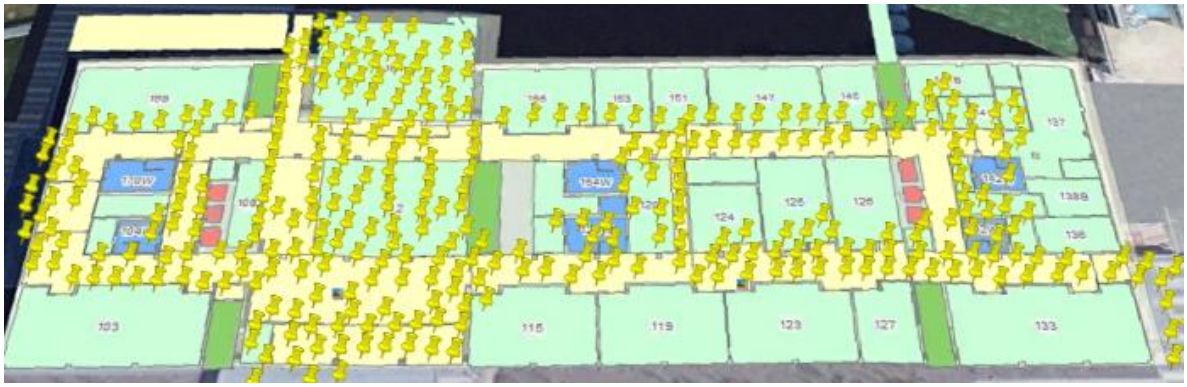
and robust with the increase in used trajectories. However, it is difficult to directly evaluate the accuracy of radio map databases. Therefore, the radio map databases from different numbers of trajectories are used to estimate WiFi positioning results, which will indirectly illustrate the performance of the radio map databases. WiFi positioning results from various databases will be discussed later.

(a) **(b)** **(c)**

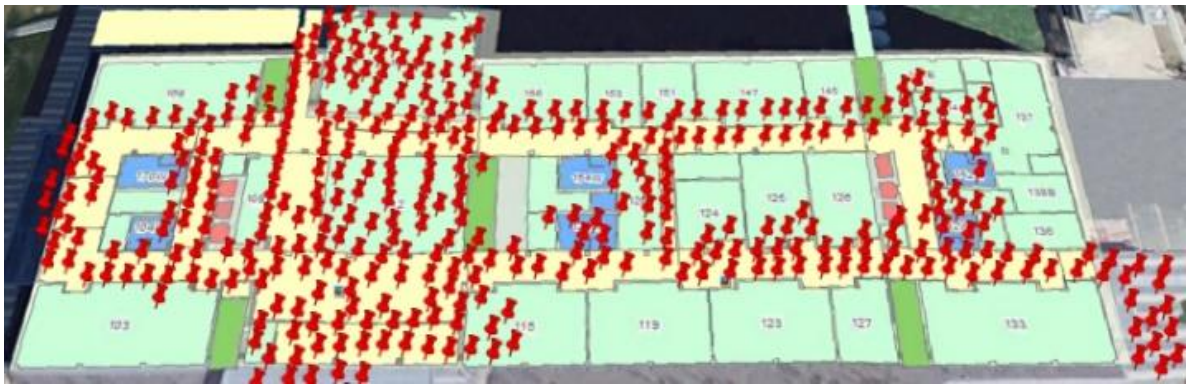
Figure 4-5 Radio map databases by using different numbers of trajectories (Scenario I: building M): (a) 6 trajectories, (b) 12 trajectories, and (c) 16 trajectories.



(a)



(b)

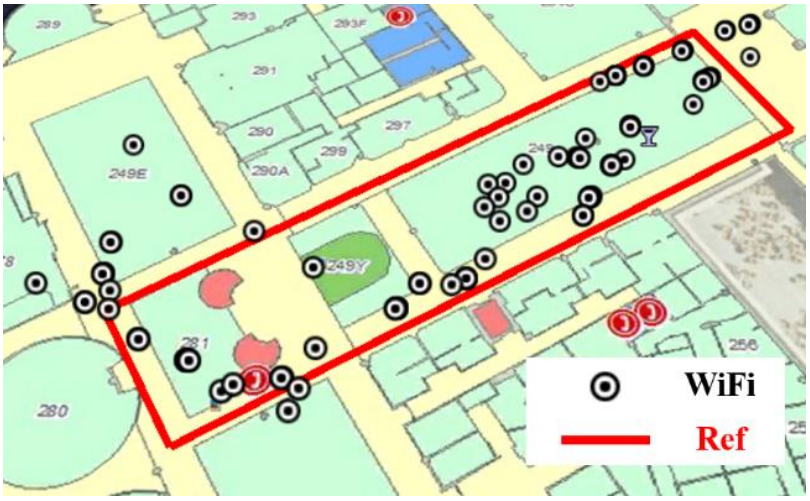


(c)

Figure 4-6 Radio map database by using different numbers of trajectories (Scenario II: building E): (a) 6 trajectories, (b) 12 trajectories, and (c) 16 trajectories.

WiFi positioning results, based on the automatically generated radio map databases in the two test sites, are shown in Figure 4-7, Figure 4-8, and Figure 4-9. Table 4-1 also summaries the WiFi positioning results. Through the figures and table, it is clear that the WiFi positioning solution improves when more trajectories are used for generating the radio map database. It is shown that the maximum RMS of position errors in these three tests is less than 5.7 meters when using most available trajectories (16 or 20) for building the radio map database. The results illustrate that the proposed system can achieve an accuracy of 5.7 meters (RMS error), without professional surveys. When comparing our results with WiFi SLAM algorithms (Bruno and Robertson 2011; Faragher

and Harle 2013; Ferris et al. 2007; Huang et al. 2011), it can be seen that the presented results are slightly less accurate. However, the proposed algorithm requires less computation loads and less survey costs for the generation of WiFi radio map databases, and this is the main advantage of the proposed system.

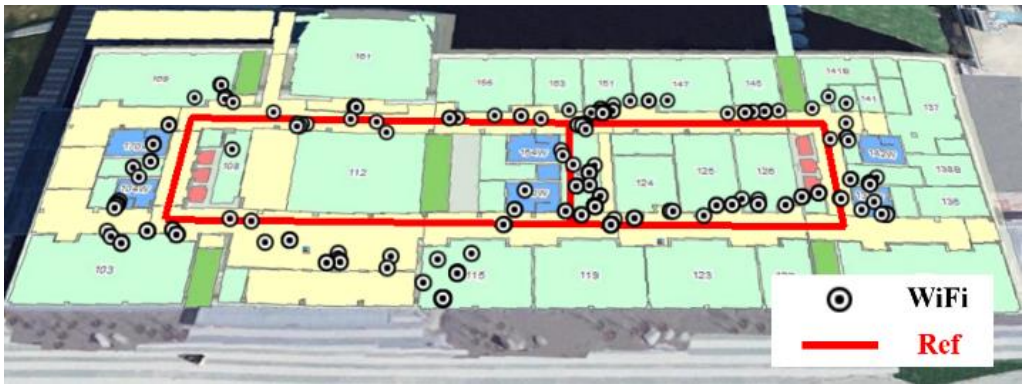


(a)

(b)

Figure 4-7 WiFi positioning results of Trajectory I (rectangle) by using different radio map databases: (a) the radio map database built from 6 trajectories and (b) the radio map database built from 16 trajectories.

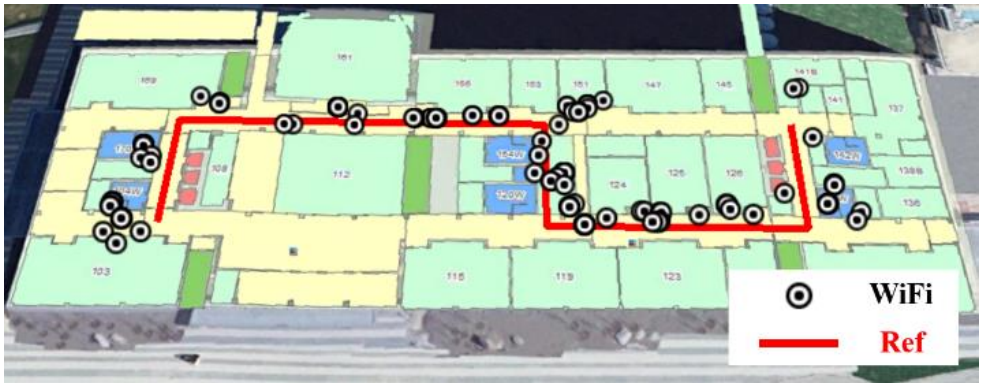
(a)



(b)

Figure 4-8 WiFi positioning results of Trajectory II (figure-eight) by using different radio map databases: (a) the radio map database built from 6 trajectories and (b) the radio map database built from 16 trajectories.

(a)



(b)

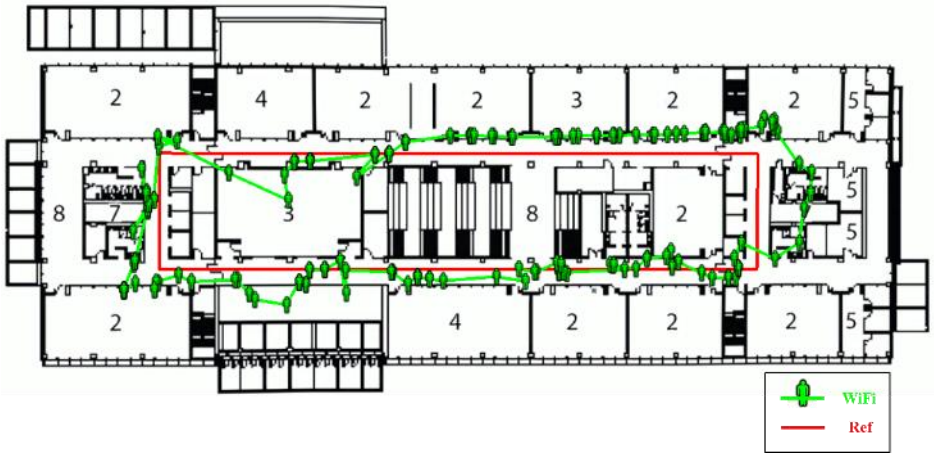
Figure 4-9 WiFi positioning results of Trajectory III (figure-s) by using different radio map databases: (a) the radio map database built from 6 trajectories and (b) the radio map database built from 16 trajectories.

Table 4-1 WiFi positioning results based on various radio map databases

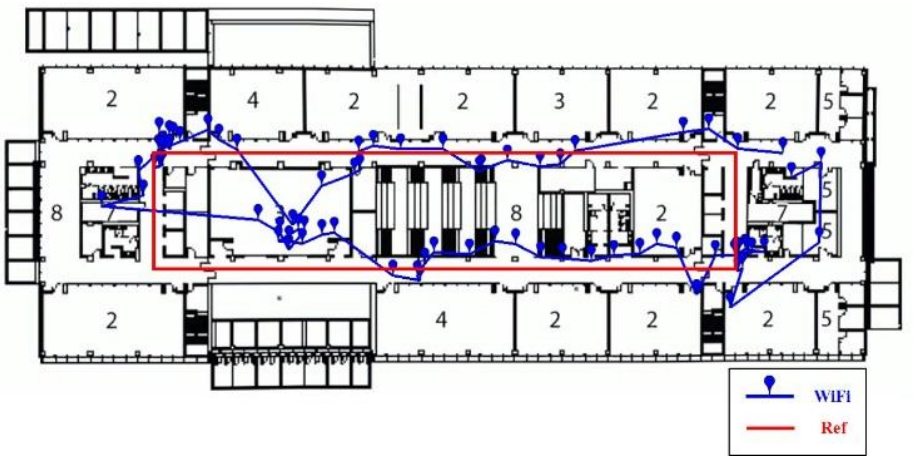
Trajectory	Number of Trajectories for Radio Map Database	Mean Error (m)	RMS Error (m)
I	6	9.5	10.7
	12	6.4	7.4
	16	5.1	5.7
II	6	6.0	7.2
	12	4.2	4.8
	20	4.6	5.3
III	6	7.6	8.9
	12	5.5	6.3
	20	4.6	5.5

4.4.2 Performance Comparison of Two automatic WPSs

Experiments were conducted in building E to compare the performance of two proposed automatic WPSs. In the fingerprinting-based background survey service, twenty trajectories were collected to build the radio map database to cover most of the area of building E. In the trilateration-based background survey service, six trajectories are collected to achieve a 5.0 meters average error of AP locations in the database of building E. Note that the trajectories used to build the radio map database are collected at the same day as the trajectories for trilateration. Five trajectories are selected to compare the positioning performance of these two systems. In each trajectory, the collected RSS values are used to estimate the user's positions by using trilateration and fingerprinting, respectively. Then, these two solutions are carefully compared. As an example, Figure 4-10 shows the performance of these two systems in trajectory I. The summary of positioning results for all five trajectories is shown in Table 4-2. It appears that the fingerprinting-based system performs better than the trilateration-based system in all five trajectories in Table 4-2. The average positioning errors of the fingerprinting-based system are about 1.8 meters smaller than the trilateration-based system.



(a)



(b)

Figure 4-10 Positioning results of two WPSs in building E: (a) fingerprinting-based WPS and (b) trilateration-based WPS.

Table 4-2 Positioning results of two automatic WPSs in building E

Positioning Error		**Fingerprinting**	**Trilateration**
Mean (m)	Trajectory I	3.24	5.78
	Trajectory II	4.55	5.44
	Trajectory III	3.03	4.49
	Trajectory IV	3.21	4.95
	Trajectory V	3.22	5.41
RMS (m)	Trajectory I	3.99	6.46
	Trajectory II	4.92	6.34
	Trajectory III	3.70	5.32
	Trajectory IV	3.79	5.92
	Trajectory V	3.80	6.48

Table 4-3 summarizes the comparison results of two systems, which includes average positioning error, memory of database, and total data time. As shown in Table 4-3, the average positioning errors of fingerprinting-based system are about 1.8 meters less than trilateration. Memory cost for the radio map database is about 7 times of the trilateration database. And, total data used to build the radio map is about 4 times of the data for building the trilateration-based database. The relative ratio of numbers of trajectories used for the fingerprinting and trilateration is 10/3. Overall, fingerprinting-based WPS provides a more accurate positioning solution at the cost of more labor and memory for the database. Even the fingerprinting-based system is more accurate than the trilateration-based system, trilateration-based system is selected for the rest of the research for the following three reasons: (1) it requires less trajectories to build the database than fingerprinting in the crowdsourcing system; (2) the memory cost for trilateration-based database is much less than fingerprinting-based database; and (3) LC integration of trilateration-based WiFi positioning solution and MEMS sensors can be used to compare with the proposed TC integration of WiFi and MEMS sensors. Both of them are based on the WiFi propagation model.

Table 4-3 Compared results of two systems in building E

	Fingerprinting	**Trilateration**
Average Position Error (m)	3.45	5.21
Memory of Database (byte)	690188	10300
Total Data Time (min)	85.41	22.95

4.5 Summary

In this chapter, an automatic fingerprinting-based WPS was proposed on handheld devices, which included a background survey service and a WiFi positioning service. Normal handheld-device-users use the background survey service, based on autonomous crowdsourcing to build and maintain the radio map database. This is automatic, and the pre-survey process is virtually labor-free, which is also a primary contribution of this work. Another contribution of the work is to compare the performance of the automatic fingerprinting-based WPS with the trilateration-based WPS.

In the fingerprinting-based WPS, a method of automatically generating fingerprints was presented based on the proposed background service and the T-PN, which run on handheld devices. Fingerprints were collected to build the radio map database after selection in the survey service. This radio map database generation was carried out by a non-professional user, not a trained surveyor, which reduced the costs of labor and time. Next, the fingerprinting-based WiFi positioning was discussed, and its performance was evaluated through the field tests. The results showed that average WiFi positioning errors were about 5.1 meters, which demonstrated the efficiency of the automatic fingerprinting-based WPS.

The proposed automatic fingerprinting-based WPS was also compared with the automatic trilateration-based WPS, which was discussed in Chapter 3. Typically, it is easier to survey and maintain a database for trilateration rather than fingerprinting, which is also illustrated by the experiments. The results of field tests showed that the fingerprinting-based WPS had a better positioning performance than the trilateration-based system. Overall, both systems based on

handheld devices without special hardware, were automatic, practical, and improved positioning solutions indoors.

Chapter Five: **WiFi/MEMS Integration for Indoor Navigation**

A handheld indoor navigator is discussed in this chapter based on the integration of low-cost MEMS sensors and WiFi. To build an accurate and practical navigator, three approaches are proposed to enhance the navigation performance: 1) The use of a MEMS solution based on PDR/INS (Inertial Navigation System/Pedestrian Dead Reckoning) integration; 2) The use of motion constraints for the proposed MEMS solution, such as NHC (Non-holonomic Constraints), ZUPT (Zero Velocity Update), and ZARU (Zero Angular Rate Update); 3) The use of LC/TC (Loosely-coupled/Tightly-coupled) integration for MEMS sensors and WiFi. Note that when using LC integration, trilateration-based WiFi solution is selected to integrate with MEMS sensors according to the compared results of two proposed automatic WPSs presented in Chapter 4.

The first approach improves the indoor pedestrian navigation solution based on the proposed MEMS solution through the use of PDR/INS integration for handheld devices. PDR used in the handheld devices usually assumes that the handheld device is level (roll and pitch are zero degrees). However, this assumption is sometimes incorrect. In these cases, the PDR-based heading, calculated by the direct integration of the vertical gyroscope, is inaccurate. The heading estimation error will finally affect the positioning accuracy. In this chapter, a MEMS solution on handheld devices for indoor navigation, based on the use of PDR/INS integration is proposed. The proposed PDR/INS-integrated MEMS solution combines the advantages of both schemes. In this algorithm, step detection and step length estimation are kept the same as the traditional PDR algorithm. The estimated step length is used to calculate the forward speed, which works as the velocity update for the INS to limit the velocity error, and further limit the position error and attitude error. Therefore, the PDR/INS-integrated MEMS solution is superior to the INS solution. The heading

from the PDR/INS integration also performs better when compared with PDR because it considers the effect of the roll and pitch.

The second improvement is due to the use of motion constraints, such as NHC, ZUPT, and ZARU for the MEMS navigation solution. With these motion constraints, the indoor pedestrian navigator is expected to provide a better navigation solution. The third approach improves the performance of the handheld navigator through the use of LC and TC integrations of WiFi and MEMS sensors. Most research has focused on the integration of WiFi and body-mounted MEMS sensors (Chai et al. 2012; Evennou and Marx 2006; Frank et al. 2009), which are not as convenient as handheld devices for pedestrians. In this chapter, the LC and TC integrations of WiFi and MEMS sensors on handheld devices will be developed. In the LC integration, WiFi positions are used as the updates for the MEMS sensors. As per the discussion in Chapter 2, WiFi positions were mainly calculated through fingerprinting or trilateration (Gezici 2008). Fingerprinting usually provides a more accurate position solution, but at the cost of extensive work in the pre-survey phase. To reduce the survey work in the proposed system, trilateration is adopted in this research to provide a WiFi position solution for the LC integration. LSQ is usually used to adjust an optimal solution for the trilateration. Besides the WiFi position solution, an approximate position accuracy is also derived from the position covariance matrix in the LSQ, which works as an indicator to determine whether WiFi position is accurate enough for LC integration. The LC integration has one main drawback: no WiFi positions are provided as updates for MEMS sensors if the observed APs are less than three. This drawback limits the navigation performance of LC integration, especially in an environment with sparsely deployed APs. A TC integration is proposed to overcome this limitation, and improve the navigation performance. Different from the LC integration, which is

based on the MEMS navigation solution and WiFi positioning solution, the proposed TC integration integrates the raw data of MEMS sensors with WiFi-RSS-based distances/ranges. 15 states for MEMS sensors (3D position error, 3D velocity error, 3D attitude error, gyroscope drift, and accelerometer bias) and 1 state (RSS bias) for WiFi are used as the state vector in the EKF for the TC integration. The main benefit of this method is that the drift of MEMS sensors can be reduced by WiFi, even if only one or two APs are available. The introduction of the WiFi RSS bias in the TC integration also improves the navigation performance of the proposed system. On the other hand, TC integration has been used for the integration of inertial sensors with GPS, RFID and USBL (George and Sukkarieh 2005; Li et al. 2006b; Morgado et al. 2006; Ruiz et al. 2012; Wendel and Trommer 2004; Yi and Grejner-Brzezinska 2006).

To demonstrate the performance of the proposed algorithms, field tests are carried out in typical indoor environments. The navigation performances of PDR, INS, the PDR/INS-integrated MEMS solution, the LC integration solution, and the TC integration solution are also compared in this research.

Three main developments will be discussed in this chapter:

- In this research, we propose an innovative MEMS navigation solution based on the integration of INS and PDR for handheld devices. The field tests show that the performance of the proposed MEMS solution is better than the traditional INS and PDR.
- Two integrated methods for MEMS sensors and WiFi, LC integration and TC integration, are proposed to improve the accuracy of indoor pedestrian navigation.
- The navigation performances of PDR, INS, the PDR/INS-integrated MEMS solution, the LC integration solution, and the TC integration solution are compared in this research.

5.1 MEMS Solution Based on INS/PDR Integration and Motion Constraints

The block diagram of the proposed MEMS solution for indoor pedestrian navigation is shown in Figure 5-1. In this proposed MEMS solution, data from gyroscopes and accelerometers first pass to the INS mechanization. The accelerometer and gyroscope data are also used for step detection and static detection, respectively. If the step detection is successful and the static detection fails, the PDR step length is estimated in the module of "step length estimation", and is further used to derive the forward speed. NHC is also used to constrain the lateral and vertical speeds of the moving platform. The PDR-based forward speed and the NHC-based lateral and vertical speeds are combined to 3-axis pseudo-velocity to work as the velocity update for the INS to limit the velocity error. If the step detection fails and static detection is successful, ZUPT and ZARU apply zero velocity and unchanging heading as the velocity and heading updates for the INS to reduce navigation errors. EKF finally outputs the proposed sensor-based navigation solution. The state vector in the EKF components are shown as follows:

$$x_s = \begin{bmatrix} \delta r_{1\times3} & \delta v_{1\times3} & \varepsilon_{1\times3} & d_{1\times3} & b_{1\times3} \end{bmatrix}^T \tag{5-1}$$

where δr , δv , and ε represent errors of position, velocity and attitude. d and b represent gyroscope drift and accelerometer bias, which are estimated and fed back to the INS mechanization. The initial position can be set by two methods: (1) wireless position solution such as WiFi and Bluetooth; and (2) user manually pre-set value. Because the user is usually static at the beginning of navigation in indoor environments, the initial velocity is set to zero. The initial heading is set manually or by using the heading of magnetometers. The initial roll and pitch are estimated from accelerometer leveling (El-Sheimy 2006). The gyroscope drift and accelerometer

bias are modelled as first order Gaussian-Markov processes. For details about the reference frames used in this section, please refer to Chapter 2.

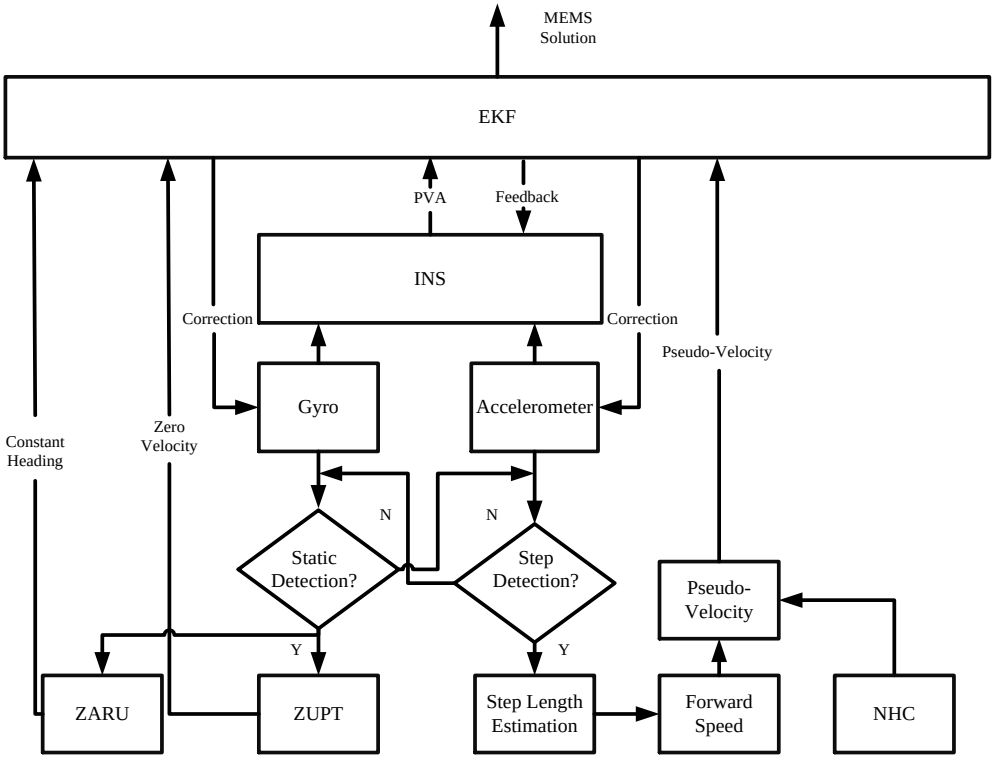


Figure 5-1 Block diagram of the proposed MEMS solution.

Angular rates and accelerations from the gyroscopes and accelerometers are used to detect the status of the pedestrian: moving or static. The status of the pedestrian is determined as “moving”, if the following two conditions are satisfied: 1) the standard deviation of the angular rate norms during a certain time is larger than the threshold; and 2) steps are detected by using the approach discussed in Chapter 2. On the other hand, the status of the pedestrian is determined as “static”, if the following two conditions are satisfied: 1) the standard deviation of the angular rate norms during a certain time is less than the threshold; and 2) no steps are detected.

For the "moving" case, the step length estimation has been discussed in Chapter 2. To use the step length to provide information about the forward speed, we assume that a pedestrian's moving speed is constant for a short time. This assumption is correct for most moving cases of pedestrians. The forward speed can be derived from the step length as follows:

$$v_{forward} = SL / T_{step} \tag{5-2}$$

where SL represents the step length, and T_{step} represents the duration of a step. NHC is also used to constrain the lateral and vertical speeds of the pedestrian. Combining the NHC and PDR-based forward speed, the pseudo-velocity vector in the body frame is given by

$$v^b_{pseudo} = \begin{bmatrix} SL / T_{step} & 0 & 0 \end{bmatrix}^T \tag{5-3}$$

The pseudo-velocity-vector is used for the INS velocity update to improve the MEMS navigation performance. The misclosure of the velocity in the body frame is given by

$$\delta z = v^b_{INS} - v^b_{pseudo} \tag{5-4}$$

where $v^b_{INS} = \left(C^n_b\right)^T \cdot v^n_{INS}$ represents the INS-based velocity in the body frame; C^n_b represents the transformation matrix from the body frame to navigation frame; and v^n_{INS} represents INS-based velocity in the navigation frame. Finally, the observation model for the pseudo-velocity-vector update is expressed in

$$\delta v^b = H_{v^b} \delta x + v_{v^b} \tag{5-5}$$

where v_{v^b} represents the measurement noise. H_{v^b} represents the corresponding design matrix:

$$H_{v^b} = \begin{bmatrix} 0_{3\times3} & \left(C_b^n\right)^T & -\left(C_b^n\right)^T V^n & 0_{3\times6} \end{bmatrix} \tag{5-6}$$

where V^n is the skew-symmetric matrix of v^n.

If "static" is detected, ZUPT and ZARU are used as the updates for the INS to limit the navigation error. The ZUPT-based zero velocity vector in the body frame is given by

$$v_{ZUPT}^b = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \tag{5-7}$$

Similar to the pseudo-velocity vector, the ZUPT-based zero velocity vector is used as the velocity update for the INS. If the pedestrian is detected as "static", the pedestrian heading is unchanging based on ZARU. Therefore, misclosure of the heading for the INS heading update is given by

$$\delta z = \psi_{INS} - \psi_{pre-stored} \tag{5-8}$$

where ψ_{INS} is the INS-based heading; and $\psi_{pre-stored}$ is the pre-stored heading of the last epoch before the "static" is detected. Finally, the observation model for the heading update is expressed in

$$\delta\psi = H_\psi \delta x + v_\psi \tag{5-9}$$

where v_ψ represents the measurement noise; and H_ψ represents the corresponding design matrix:

$$H_\psi = \begin{bmatrix} 0_{1\times6} & \partial\psi/\partial\varepsilon_N & \partial\psi/\partial\varepsilon_E & \partial\psi/\partial\varepsilon_D & 0_{1\times7} \end{bmatrix} \tag{5-10}$$

For details of $\partial\psi / \partial\varepsilon_N$, $\partial\psi / \partial\varepsilon_E$, and $\partial\psi / \partial\varepsilon_D$, please refer to (Syed et al. 2008).

5.2 LC Integration of WiFi and MEMS Sensors for Indoor Navigation

An overview of the proposed LC integration of WiFi and MEMS sensors on handheld devices such as smartphones is illustrated in Figure 5-2. The module outside the dashed boxes is the proposed MEMS solution based on the integration of PDR and INS. The module inside the dashed boxes is the WiFi solution for the whole WiFi/MEMS integrated solution. The proposed MEMS solution has been discussed in detail in the last section. In this section, the focus is mainly on the WiFi part and LC integration of WiFi and MEMS sensors. Trilateration is used to estimate WiFi positions and their variances. In LC integration, WiFi positions, with variances less than a pre-set threshold, are selected as the updates for MEMS sensors.

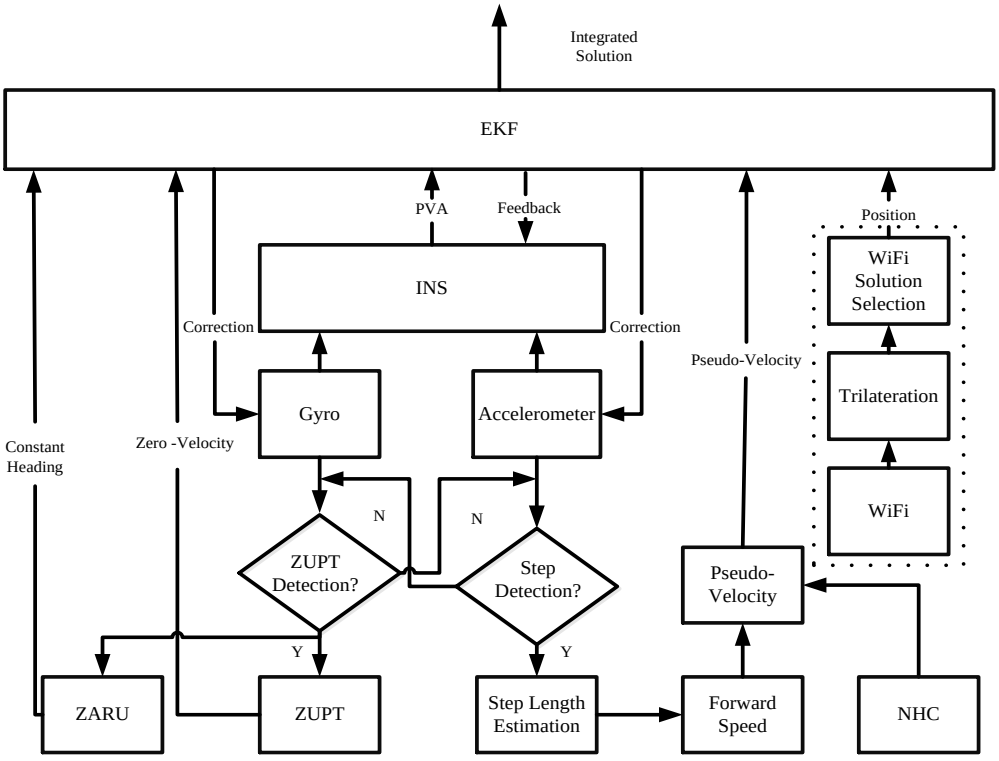


Figure 5-2 Block diagram of the LC integration of WiFi and MEMS sensors.

WiFi is usually used as an update for MEMS sensors in an indoor environment to improve the navigation performance if it is available. Both trilateration-based and fingerprinting-based WiFi positioning solutions can be used as updates in LC integration. In this research, trilateration is selected for WiFi positioning for the following three reasons: (1) it requires less trajectories to build the database than fingerprinting in the crowdsourcing system; (2) the memory cost for trilateration-based database is much less than fingerprinting-based database; and (3) LC integration of trilateration-based WiFi positioning solution and MEMS sensors can be used to compare with the proposed TC integration of WiFi and MEMS sensors. However, a trilateration-based WiFi solution can be noisy due to the complex characteristics of an indoor environment. Therefore, when using the LC integration, it is significant to use an approach to select accurate WiFi positions. It is fortunate that variances of WiFi positions are estimated in the state covariance matrix of the LSQ. Although these variances are not perfectly estimated, they still can be used as a rough indicator for selecting the WiFi positions for LC integration. In this research, WiFi positions with variances less than 20 square meters are chosen as the updates for the MEMS sensors. The misclosure of the WiFi-based position measurements is given by

$$\delta z_{WiFi} = \left(\begin{bmatrix} \varphi \\ \lambda \\ h \end{bmatrix}_{MEMS} - \begin{bmatrix} \varphi \\ \lambda \\ h \end{bmatrix}_{WiFi} \right) \begin{bmatrix} M+h \\ (N+h)\cos\varphi \\ 1 \end{bmatrix}^{T} \tag{5-11}$$

where φ_{MEMS}, λ_{MEMS}, and h_{MEMS} are MEMS-estimated latitude, longitude, and altitude; φ_{WiFi}, λ_{WiFi}, and h_{WiFi} are WiFi-based latitude, longitude and altitude. M is the meridian radius of the earth's curvature; and N is the prime vertical radius of the earth's curvature. The observation equation for the WiFi position measurements is formulated as

$$\delta z_{WiFi} = H_{WiFi}\delta x + v_{WiFi} \tag{5-12}$$

where v_{WiFi} represents the measurement noise of the WiFi positions; and H_{WiFi} represents the corresponding design matrix which can be expressed as

$$H_{WiFi} = \begin{bmatrix} (M+h) & 0 & 0 & \\ 0 & (N+h)\cos\varphi & 0 & 0_{3\times 12} \\ 0 & 0 & 1 & \end{bmatrix} \tag{5-13}$$

The covariance matrix, R_{WiFi}, for the WiFi-based position measurements is given by

$$R_{WiFi} = diag\left(\begin{bmatrix} \sigma_{lat}^2 & \sigma_{lon}^2 & \sigma_{alt}^2 \end{bmatrix}\right) \tag{5-14}$$

where σ_{lat}^2, σ_{lon}^2, and σ_{alt}^2 represent the variances of $\begin{bmatrix} \varphi & \lambda & h \end{bmatrix}_{WiFi}^T$ in meters.

5.3 TC Integration of WiFi/MEMS for Indoor Navigation

The block diagram of the proposed TC WiFi/MEMS integration for indoor pedestrian navigation is shown in Figure 5-3. The proposed system is made up of three parts: the proposed MEMS solution based on the PDR/INS integration and motion constraints, WiFi-based range estimation, and EKF-based TC integration. The proposed MEMS solution has been given in detail in Section 5.1. In the TC integration, the proposed MEMS solution is used to generate the MEMS-based range information. In WiFi-based range estimation, RSS values from the handheld devices pass to the propagation model to generate the range. In the part of EKF-based TC integration, the range differences between MEMS-based ranges and WiFi-based ranges pass to the EKF to estimate the state vector errors. The estimated state vector errors (3D position, velocity, and attitude error;

accelerometer bias, gyroscope drift; and WiFi RSS bias) are fed back to the INS and to the WiFi range estimation. EKF outputs the final integrated navigation solution for pedestrians in indoor environments.

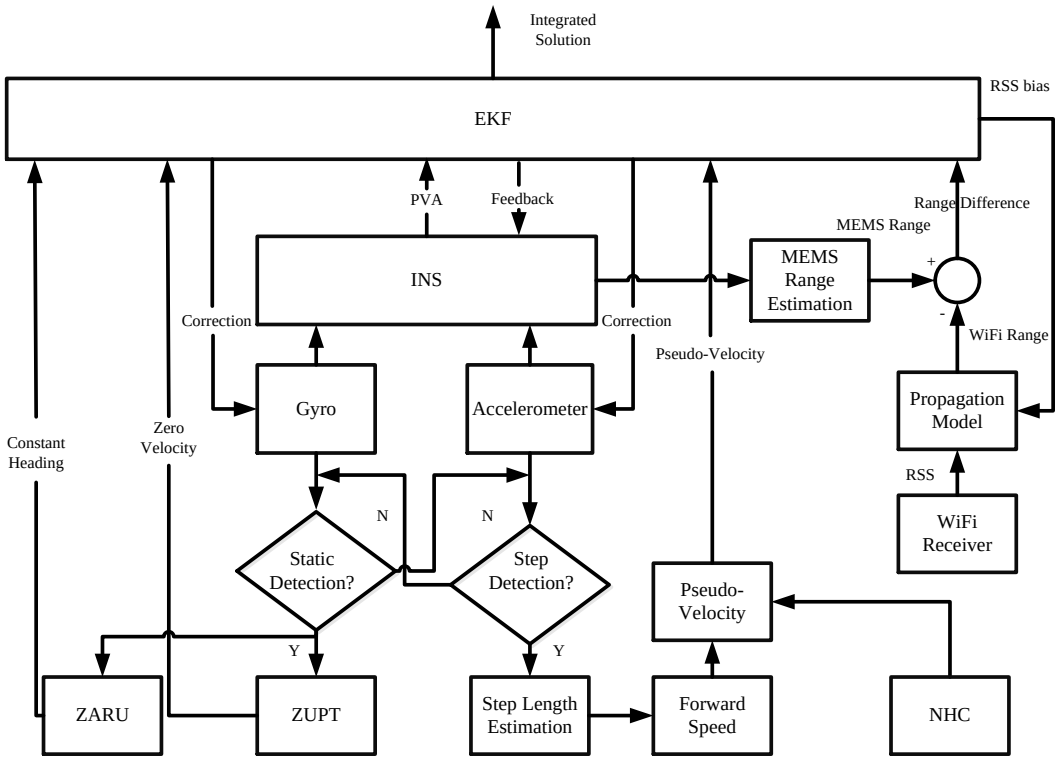


Figure 5-3 Block diagram of the TC integration of WiFi and MEMS sensors.

In the following section, the TC integration of MEMS and WiFi is described in detail, including "MEMS-based range", "WiFi-based range", "system model of TC integration" and "observation model of TC integration". In this research, WiFi-based ranges are calculated based on the WiFi propagation model. The main advantage of TC integration is that WiFi-based ranges can be used to aid MEMS in cases where less than three WiFi APs are observed, whereas LC integration cannot estimate the WiFi positions based on trilateration to aid the MEMS. System tests will show that the proposed TC WiFi/MEMS integration has better performance than the LC integration, especially in an environment with a sparse deployment of WiFi APs.

5.3.1 MEMS-Based Range

TC WiFi/MEMS integration involves the use of new measurement data, namely the MEMS-based range, given by

$$d_{MEMS,k} = \sqrt{\begin{bmatrix}\left(\lambda_{MEMS} - \lambda_{AP,k}\right)\left(N+h\right)\cos\varphi_{MEMS}\end{bmatrix}^2 + \left[\left(\varphi_{MEMS} - \varphi_{AP,k}\right)\left(M+h\right)\right]^2 + \left(h_{MEMS} - h_{AP,k}\right)^2} \tag{5-15}$$

where λ_{MEMS}, φ_{MEMS} and h_{MEMS} represent MEMS position coordinates (longitude, latitude, and altitude); $\lambda_{AP,k}$, $\varphi_{AP,k}$ and $h_{AP,k}$ represent position coordinates of k^{th} WiFi AP (longitude, latitude, and altitude); M represents the meridian radius of the earth curvature; and N represents the prime vertical radius of the earth curvature.

5.3.2 WiFi-Based Range

Recall the typical propagation model, discussed in Chapter 2, and rewrite it as follows:

$$RSS = A - 10 \cdot n \cdot log_{10}(d) + X_{\sigma} \tag{5-16}$$

where RSS represents the received signal strength in dBm at a distance, d, from the transmitter. A represents a constant which depends on several factors: averaged fast and slow fading, transmitter gain, receiver gain, and transmitted power. Therefore, in practice, its value is usually known beforehand (Mazuelas et al. 2009). n represents the path loss exponent with typical values, 2.0 ~ 6.0, in indoors. X_{σ} represents the shadow noise modeled as a Gaussian random variable with zero mean and standard deviation, σ_{RSS}. The range between the receiver and the transmitter can

be estimated by the maximum likelihood estimator (MLE), and the result is given by (Mazuelas et al. 2009):

$$\hat{d}_{RSS} = 10^{(A-RSS)/10\cdot n} \tag{5-17}$$

The experimental standard deviation of RSS values, σ_{RSS} , is almost independent of d . By differentiating the propagation model in (5-16) with respect to d , we obtain

$$\frac{\partial RSS}{\partial d} = -\frac{10\cdot n}{\ln(10)\cdot d} \tag{5-18}$$

Therefore, the standard deviation of the range d is given by

$$\sigma_d = \ln(10)\cdot d\cdot \sigma_{RSS} / 10\cdot n \tag{5-19}$$

σ_d is linearly proportional to d , which illustrates the fact that the uncertainty of the range estimation grows with the range d . Note that there are other propagation models that consider the effects of walls and floors (Bahl and Padmanabhan 2000) (Lott and Forkel 2001). However, they are not suitable for a real-time navigation system because a priori information of walls and floors are usually unavailable.

RSS measurements usually contain a bias for several reasons such as the inaccurate pre-set value of the constant A in (5-16). Therefore, the estimated range, $\hat{d}_{RSS}$, is not equal to the geometric range, d , between the transmitter and the receiver. The RSS bias, b_{RSS} , is considered to compensate the difference between $\hat{d}_{RSS}$ and d . Therefore, the geometric range is given by

$$d = 10^{A-RSS-b_{RSS}/10\cdot n} = 10^{A-RSS/10\cdot n}\cdot 10^{-b_{RSS}/10\cdot n} = \hat{d}_{RSS}\cdot 10^{-b_{RSS}/10\cdot n} \tag{5-20}$$

By reorganizing (5-20), we obtain

$$\hat{d}_{RSS} = d\cdot 10^{b_{RSS}/10\cdot n} = d\cdot f\left(b_{RSS}\right) \tag{5-21}$$

where $f\left(b_{RSS}\right) = 10^{b_{RSS}/10\cdot n}$. We assume the absolute value of the RSS bias is less than 4 dBm. Due to $2 \le n \le 6$ and $abs\left(b_{RSS}\right) \le 4dbm$, $10^{b_{RSS}/10\cdot n}$ is close to zero. Therefore, $f\left(b_{RSS}\right)$ is linearized at the point of $b_{RSS} = 0$ by using the Taylor expansion, and the result is given by

$$\begin{aligned} f\left(b_{RSS}\right) &= f\left(0\right) + \frac{\partial f\left(b_{RSS}\right)}{\partial b_{RSS}}|_{b_{RSS}=0}\cdot b_{RSS} + \cdots \\ &\approx 1 + \left(\ln 10/10\cdot n \cdot 10^{b_{RSS}/10\cdot n}\right)|_{b_{RSS}=0}\cdot b_{RSS} \\ &= 1 + \ln 10\cdot b_{RSS}/10\cdot n \end{aligned} \tag{5-22}$$

Substitute (5-22) into (5-21), we obtain the relationship between $\hat{d}_{RSS}$ and d:

$$\hat{d}_{RSS} = d + \left(\ln 10\cdot d/10\cdot n\right) b_{RSS} \tag{5-23}$$

In the TC WiFi/MEMS integration, the RSS bias b_{RSS} is also put in the state vector, and estimated through the EKF. Therefore, the system can also improve the estimation of WiFi-based ranges by using the feedback of the estimated RSS bias, further improving the navigation performance.

5.3.3 System Model of TC Integration

In TC integration of MEMS and WiFi, error states in the EKF consist of two parts. The first part is the sensor error states. Its system dynamic equation is given as

$$\delta \dot{x}_s = F_s \delta x_s + G_s w_s \tag{5-24}$$

As per discussion in Chapter 2, the sensor error state vector, δx_s, contains 15 states (3D position, velocity, and attitude error; accelerometer bias, gyroscope drift). For the details about the dynamic matrix, F_s, please refer to (Aggarwal et al. 2010). $w_s = \begin{bmatrix} w_1 & \cdots & w_{15} \end{bmatrix}^T$, in which the elements comply with the assumptions of zero-mean and Gaussian distributed white noise and are uncorrelated with each other. Thus, the corresponding G_s is a unit matrix with a rank of 15.

The second part of the error states is the WiFi error state. In this research, WiFi RSS bias is used to compensate the error in the propagation model to estimate a more accurate range. WiFi RSS bias is modeled as a random walk process. The differential equation can be written as follows:

$$\dot{b}_{RSS} = w_{b_{RSS}} \tag{5-25}$$

where $w_{b_{RSS}}$ is the white noise. The WiFi system dynamic model is given by

$$\delta \dot{x}_W = F_W \delta x_W + G_W w_W \tag{5-26}$$

where $\delta x_W = b_{RSS}$, $F_W = 0$, $G_W = 1$, and $w_W = w_{b_{RSS}}$.

By combining (5-24) and (5-26), we have the following system model for the TC WiFi/MEMS integration.

$$\begin{bmatrix} \delta\dot{x}_s \\ \delta\dot{x}_W \end{bmatrix} = \begin{bmatrix} F_s & 0 \\ 0 & F_W \end{bmatrix} \begin{bmatrix} \delta x_s \\ \delta x_W \end{bmatrix} + \begin{bmatrix} G_s & 0 \\ 0 & G_W \end{bmatrix} \begin{bmatrix} w_s \\ w_W \end{bmatrix} \quad \text{(5-27)}$$
$$i.e.\ \ \delta\dot{x} = F\delta x + Gw$$

5.3.4 Observation Model of TC Integration

The range differences between the WiFi-based ranges and the MEMS-based ranges are used as the observation vector, $\delta\mathbf{z}$, in the TC EKF. By assuming there are m APs in-view, the measurements can be written as

$$\delta\mathbf{z} = \begin{bmatrix} \delta z_{1,range} \\ \vdots \\ \delta z_{\mathrm{m},range} \end{bmatrix} = \begin{bmatrix} d_{MEMS,1} - d_{WiFi,1} \\ \vdots \\ d_{MEMS,\mathrm{m}} - d_{WiFi,\mathrm{m}} \end{bmatrix} \quad \text{(5-28)}$$

where $d_{MEMS,\mathrm{k}}$ is the MEMS-estimated range based on (5-15), and $d_{WiFi,\mathrm{k}}$ is the k^{th} AP's WiFi-based range measurement. Through (5-23), the WiFi-based range of the k^{th} AP is given by

$$d_{WiFi,\mathrm{k}} = d_k + \left(\ln 10 \cdot d_k \big/ 10 \cdot n\right) b_{RSS} + v_{d_k} \quad \text{(5-29)}$$

where v_{d_k} is the white noise of $d_{WiFi,\mathrm{k}}$. d_k represents the geometric range between the pedestrian and the k^{th} WiFi AP, which is expressed as

$$d_k = \sqrt{\left[\left(\lambda - \lambda_{AP,k}\right)\left(N+h\right)\cos\varphi\right]^2 + \left[\left(\varphi - \varphi_{AP,k}\right)\left(M+h\right)\right]^2 + \left(h - h_{AP,k}\right)^2} \quad \text{(5-30)}$$

where λ, φ, and h represent the filtered pedestrian's coordinates (longitude, latitude, and altitude); $\lambda_{AP,k}$, $\varphi_{AP,k}$, and $h_{AP,k}$ represent the coordinates of the k^{th} WiFi AP (longitude, latitude,

and altitude). By using the Taylor expansion for (5-29) and ignoring high-order errors, the range error model is given in

$$\delta d \approx e_k^T \cdot \begin{bmatrix} \delta\varphi & \delta\lambda & \delta h \end{bmatrix}^T + \left(\ln 10 \cdot d_k \Big/ 10 \cdot n \right) b_{RSS} \tag{5-31}$$

where

$$e_k = \begin{bmatrix} e_{kx} \\ e_{ky} \\ e_{kz} \end{bmatrix} = \left(1 + \ln 10 \cdot b_{RSS} \Big/ 10 \cdot n \right) \begin{bmatrix} \left(\varphi - \varphi_{AP,k} \right) \left(M + h \right) \Big/ d_k \\ \left(\lambda - \lambda_{AP,k} \right) \left(N + h \right) \cos\varphi \Big/ d_k \\ h - h_{AP,k} \Big/ d_k \end{bmatrix} \tag{5-32}$$

Therefore, the observation model for the range differences is given by

$$\begin{aligned} \delta z_d &= \begin{bmatrix} d_{MEMS,1} - d_{WiFi,1} \\ \vdots \\ d_{MEMS,\mathrm{m}} - d_{WiFi,\mathrm{m}} \end{bmatrix} \\ &= \begin{bmatrix} e_{1x} & e_{1y} & e_{1z} \\ \vdots & \vdots & \vdots \\ e_{mx} & e_{my} & e_{mz} \end{bmatrix} \begin{bmatrix} \delta\varphi \\ \delta\lambda \\ \delta h \end{bmatrix} + \begin{bmatrix} \ln 10 \cdot d_1 \Big/ 10 \cdot n \\ \vdots \\ \ln 10 \cdot d_m \Big/ 10 \cdot n \end{bmatrix} b_{RSS} + \begin{bmatrix} v_{d_1} \\ \vdots \\ v_{d_m} \end{bmatrix} \\ &= G_{m\times 3} \begin{bmatrix} \delta\varphi & \delta\lambda & \delta h \end{bmatrix}^T + B_{m\times 1} \cdot b_{RSS} + v_{d,m\times 1} \end{aligned} \tag{5-33}$$

Finally, the observation model for TC integration is written as

$$\delta z = H \delta x + v \tag{5-34}$$

where $\delta z = \delta z_d$ represents the measurement vector, and $v = v_{d,m\times1}$ represents the measurement noise vector, and H is the design matrix, which is expressed as

$$H = \begin{bmatrix} G_{m\times3} & 0_{m\times12} & B_{m\times1} \end{bmatrix} \tag{5-35}$$

5.4 Test Results and Performance Analysis

To evaluate the performance of the proposed pedestrian navigation algorithms, several experiments were performed with three smartphones (Samsung Galaxy S III). Three pedestrians were involved in collecting field experiment data. Smartphones that contain an accelerometer triad, a gyroscope triad, and a WiFi receiver were used to collect the experimental data. The field experiment data was collected in building E (about 120m × 40m) as shown in Figure 5-4. Three tasks were carried out in the field tests. The first task validated the performance of the proposed PDR/INS integrated MEMS pedestrian navigation algorithm. This section also compared the proposed MEMS solution with traditional PDR and INS algorithms. The second task showed the performance of the proposed WiFi/MEMS LC integration algorithm. This section also compared the proposed WiFi/MEMS LC integration solution with the proposed MEMS solution (PDR/INS), PDR, and INS solutions. The last test demonstrated the performance of the proposed WiFi/MEMS TC integration, and compared it with WiFi/MEMS LC integration, the proposed MEMS solution (PDR/INS), PDR, and INS.

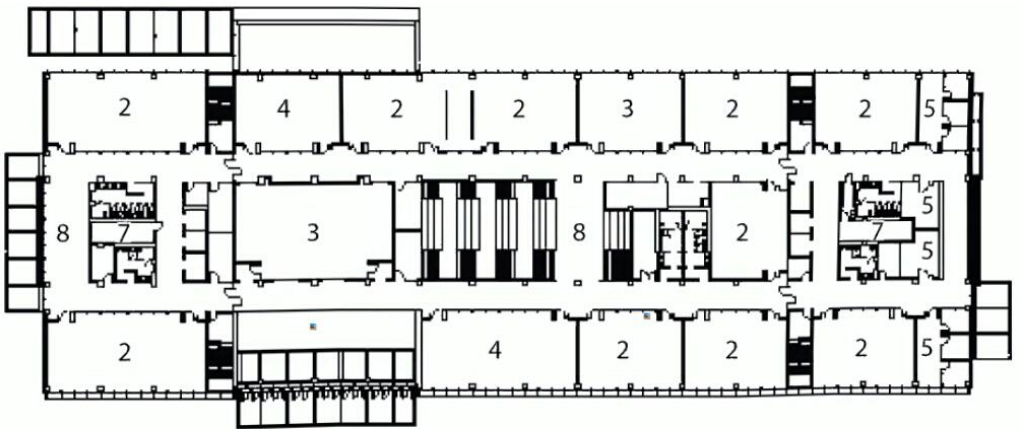
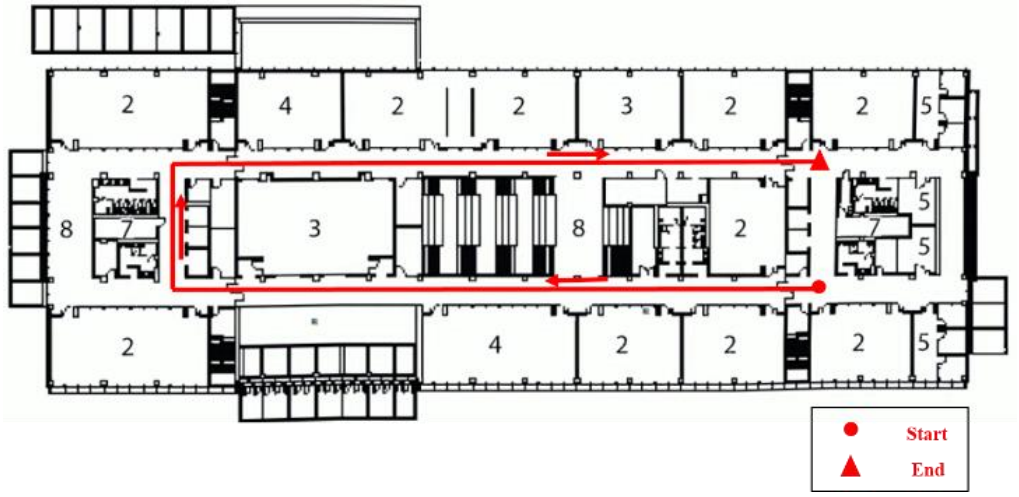


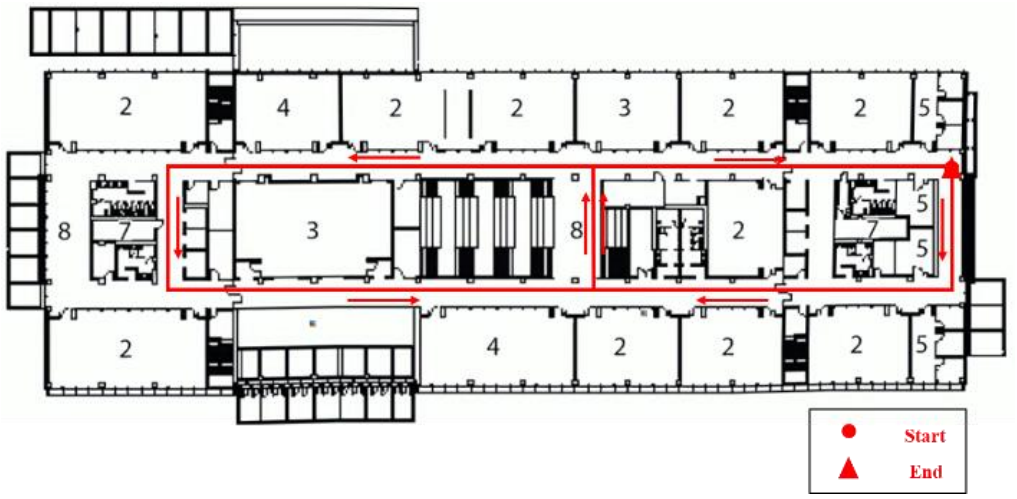
Figure 5-4 Field test area: building E.

5.4.1 PDR/INS Integrated MEMS Pedestrian Navigation

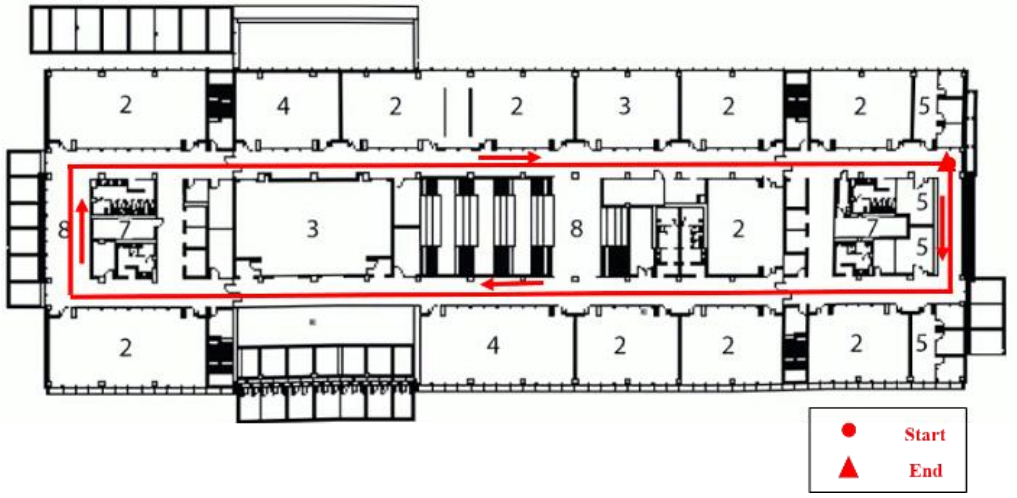
Many experimental trajectories, collected by different pedestrians with various smartphones in the building E, and three trajectories were selected to evaluate the performance of the proposed MEMS solution as shown in Figure 5-5. These three trajectories are also utilized in subsection 5.4.2 to illustrate the performance of WiFi/MEMS LC integration.



(a)



(b)



(c)

Figure 5-5 Three experimental trajectories in building E: (a) Trajectory I, (b) Trajectory II, and (c) Trajectory III.

Performance of the proposed PDR/INS integrated MEMS pedestrian navigation algorithm is mainly illustrated by using trajectory I. Two other trajectories are also used to test the performance of the proposed MEMS navigation solution, and the results are summarized in subsection 5.4.2. The proposed MEMS solution, PDR solution, and the reference trajectory in experimental trajectory I are shown in Figure 5-6. Note that reference trajectory is provided by using several pre-set markers' positions from the floor plan of the building. The Cumulative error percentages

of PDR and the proposed MEMS solution in Trajectory I is shown Figure 5-7. In Figure 5-7, the maximum navigation errors for the proposed method and traditional PDR are about 13 and 32 meters, separately. This shows that the proposed method performs better than the PDR. The average heading drift of the proposed method is also smaller than the PDR.

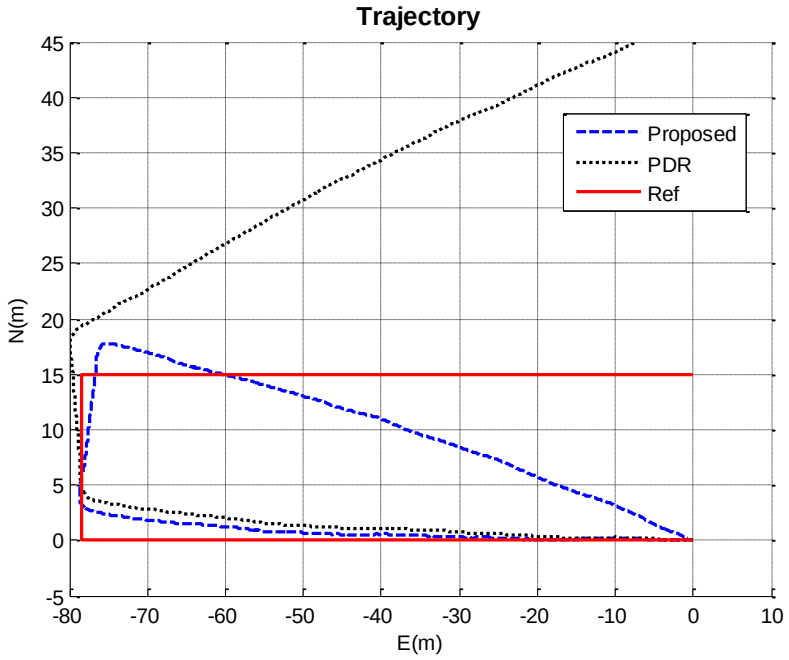


Figure 5-6 Trajectories of PDR, PDR/INS integrated MEMS solution, and reference.

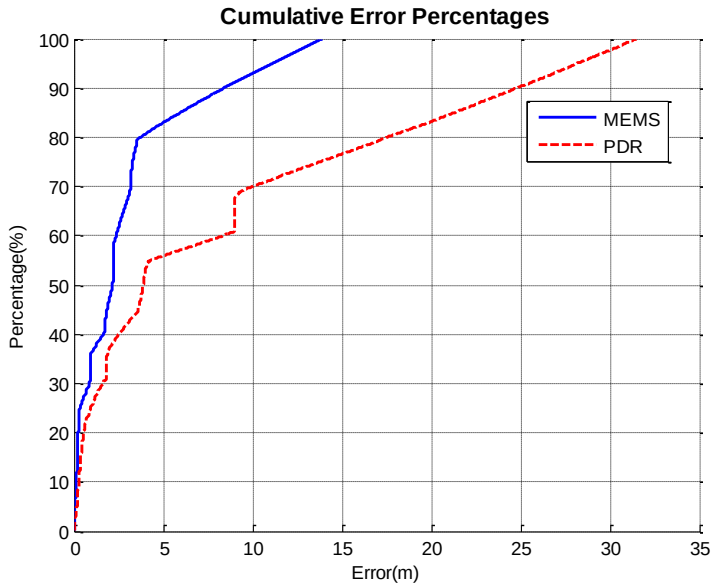
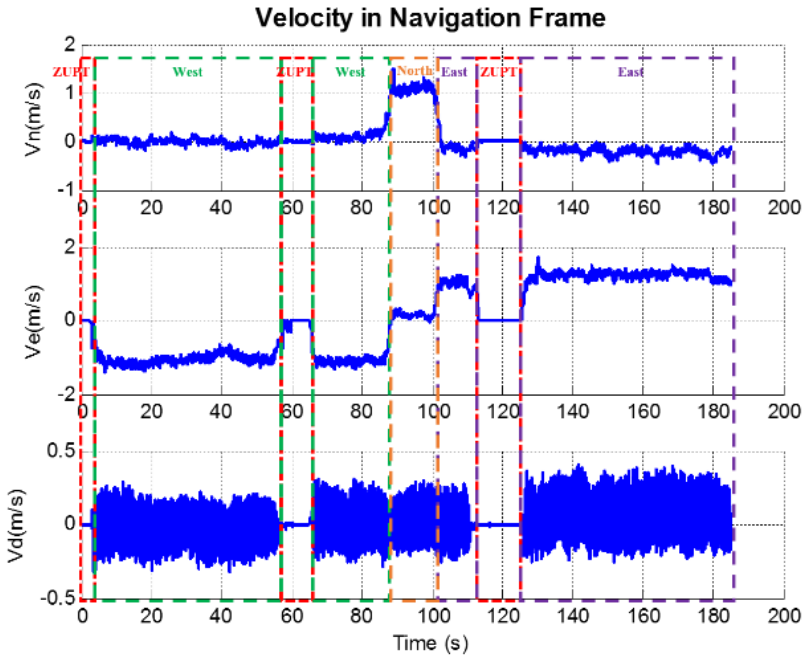
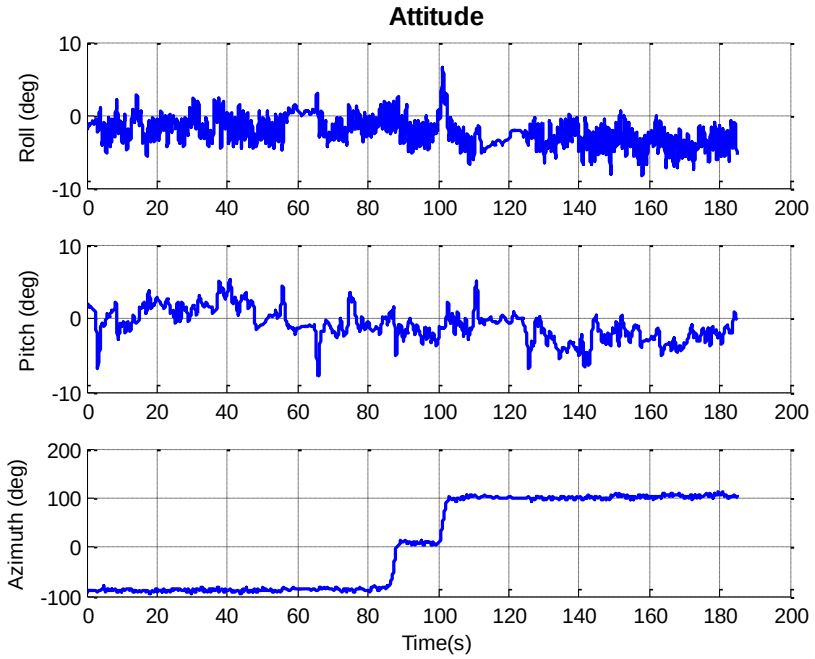


Figure 5-7 Cumulative error percentages of PDR and the proposed MEMS solution (Trajectory I).

The velocity solution of the proposed algorithm is shown in Figure 5-8 (a). Figure 5-8 clearly shows the user's moving status: a) keeps static (ZUPT), b) walks west, c) keeps static (ZUPT), d) walks west, e) walks north, f) walks east, g) keeps static (ZUPT), and h) walks east. The moving trend successfully fits the trajectory in Figure 5-6. The walking speed is in the typical range of a normal person. The pseudo-velocity update and ZUPT play an important role in accurately estimating the user's velocity. Without the pseudo-velocity update and ZUPT, the estimated velocity and position solution drifts quickly. The attitude solution of the proposed method is shown in Figure 5-8 (b). Roll and pitch angles are between -10 degrees and 10 degrees in this trajectory. The estimated azimuth trend is as follows: a) about -90 degrees, b) about 0 degree, and c) about 100 degrees. The true azimuth trend is as follows: a) -90 degrees, b) 0 degree, and c) 90 degrees. The estimated azimuth from the proposed method is close to the true azimuth.



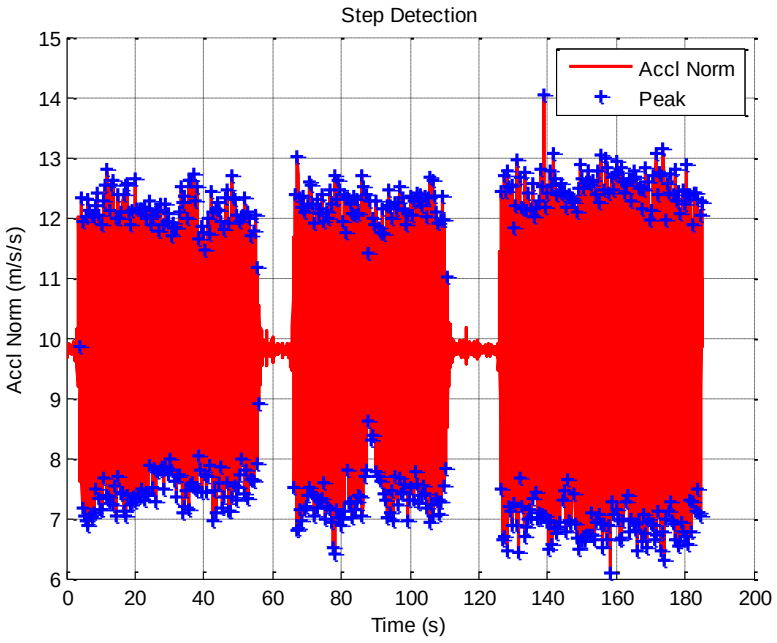
(a) Velocity



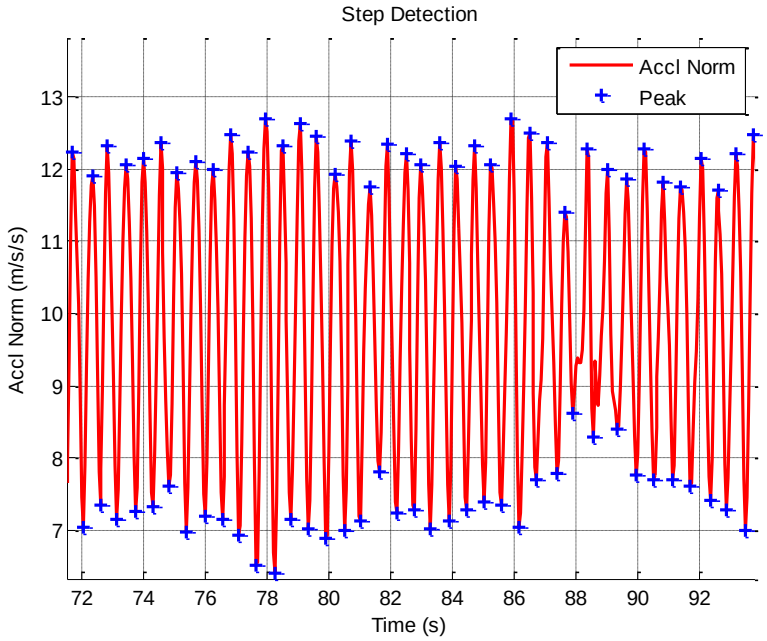
(b) Attitude

Figure 5-8 Velocity and attitude solutions of the proposed MEMS solution.

The results of the step detection, step length estimation and pseudo-velocity are shown and discussed as follows. The step detection results are shown in Figure 5-9. In this figure, acceleration norm and detected peaks are described by "-" and "+". Figure 5-9 (b) zooms in on some parts of the trajectory which shows that the step detection algorithm can successfully detect the peaks and steps. The step length estimation result is shown in Figure 5-10. The user's step length is around 0. 64 meters as shown in this figure. The step-length-derived pseudo-velocity is shown in Figure 5-11. The pseudo-velocity is calculated from the step length. The average velocity of this user is about 1.20 m/s.



(a)



(b)

Figure 5-9 Step detection results: (a) whole trajectory and (b) zoom-in of some parts of the trajectory.

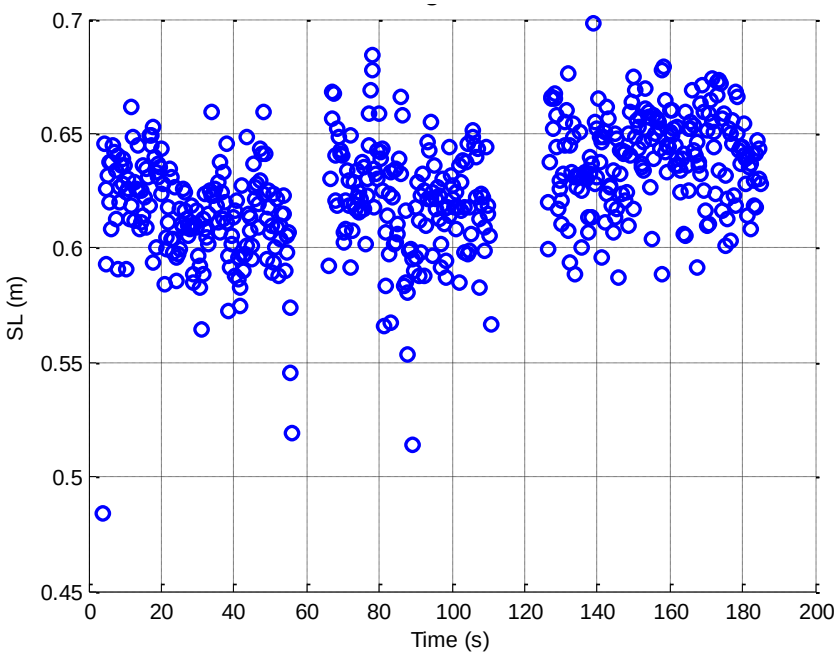


Figure 5-10 Result of step length estimation.

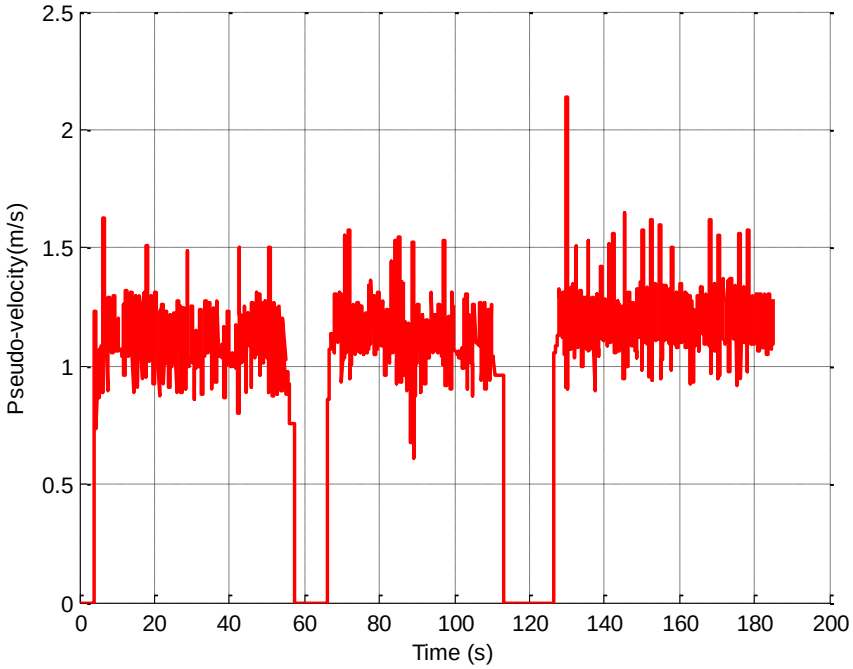


Figure 5-11 Result of pseudo-velocity from step length.

In order to illustrate the performance of the proposed PDR/INS integrated MEMS navigation solution, results of the stand-alone PDR and INS navigation are discussed next. Pure PDR results are shown in Figure 5-6 and Figure 5-12. As we discussed before, the maximum navigation error of PDR is about 32 meters in Figure 5-6, which is worse than the proposed method. In Figure 5-12, user's 2D velocities (north velocity and east velocity) are clearly shown. User's azimuth information is also shown in Figure 5-12.

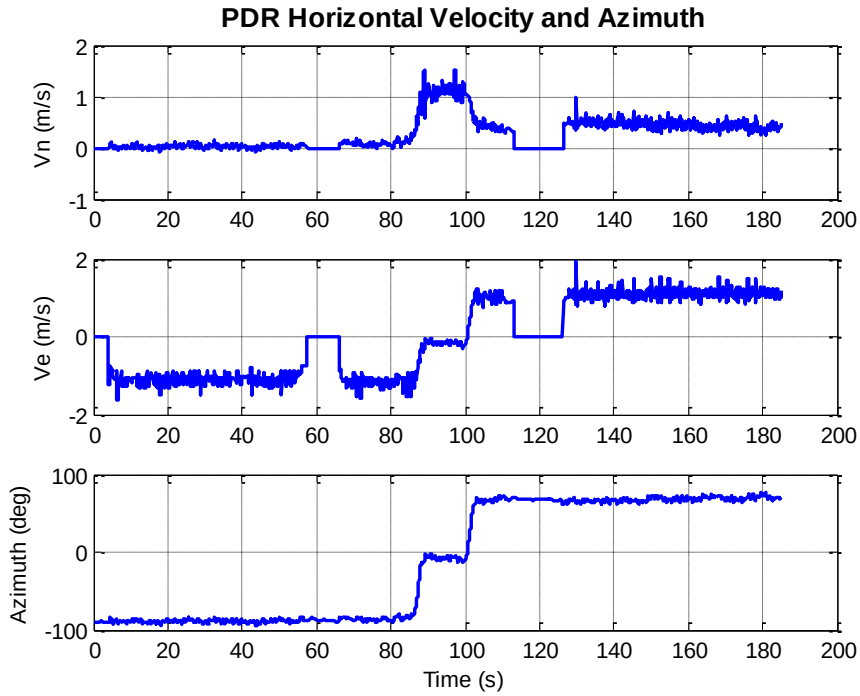
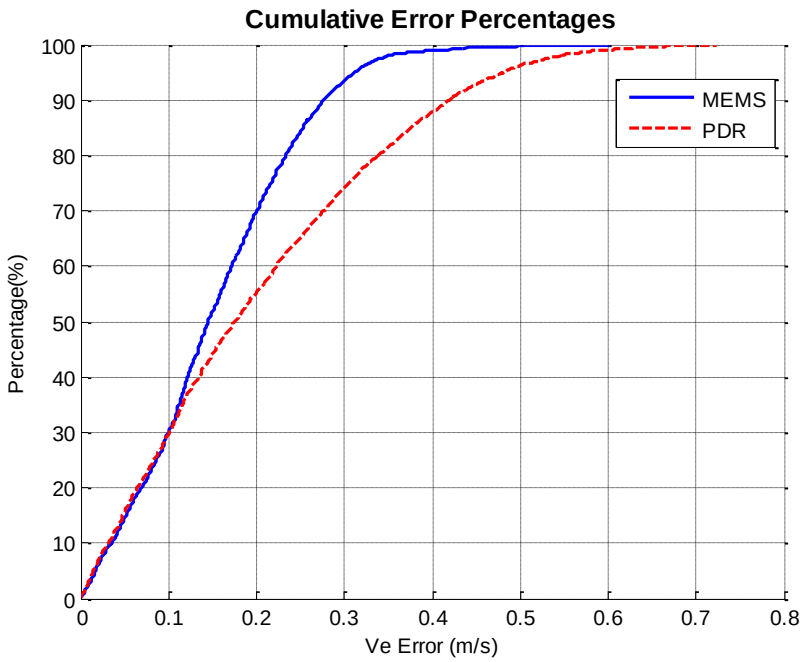


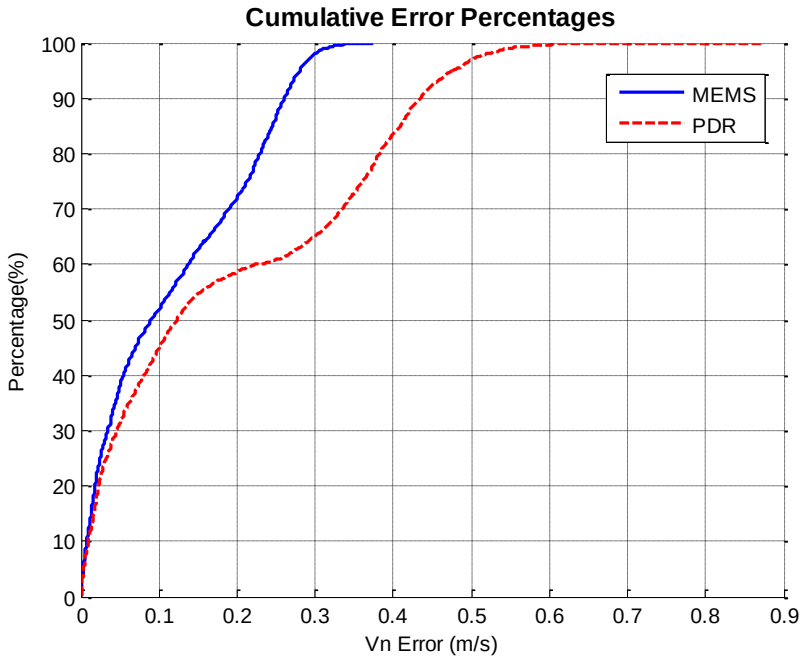
Figure 5-12 Results of PDR horizontal velocity and azimuth.

Cumulative error percentages of horizontal velocities and azimuth from PDR and the proposed method are shown in Figure 5 13. Figure 5 13 (a) and (b) show the estimation errors of east velocity and north velocity by using proposed method and PDR. In Figure 5 13 (a), the 50% and 95% errors of estimated east velocity by using PDR are about 0.17m/s and 0.42m/s, respectively, while the 50% and 95% errors of estimated east velocity by using the proposed method are about 0.14m/s and 0.27m/s, respectively. In Figure 5 13 (b), the 50% and 95% errors of estimated north velocity by using PDR are about 0.12m/s and 0.43m/s, respectively. The 50% and 95% errors of estimated north velocity by using the proposed method are about 0.09m/s and 0.26m/s, respectively. In Figure 5 13 (c), estimation errors of azimuth by using proposed method and PDR are shown. The 50% and 95% azimuth errors by using PDR are about 8 degrees and 23 degrees, respectively. The 50% and 95% azimuth errors by using the proposed method are about 6 degrees and 14 degrees,

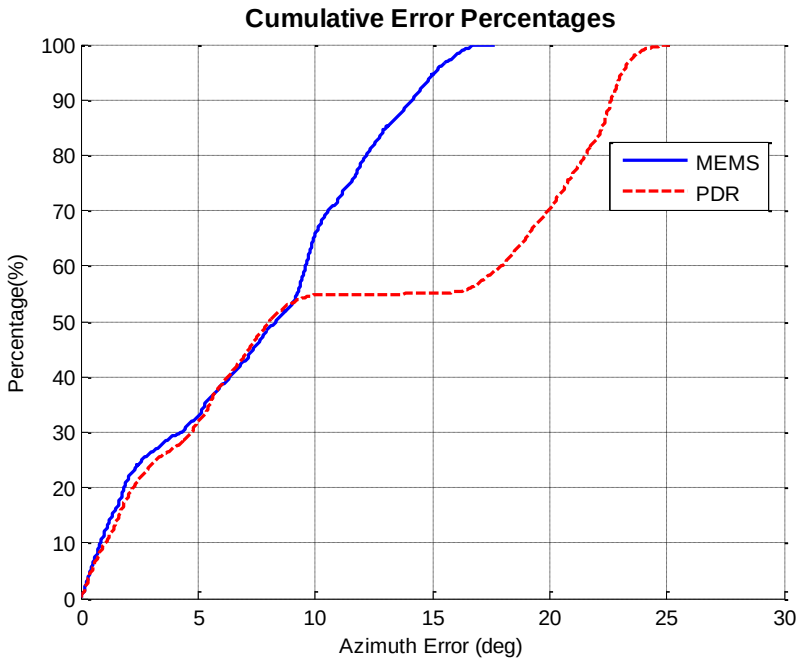
respectively. In sum, Figure 5 13 clearly shows that the proposed method has a more accurate estimation for horizontal velocities and azimuth.



(a) East Velocity



(b) North Velocity



(c) Azimuth

Figure 5-13 Cumulative error percentages of PDR and the proposed MEMS solution (Trajectory I).

Results of the pure INS algorithm are shown in Figure 5-14, Figure 5-15, and Figure 5-16. Figure 5-14 show the horizontal trajectory by using pure INS algorithm. The maximum drift reaches up to 73km at the end of the end of the trajectory. Therefore, pure INS algorithm cannot provide an accurate navigation solution for pedestrians by using handheld devices. The INS velocity solution is shown in Figure 5-15. In this figure, it is clear that the velocity solution are not located in the typical range of pedestrian walking velocity. The velocity drifts to several hundred meters per second by the end of the trajectory. Figure 5-16 shows the attitude solution from the INS algorithm. The estimated roll and pitch can reach about ±50 degrees sometimes, which highly drift from the true roll and pitch range (-10 degree ~ 10 degree). The estimated heading also drifts from the true value when compared with the reference trajectory in Figure 5-6. Figure 5-14, Figure 5-15, and

Figure 5-16 show that estimated position, velocity, and attitude all highly drift from true values by using pure INS algorithm.

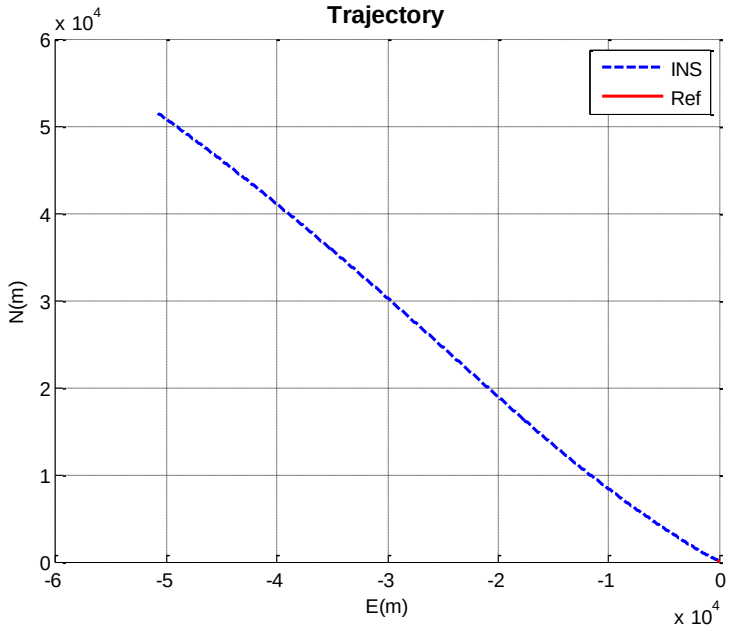


Figure 5-14 INS trajectory and reference.

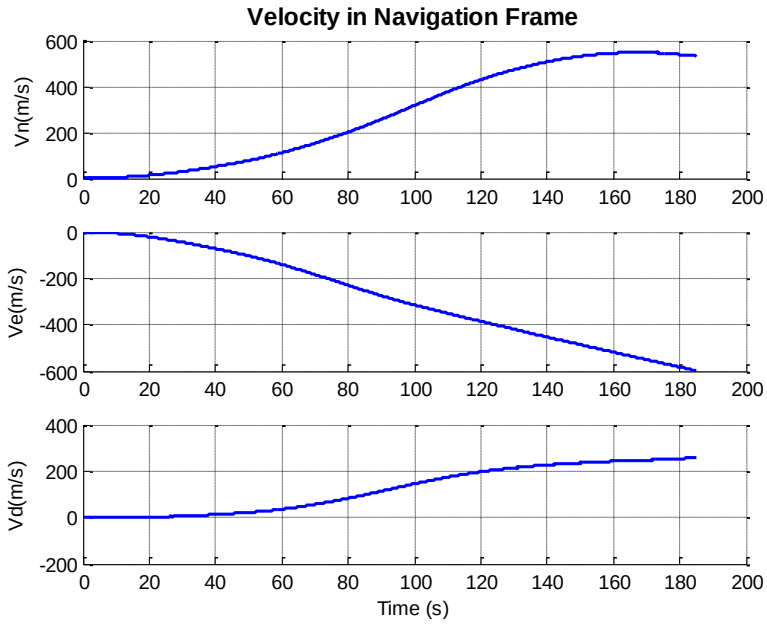


Figure 5-15 INS velocity result.

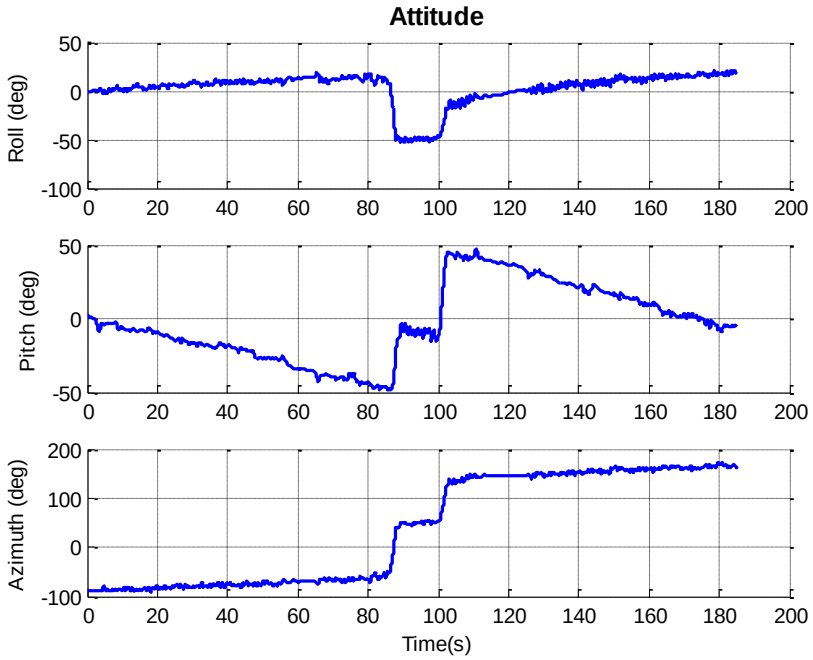
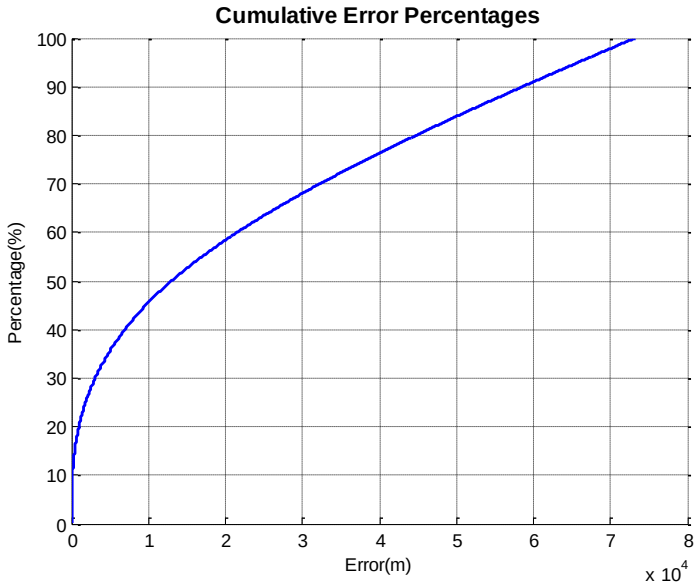
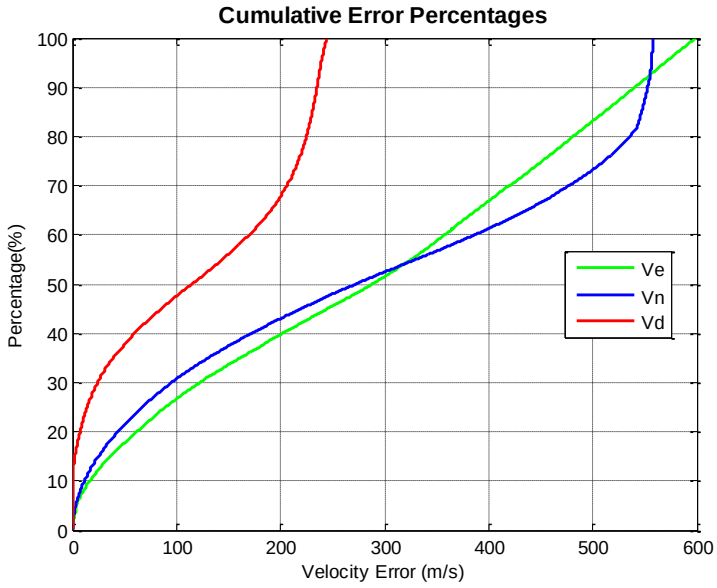


Figure 5-16 INS attitude result.

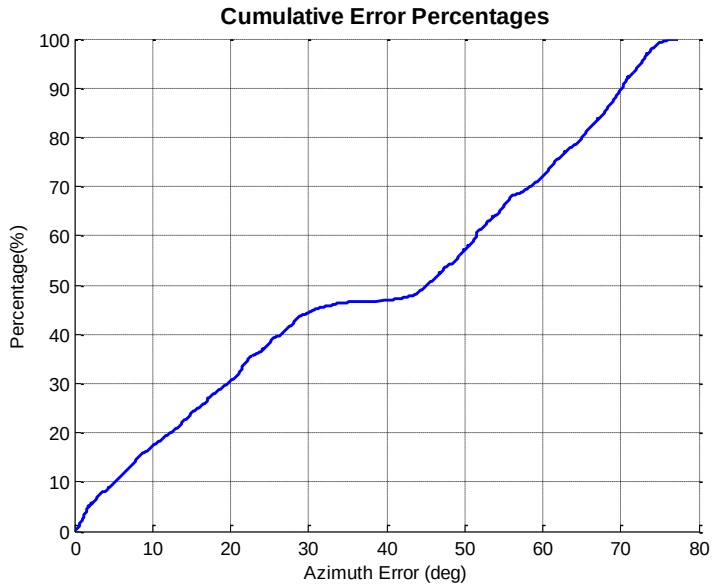
Cumulative error percentages of horizontal position, velocities, and azimuth of INS are shown in Figure 5-17. In Figure 5-17, the horizontal position error accumulates quickly, and extends to about 80km by the end of the trajectory. In Figure 5-17, the maximum velocity error achieves several hundred meters per second. In Figure 5-17, the maximum heading error reaches about 77 degrees. As shown in Figure 5-17, INS navigation errors are much larger than errors of the PDR and the proposed algorithm. Therefore, the INS solution will not be shown in the next set of figures for the comparison of navigation performance.



(a) Horizontal position



(b) Velocity



(c) Azimuth

Figure 5-17 Cumulative error percentages of INS mechanization (Trajectory I).

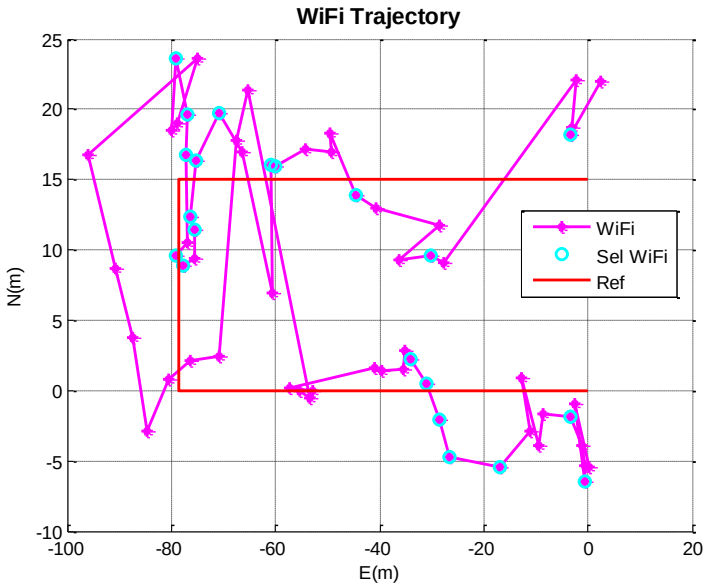
Table 5-1 Positioning performance of different algorithms in Trajectory I

Algorithm	Error (m)			
	Maximum	**Minimum**	**Mean**	**RMS**
INS	72441.04	0.00	21458.20	30788.21
PDR	31.48	0.00	8.44	12.57
PDR/INS Integration	13.93	0.00	2.91	4.40

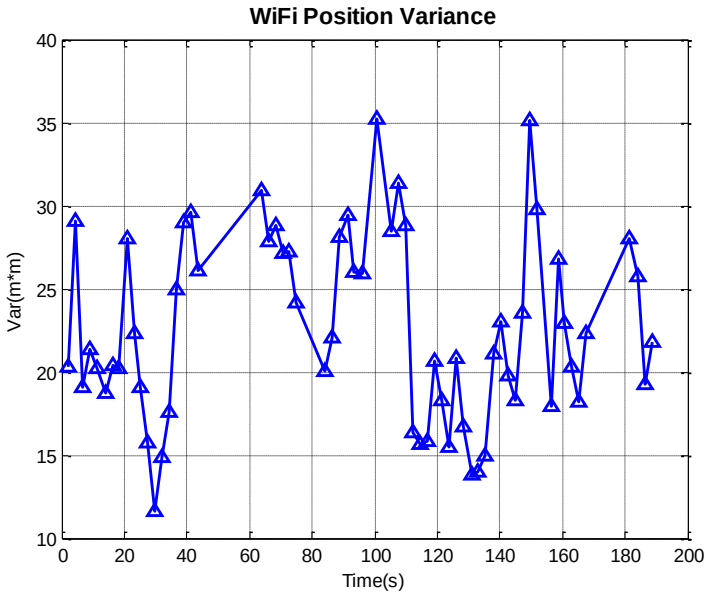
5.4.2 LC WiFi/MEMS Integration for Indoor Navigation

The results (position and variance) of trilateration-based WiFi positioning of Trajectory I are shown in Figure 5-18. The results contain very large errors reaching 15 meters at many points. To improve the performance, some WiFi positions with large variances are left out of the integration. In this research, WiFi positions with variances less than 20 m^2 are selected for integration as shown

in Figure 5-18 (a). The threshold, 20 m^2, is obtained by field tests. In Figure 5-18 (a), "Sel WiFi" represents the selected WiFi positions with variances less than 20 $m^{2.}$



(a)



(b)

Figure 5-18 Trilateration-based WiFi positioning solution: (a) trajectory and (b) variances.

The trajectory of the proposed LC integration of MEMS/WiFi in Trajectory I is shown as the "blue dash line" in Figure 5-19. This trajectory is taken by "Pedestrian 1" using "Smartphone A". The proposed MEMS solution, PDR and the reference are shown as a "green dash dot line", "black dot line" and "red solid line", respectively. The proposed MEMS solution drifts less than the PDR solution. Selected WiFi positions are also shown as "circles" in Figure 5-19. With the help of the selected WiFi positions, the LC MEMS/WiFi integrated solution drifts less than the proposed MEMS solution as shown in Figure 5-19. The performance of LC solution is also better than the WiFi solution when one compares Figure 5-18 (a) with Figure 5-19.

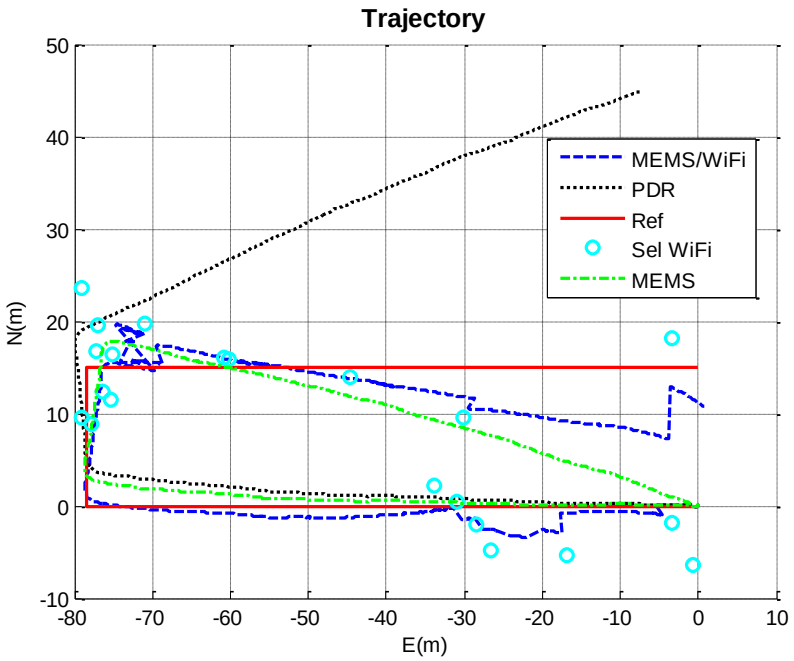


Figure 5-19 Trajectories of PDR, the proposed MEMS solution, and WiFi/MEMS LC integration (Trajectory I: Pedestrian 1, Smartphone A).

The cumulative error percentages of the proposed MEMS solution, PDR, and LC WiFi/MEMS integration (Trajectory I) are shown in Figure 5-20. The positioning performance of different algorithms in Trajectory I are depicted Table 5-2. As shown in Figure 5-20 and Table 2-1, LC

integration has the best navigation performance, and the PDR has the worst navigation performance.

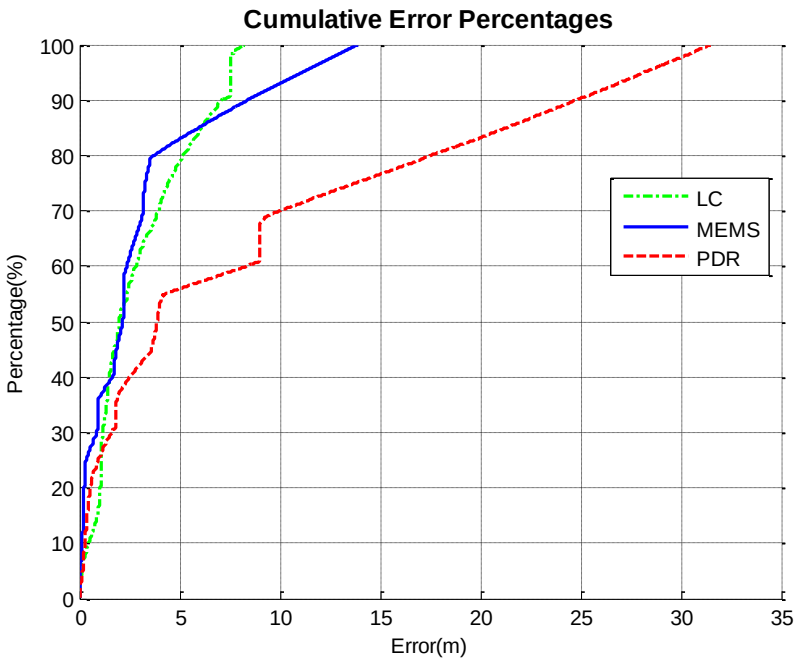


Figure 5-20 Cumulative error percentages of PDR, the proposed MEMS solution, and WiFi/MEMS LC integration (Trajectory I).

Table 5-2 Positioning performance of different algorithms in Trajectory I

Algorithm	Error (m)			
	Maximum	Minimum	Mean	RMS
PDR	31.48	0.00	8.44	12.57
PDR/INS Integration	13.93	0.00	2.91	4.40
LC (MEMS/WiFi)	8.22	0.00	2.91	3.73

The second trajectory of the proposed LC integration of MEMS/WiFi is shown as the "blue dash line" in Figure 5-21 as taken by "Pedestrian 2" using "Smartphone B". Similar to Trajectory I, the

proposed MEMS solution drifts less than the PDR solution, and the LC MEMS/WiFi integration solution drifts even less than the proposed MEMS solution as shown in Figure 5-21. The results of trilateration-based WiFi positioning in Trajectory II are shown in Figure 5-22. The LC integration solution also performs better than the WiFi solution when comparing Figure 5-21 with Figure 5-22.

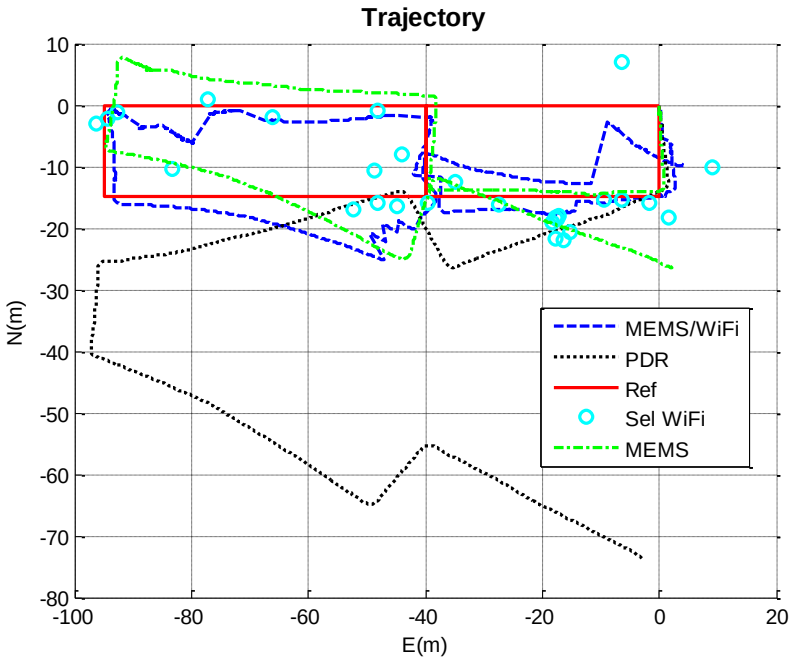


Figure 5-21 Trajectories of PDR, the proposed MEMS solution, and WiFi/MEMS LC integration (Trajectory II: Pedestrian 2, smartphone B).

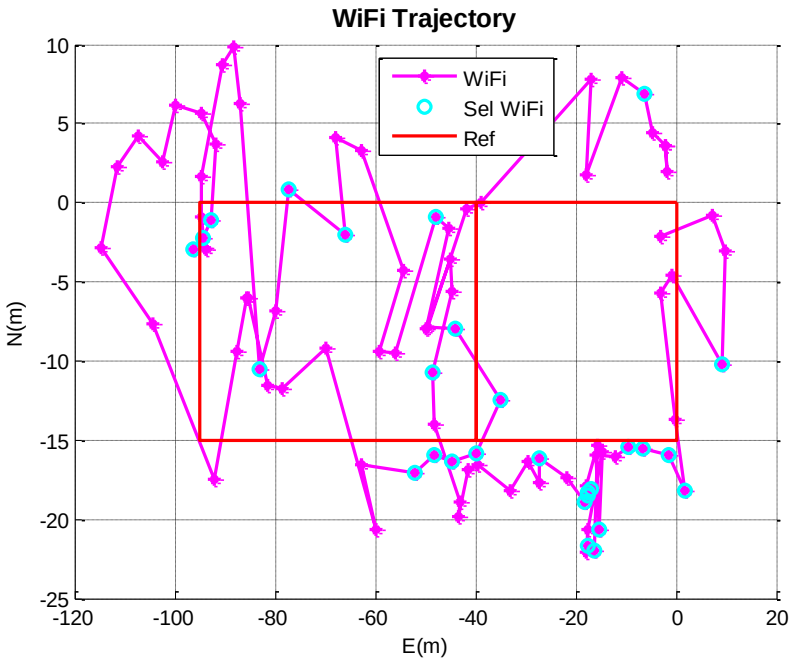


Figure 5-22 WiFi trajectory (Trajectory II).

The cumulative error percentages of the proposed MEMS solution, PDR and LC WiFi/MEMS integration in the second trajectory are shown in Figure 5-23, and the positioning performance of different algorithms for Trajectory II are depicted in Table 5-3. In Trajectory II, the WiFi/MEMS integrated system improves about 3.5 meters (RMS) when compared with the proposed MEMS solution, which is more accurate than in Trajectory I (0.7 meters). This is because the proposed MEMS solution drifts more in Trajectory II due to a longer walking period.

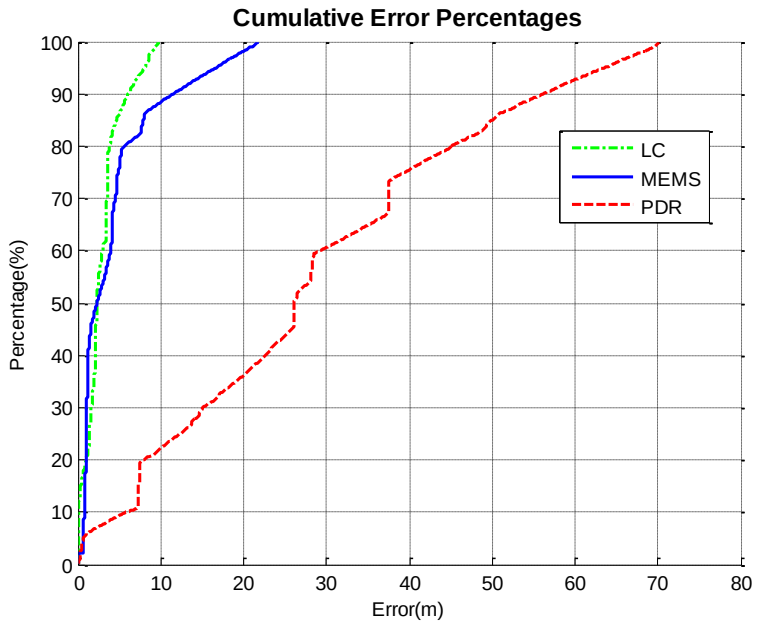


Figure 5-23 Cumulative error percentages of PDR, the proposed MEMS solution, and WiFi/MEMS LC integration (Trajectory II).

Table 5-3 Positioning performance of different algorithms in Trajectory II

Algorithm	Error (m)			
	Maximum	Minimum	Mean	RMS
PDR	75.03	0.04	27.92	34.00
PDR/INS Integration	27.59	0.00	5.72	8.28
LC (MEMS/WiFi)	12.21	0.00	4.17	4.87

The third trajectory of the proposed LC integration of MEMS and WiFi is shown as the "blue dash line" in Figure 5-24 as taken by "Pedestrian 3" using "Smartphone C". Similar to Trajectory I and Trajectory II, the proposed MEMS solution drifts less than the PDR solution, and the LC integrated MEMS/WiFi solution drifts even less than the proposed MEMS solution as shown in Figure 5-24. The results of trilateration-based WiFi positioning in Trajectory III are shown in Figure 5-25. The

LC MEMS/WiFi integration solution also performs better than the WiFi-only solution when comparing Figure 5-24 with Figure 5-25.

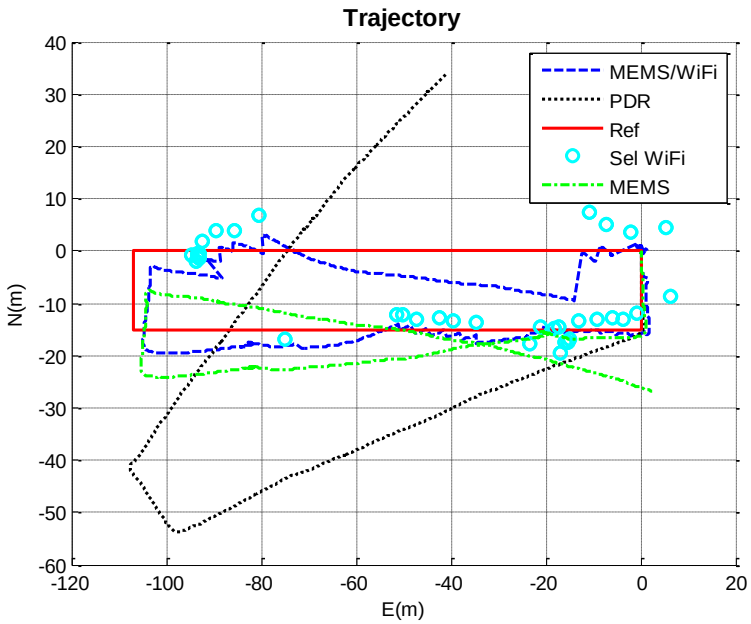


Figure 5-24 Trajectories of PDR, the proposed MEMS solution, and WiFi/MEMS LC integration (Trajectory III: Pedestrian 3, Smartphone C).

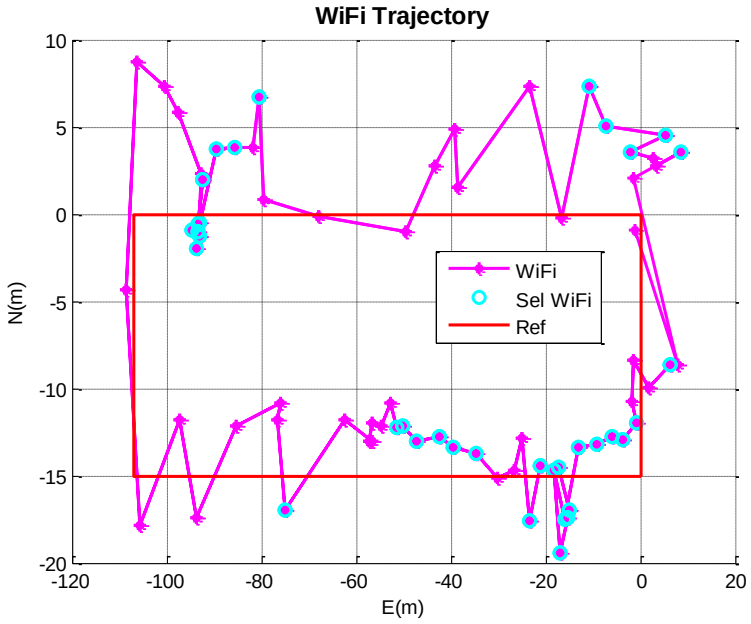


Figure 5-25 WiFi trajectory (Trajectory III).

The cumulative error percentages of the proposed MEMS solution, PDR and LC integration of WiFi/MEMS in the third trajectory are shown in Figure 5-26, and the positioning performance of different algorithms for Trajectory III are depicted in Table 5-4.

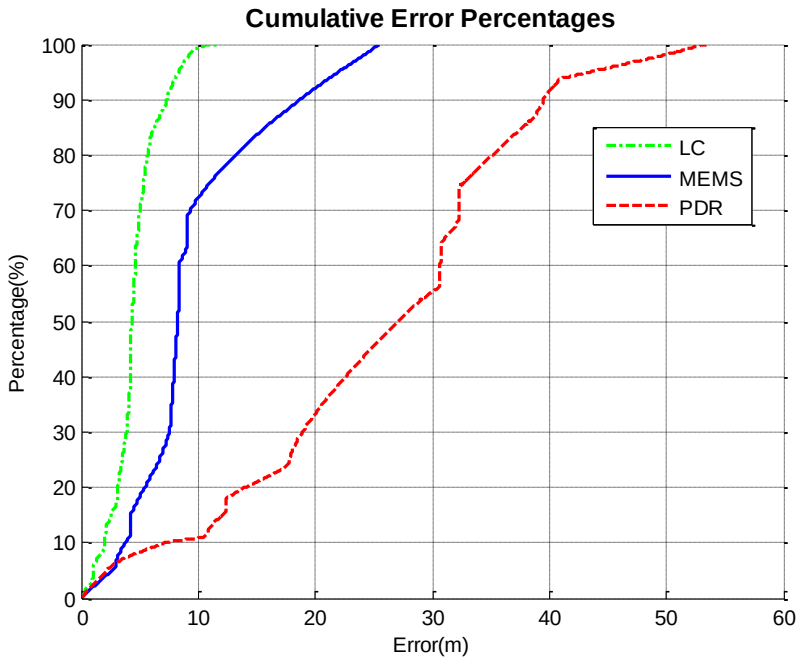


Figure 5-26 Cumulative error percentages of PDR, the proposed MEMS solution, and WiFi/MEMS LC integration (Trajectory III).

Table 5-4 Positioning performance of different algorithms in Trajectory III

Algorithm	Error (m)			
	Maximum	**Minimum**	**Mean**	**RMS**
PDR	53.49	0.05	25.35	28.08
PDR/INS Integration	25.51	0.01	9.55	11.02
LC (MEMS/WiFi)	11.66	0.01	4.49	4.90

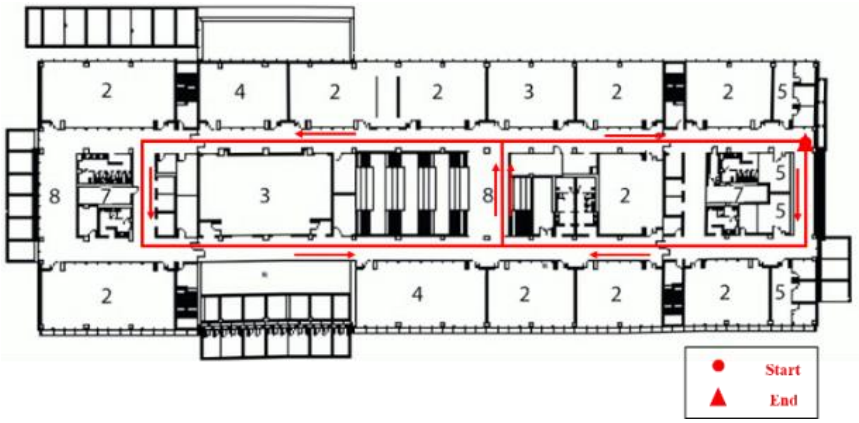
The positioning performance in the three trajectories is summarized in Table 5-5. Overall, positioning errors of the proposed PDR/INS integrated MEMS solution decrease by 60%-75% when compared with the PDR. Positioning errors of the proposed LC integration of WiFi/MEMS are further decreased by about 15% - 55% when compared with the proposed MEMS solution.

Table 5-5 Summary of the positioning performance of three trajectories

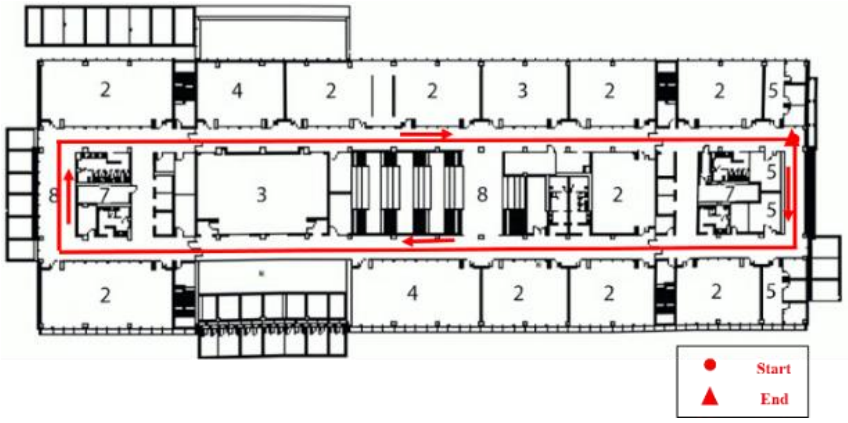
Algorithm	RMS Error (m)		
	Trajectory I	**Trajectory II**	**Trajectory III**
PDR	12.57	34.00	28.08
MEMS(PDR/INS)	4.40	8.28	11.02
LC(MEMS/WiFi)	3.73	4.87	4.90

5.4.3 TC WiFi/MEMS Integration for Indoor Navigation

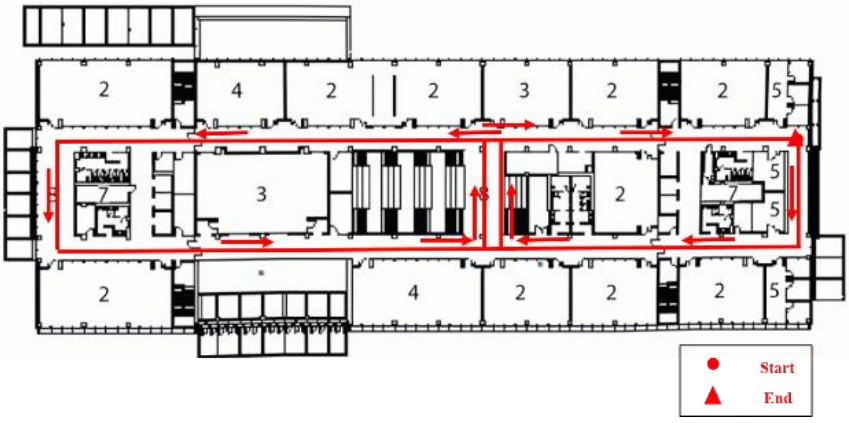
To evaluate the performance of TC WiFi/MEMS integration for indoor navigation, several experiments were performed with three different smartphones. Three pedestrians were involved in collecting field experiment data. Smartphones that contain an accelerometer triad, a gyroscope triad, and a WiFi receiver were used to collect this data. Three experimental trajectories taken by separate pedestrians with various smartphones were in building E (about 120m × 40m) as shown in Figure 5-27. The average number of observable WiFi APs is about 20 in building E.



(a)



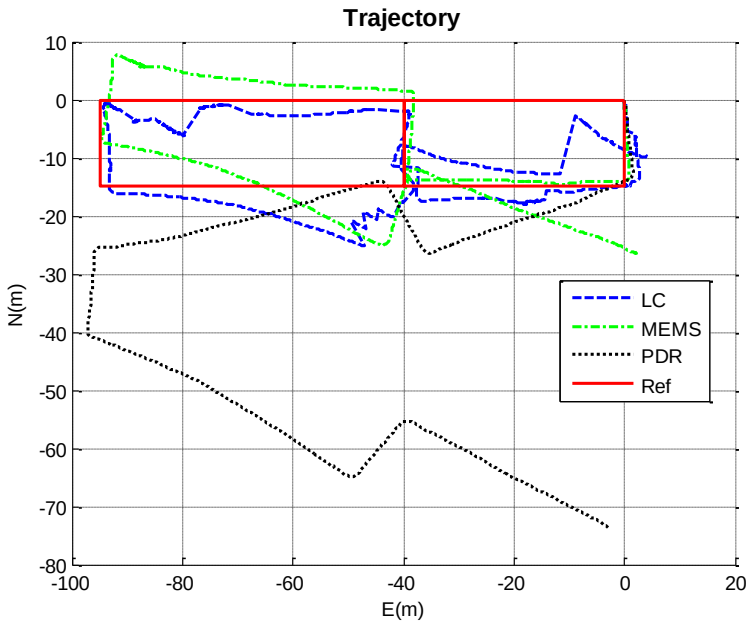
(b)



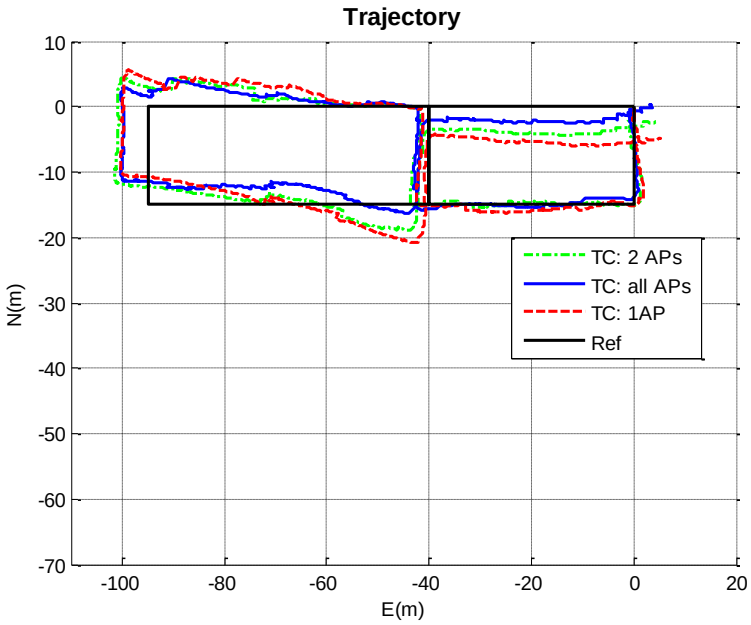
(c)

Figure 5-27 Three experimental trajectories in building E: (a) Trajectory I, (b) Trajectory II, and (c) Trajectory III.

“Pedestrian 1” using “Smartphone A” performed the first experiment in nearly 5 minutes. The navigation solutions and error probabilities of different approaches in Trajectory I are shown in Figure 5-28 and Figure 5-29, respectively. The approaches used for navigation performance comparison include PDR, PDR/INS integrated MEMS solution, LC WiFi/MEMS integration and TC WiFi/MEMS integration. The proposed MEMS solution based on PDR/INS integration and motion constraints had an RMS position error of 10.83m, which is more accurate than the PDR RMS position error of 27.79m. The traditional INS navigation results are not depicted in Figure 5-28 and Figure 5-29 due to their large RMS position error of 13855.30m. The proposed TC WiFi/MEMS integration, using all observable APs, had a RMS position error of 3.72m, which is better than the RMS position error of the LC integration, which was 4.87m. This difference is due to the contribution of estimating the RSS bias. As shown in Figure 5-28 and Figure 5-29, navigation solutions of TC integrations, using two selected APs and one selected AP, were used to illustrate the performance of TC integration in an environment of the sparse deployment of WiFi APs. This showed RMS position errors of 5.77m and 5.87m, respectively. Note that in this experiment, TC integrations using one or two APs do not mean that there are only one or two APs in the whole building. In fact, there are several APs in the building, and there are several observed APs at each position of the user. In the experiment, we select one or two observed APs at each position to simulate the cases where there are only one or two observable APs in some environments. The simulated cases are used to calculate the results of TC integrations by using two selected APs or one selected AP. Their navigation performance is worse than the TC integration which used all observable APs. However, they perform better than the proposed MEMS solution. The results illustrate that TC integration can improve the navigation performance of the proposed MEMS solution, even if less than 3 APs are observed.



(a)



(b)

Figure 5-28 Navigation solutions in Trajectory I (Pedestrian 1, Smartphone A): (a) PDR, the proposed MEMS solution, and WiFi/MEMS LC integration; and (b) WiFi/MEMS TC integration using different number of APs.

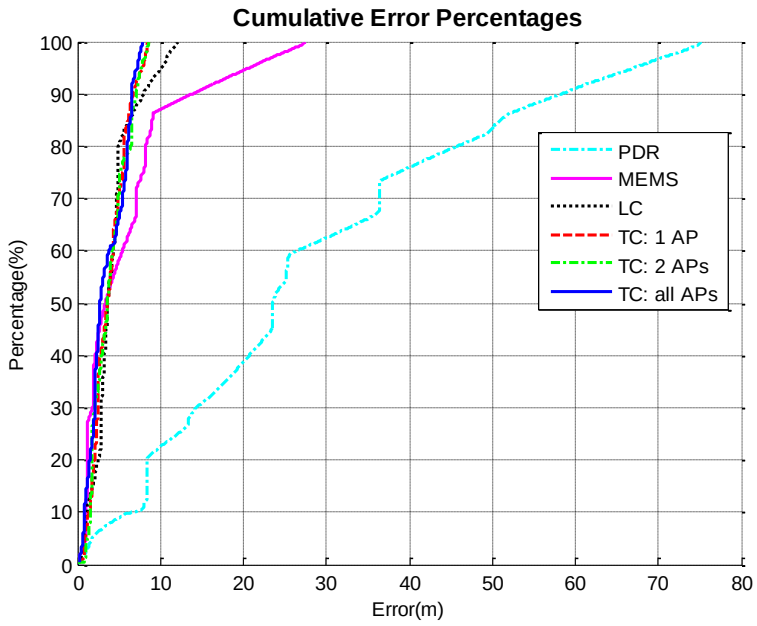
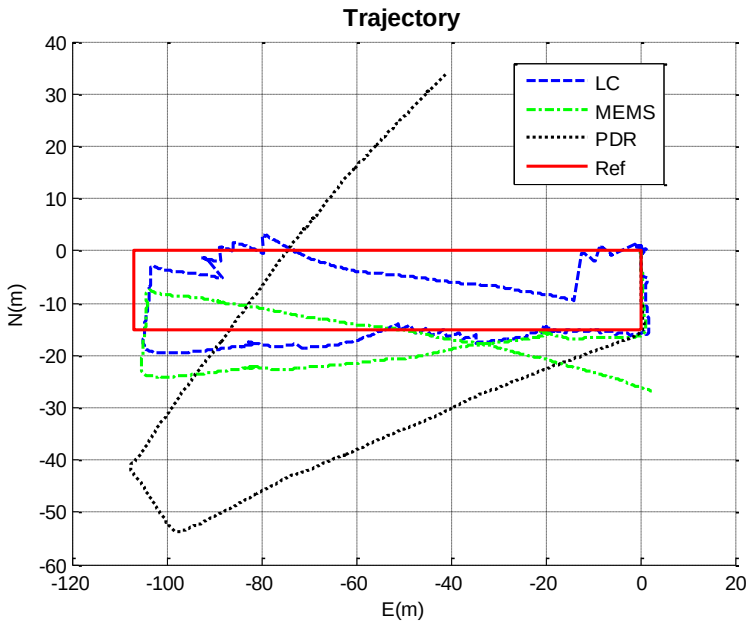
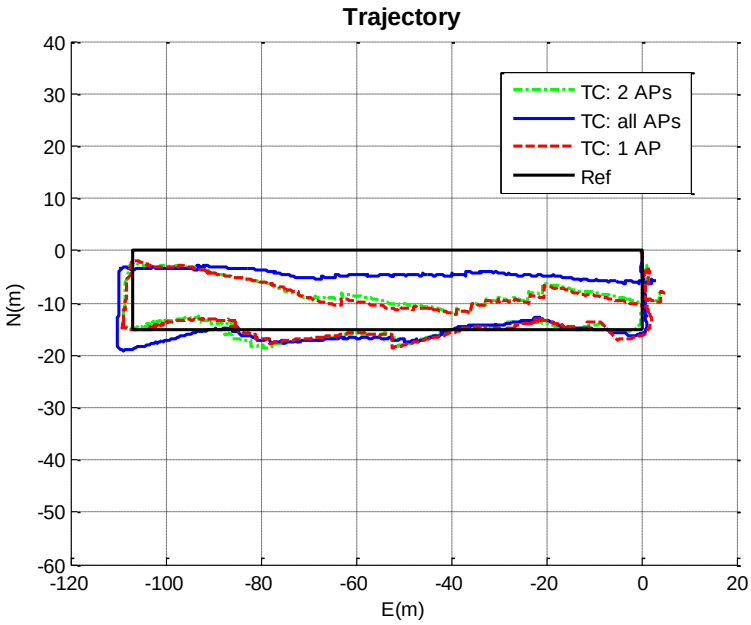


Figure 5-29 Cumulative error percentages of PDR, the proposed MEMS solution, WiFi/MEMS LC integration, and WiFi/MEMS TC integration using different numbers of APs in Trajectory I.

"Pedestrian 2" using "Smartphone B" performed the second experiment for about 4 minutes. The navigation solutions and error probabilities of different approaches in Trajectory II are shown in Figure 5-30 and Figure 5-31, respectively. The proposed MEMS solution had a RMS position error of 11.02m, which is better than the PDR RMS position error of 34.00m. The proposed TC WiFi/MEMS integration, using all observable APs, had a RMS position error of 4.19m, which is slightly better than the 4.87m RMS position error of the LC integration. With two selected APs and one selected AP, TC integrations had RMS position errors of 4.36m and 4.44m, respectively. Like the first trajectory, their navigation performance was worse than the TC integration, which used all observable APs. However, both the previous two cases had a better navigation performance than in the proposed MEMS solution. The results also illustrate that TC integration can improve the navigation performance of the proposed MEMS solution even if less than 3 APs are observed. These outcomes confirm the results of the first trajectory.



(a)



(b)

Figure 5-30 Navigation solutions in Trajectory II (Pedestrian 2, Smartphone B): (a) PDR, the proposed MEMS solution, and WiFi/MEMS LC integration; (b) WiFi/MEMS TC integration using different number of APs.

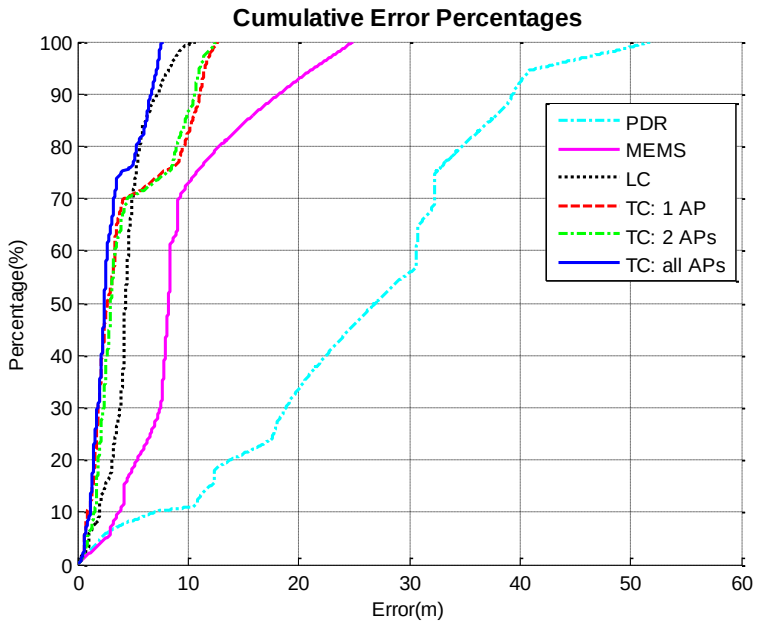
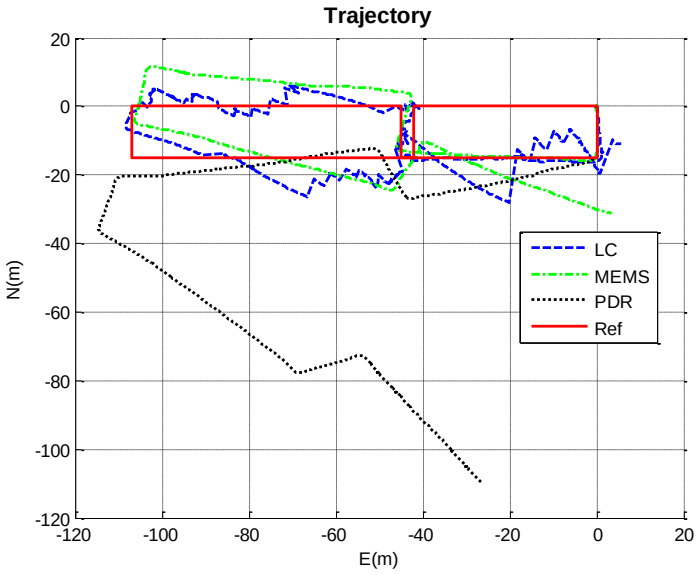


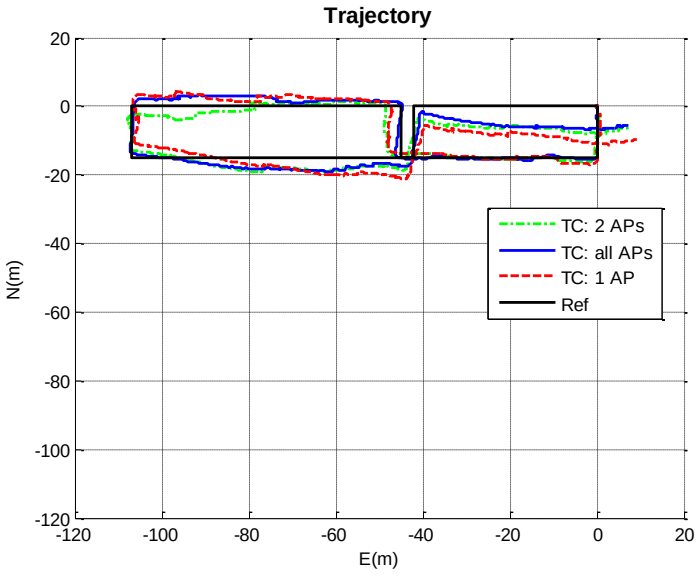
Figure 5-31 Cumulative error percentages of PDR, the proposed MEMS solution, WiFi/MEMS LC integration, and WiFi/MEMS TC integration using different numbers of APs in Trajectory II.

"Pedestrian 3" using "Smartphone C" performed the third experiment for approximately 5 minutes. The navigation solutions and error probabilities of different approaches in Trajectory III are shown in Figure 5-32 and Figure 5-33, respectively. The proposed MEMS solution had a RMS position error of 10.39m, which performed better than the PDR RMS position error of 45.96m. The proposed TC WiFi/MEMS integratioan, using all observable APs, had a RMS position error of 2.49m, which was superior to the 7.95m RMS position error of the LC integration. In this trajectory, the improvement from LC integration to TC integration is much more than in the other two trajectories. This is most likely because the RSS bias in the third experiment's trajectory is larger than the other two trajectories. With the successful estimation of the large RSS bias, TC integration has a better performance than LC integration. By using two selected APs and one selected AP, TC integrations had RMS position errors of 3.12m and 4.13m, respectively. Similar

to the other two trajectories, their navigation performance was worse than the TC integration, using all APs. However, they had a better navigation performance than the proposed MEMS solution.



(a)



(b)

Figure 5-32 Navigation solutions in Trajectory III (Pedestrian 3, Smartphone C): (a) PDR, the proposed MEMS solution, and WiFi/MEMS LC integration; (b) WiFi/MEMS TC integration using different numbers of APs.

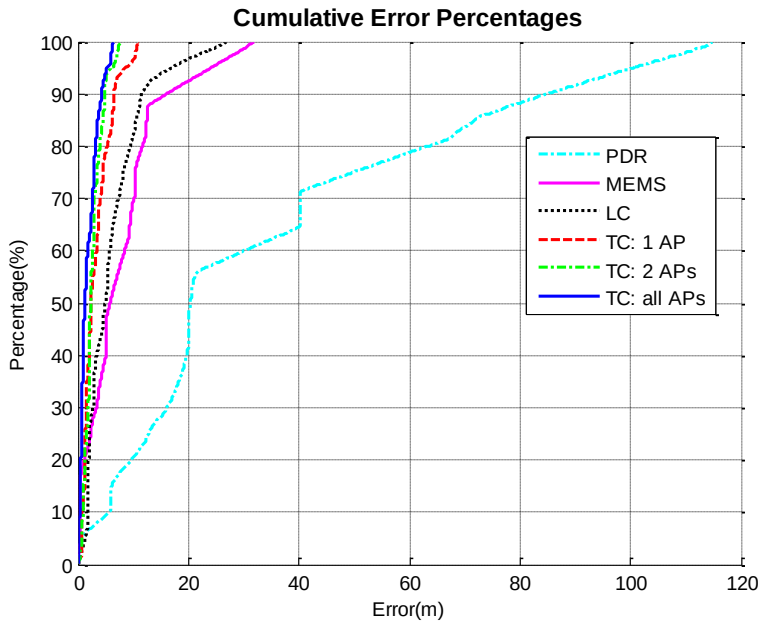


Figure 5-33 Cumulative error percentages of PDR, the proposed MEMS solution, WiFi/MEMS LC integration, and WiFi/MEMS TC integration using different numbers of APs in Trajectory III.

The position performance of different approaches in the three experiments are summarized in Table 5-6. This table also illustrates the efficiency of the proposed indoor pedestrian navigator, based on the WiFi/MEMS TC integration.

Table 5-6 Summary of positioning performance of different algorithms

Algorithm	RMS Position Error (m)		
	T1	T2	T3
INS	13855.30	49029.47	49528.85
PDR	27.79	34.00	45.96
PDR/INS integrated MEMS solution	10.83	11.02	10.39
LC	4.87	4.87	7.95
TC with all APs	3.72	4.19	2.49
TC with 2APs	5.77	4.36	3.12
TC with 1AP	5.87	4.44	4.13

5.5 Summary

This chapter introduced an innovative algorithm, based on the integration of INS and PDR, which was proposed for the estimation of the MEMS navigation solution on handheld devices. Then, two integrated methods for MEMS and WiFi, LC integration and TC integration, were proposed to improve the accuracy of indoor pedestrian navigation. Lastly, the navigation performances of PDR, INS, PDR/INS-integrated MEMS solution, LC integration solution, and TC integration solution were evaluated and compared in this chapter.

First, an innovative low-cost MEMS navigation solution for pedestrians was presented in this chapter. The proposed navigation solution utilized traditional INS and PDR mechanizations to introduce an integrated PDR/INS solution. The proposed scheme reduced the drift of the MEMS

solution, and improved the performance of pedestrian navigation. In this algorithm, PDR-based forward speed and NHC-based lateral and vertical speeds were combined to the pseudo-velocity which worked as a velocity update for the INS. ZUPT and ZARU were also used as velocity and heading update measurements in the EKF, if the pedestrian was determined as "static". These updates effectively reduced the accumulated position, velocity, and attitude errors when there were no other aiding sources.

To further limit the drift of the proposed MEMS solution, WiFi was used as updates for them. Two different integration schemes for WiFi and MEMS sensors, LC integration and TC integration, were given in detail in this chapter. In the LC integration, selected trilateration-based WiFi positions were used as updates for MEMS sensors to improve the performance of the indoor pedestrian navigation. In the TC integration, WiFi RSS-based-ranges were used as updates for MEMS-based ranges. The performance of the TC integration was better than the LC integration mainly because the RSS bias was estimated in the TC integration. Furthermore, TC integration also improved the performance for the MEMS-based navigation solution when the observable APs were less than three. In this case, there were not enough measurements for the trilateration to calculate the WiFi positions, and LC integration did not improve the performance of the MEMS solution.

The proposed MEMS solution, LC integration solution, and TC integration solution were tested in indoor pedestrian experiments to demonstrate their performance. By testing the proposed systems with different pedestrians and various smartphones, the performance of the proposed systems did not rely on specific pedestrians or smartphones. The navigation performance of PDR, INS, PDR/INS-integrated MEMS solution, LC integration solution, and TC integration solution was

also compared in this chapter. The test results showed the TC integration of WiFi and MEMS had the best navigation performance. This proposed TC integration solution could be used in both environments with dense and sparse deployments of WiFi APs. The proposed indoor pedestrian navigator could be easily implemented on most handheld devices such as smartphones.

Chapter Six: **Conclusions and Recommendations**

In this thesis, the development and implementation of an automatic and seamless indoor navigation solution through the cooperation of WiFi and MEMS sensors was presented. This work targeted the hardware of current handheld devices such as smartphones and tablets. The research focused on improving the automation, accuracy, and reliability of indoor navigation systems. This chapter summarizes the conclusions and contributions of the research and provides recommendations and possible future work.

6.1 Conclusions

The main focus of this research is on pedestrian navigation by using handheld devices in indoor environments. The overall objective of this project was to design and test algorithms for improving the automation, reliability and accuracy of indoor navigation solutions with the cooperation of WiFi and MEMS sensors. The accuracy objective in the proposed system is to achieve the best accuracy for indoor navigation based on current hardware of handheld devices (smartphones and tablets). However, the accuracy objective has a lower priority than the automation and reliability of the navigation system. Current handheld indoor navigation systems based on WiFi and MEMS sensors usually work in either of two modes, in the first mode WiFi helps MEMS sensors to limit the drifts; or in the second mode that MEMS sensors help WiFi to build the databases. The challenge of these systems is that they do not work in both modes and are not really cooperative. This research developed algorithms that works in both modes and aims to provide an automatic indoor navigation solution. The thesis led to the following conclusions:

1. An automatic trilateration-based WPS was investigated and implemented in this research, which consisted of the background survey service and WiFi positioning service. This WPS

provides a trilateration-based WiFi positioning solution, with virtually no cost to build and to maintain a trilateration-based WiFi database. This removes the limitations that most current WPSs require time-consuming and labor-intensive pre-surveys to build this database. Different approaches were investigated to improve the accuracy of both the WiFi database and the user's position in indoor environments.

- ✓ ***Crowdsourcing-based WiFi database building:*** A background survey service based on crowdsourcing for automatically localizing APs and estimating PPs through the use of a MEMS navigation solution was developed. To achieve this, when the requirements for estimating propagation parameters and AP locations were satisfied, the estimation results were recorded in the database for future positioning use. The method is user-friendly, easy to implement, and robust in changing indoor environments since the database can be continuously updated using crowdsourcing without any survey costs. The performance of the algorithms was evaluated by both simulations and experiments. Results demonstrate the efficiency for building and maintaining the database by using the background survey service. Experimental results of two real-world scenarios show that the average estimation errors of AP locations are less than 6.0 meters. Results also show that the accuracy of the database is improved when more data is uploaded through crowdsourcing. Although the estimates of AP locations are not very accurate, they still are efficient and practical for trilateration-based WiFi positioning, especially when considering no labor costs are required.
- ✓ ***WiFi positioning:*** WiFi positioning service, through the use of the automatically generated database, was investigated and implemented. The positioning algorithms included two parts: LSQ estimation for user positioning and result optimization. Different smartphones

were used to evaluate the performance of WiFi positioning in two scenarios. The results show that average positioning errors in different situations were all less than 6.5 meters. It was observed that the hardware difference between the devices used for database generation and user positioning could cause slight changes in positioning performance. Also, WiFi positioning performance is enhanced if more APs are available for user positioning.

2. An automatic fingerprinting-based WPS was investigated and implemented in this project, which consisted of the background survey service and WiFi positioning service. This WPS provides a fingerprinting-based WiFi positioning solution, with a database that requires no cost to build and to maintain. This removes the limitations of most current WPSs which require complex pre-surveys so as to create their fingerprinting-based databases. Both the WiFi database and the user's position in indoor environments were improved through different approaches.
   - ✓ ***Crowdsourcing-based WiFi database building:*** A background survey service based on crowdsourcing that automatically generates fingerprints through the use of a MEMS navigation solution was investigated. This radio map database generation was carried out by a non-professional user (an untrained surveyor) through crowdsourcing, which reduces the costs of labor and time. Results show that the accuracy and coverage of the database are improved when more data is uploaded through crowdsourcing.
   - ✓ ***WiFi positioning:*** WiFi positioning service through the use of the automatically generated radio map database was researched and implemented. An improved positioning approach was developed to match the automatically generated database. The performance of the WiFi positioning was evaluated through the field tests. The results show that average WiFi

positioning errors are about 5.1 meters. Although the estimated fingerprints are not very accurate, they still are efficient for fingerprinting-based WiFi positioning, especially when considering no labor costs are required.

3. The developed automatic fingerprinting-based and trilateration-based WPSs were compared. The results of field tests show that the average positioning error of the fingerprinting-based system is about 1.8 meters less than trilateration. Memory cost for the radio map database is about 7 times that of the trilateration database. And, total data used to build the radio map is about 4 times that of the data for building the trilateration-based database. Overall, the automatic fingerprinting-based WPS provides a more accurate positioning solution but with the cost of more labor and memory for the database, and both systems are automatic, practical, and based on handheld devices that do not require special hardware.
4. An innovative algorithm, based on motion constraints and the integration of INS and PDR, was built for the estimation of the MEMS navigation solution on handheld devices. The implemented navigation solution used traditional INS and PDR mechanizations to introduce an integrated PDR/INS solution. In the developed algorithm, PDR-based forward speed and NHC-based lateral and vertical speeds were combined to the pseudo-velocity that worked as a velocity update for the INS. ZUPT and ZARU were also used as velocity and heading update measurements in the EKF, if the pedestrian was determined as "static." These updates effectively reduced the accumulated position, velocity, and attitude errors when there were no other aiding sources. Results show that the performance of the developed MEMS solution is superior to the INS, and the positioning errors of the achieved MEMS solution are decreased by 60% - 75% in different field tests when compared to the PDR solution.
5. Two different integration approaches: LC integration and TC integration were implemented

for WiFi and MEMS sensors to further limit the drifts of MEMS sensors. Selected trilateration-based WiFi positions were used as updates for MEMS sensors to improve the performance of the indoor navigation in the LC integration, whereas WiFi RSS-based-ranges were used as updates for MEMS-based ranges in the TC integration. The positioning errors of the LC and TC WiFi/MEMS integrations are further decreased by up to 55% and 76% when compared with the proposed MEMS solution in different trajectories, respectively. If only uses 2 APs and 1 AP, the TC WiFi/MEMS integration also reduces the positioning errors by up to 60% and 70% when compared with the proposed MEMS solution in different trajectories, respectively.

6. The navigation performances of PDR, INS, the PDR/INS-integrated MEMS solution, the LC integration solution, and the TC integration solution were compared in this research. The developed MEMS solution, LC integration solution, and TC integration solution were tested in indoor pedestrian experiments to demonstrate their performance. The test results also show the TC integration of WiFi and MEMS had the best navigation performance. Its average positioning error in various trajectories is 0.01% of INS, 10.38% of PDR, 32.11% of the developed MEMS solution, and 64.58% of LC integration. This developed TC integration solution can be used in both environments with dense and sparse deployments of WiFi APs. The designed indoor pedestrian navigation algorithms can be easily implemented on most handheld devices, such as smartphones.

6.2 Contributions

The main contribution of this research is the design and development of an automatic and seamless indoor navigation solution on handheld devices through the cooperation of WiFi and MEMS

sensors. This research study is a significant topic nowadays due to the enormous need for an automatic and seamless indoor navigation solution that applied to different applications. The proposed system, based on the cooperation of MEMS sensors and WiFi, runs on modern handheld devices, are low-cost and user-friendly, and it does not require additional hardware or infrastructure. Also, the improvements made to automaticity, accuracy, and reliability of the indoor navigation system provide a more practical and reliable navigation solution for users. Therefore, the contributions of this thesis are as follows:

- A convenient, practical, and trilateration-based WPS on handheld devices is proposed to highly reduce the labor of pre-surveying and to improve positioning accuracy, which consists of the background survey service and WiFi positioning service.
- Novel algorithms are proposed for the background survey service of a trilateration-based WPS, which includes AP localizations, PPs estimation, and autonomous crowdsourcing. These algorithms automatically survey, maintain, and improve the AP locations and PPs in the database for trilateration. An improved positioning approach is proposed to use the generated database for the WiFi positioning service.
- A convenient, practical, and fingerprinting-based WPS on handheld devices is proposed to highly reduce the labor cost of pre-surveying and to improve positioning accuracy, which also consists of the background survey service and WiFi positioning service.
- An autonomous background survey service based on crowdsourcing for radio map database generation is introduced on handheld devices. These algorithms automatically survey, maintain, and improve the radio map database for fingerprinting. An improved positioning approach is proposed to use the automatically generated database for the WiFi positioning service.

- The automatic fingerprint-based WPS is compared with the automatic trilateration-based WPS in this research.
- An innovative algorithm, based on the integration of INS and PDR as well as motion constraints, is proposed for the estimation of the MEMS navigation solution on handheld devices, whose performance is much better than the traditional INS and PDR algorithms.
- Two integrated methods for MEMS and WiFi, LC integration and TC integration, are proposed to improve the accuracy of indoor navigation.
- The navigation performances of PDR, INS, the PDR/INS-integrated MEMS solution, the LC integration solution, and the TC integration solution are compared in this research.

6.3 Recommendations and Future Work

Based on the achieved results and conclusions about the implementation of an automatic and seamless indoor navigation system, it is recommended to extend this research for future developments. The following are the future works that are needed to achieve a completely indoor navigation system on handheld devices.

1. Implement an enhanced integration for the MEMS-based navigation solution with the fingerprinting-based WiFi solution. Although fingerprinting requires more trajectories to build the database, and uses more memory, it has a more accurate positioning solution which will further improve the navigation accuracy of the WiFi/MEMS integration.
2. An enhanced propagation model should be investigated to fit for the propagation characteristics of the complex indoor environments. The range/distance estimation approach should also be improved to achieve a more accurate WiFi positioning solution.

3. The developed indoor navigation algorithms are only developed for the normal walking case. Nevertheless, these algorithms should be further investigated and improved for more moving cases such as running and jogging.
4. Implementing the proposed indoor navigation algorithms as a real-time software on handheld devices, such as smartphones and tablets. Until now, all the developments are in post-processing mode. A real-time system will be developed and the algorithms will be optimized for the real-time applications.
5. The real-time software should be tested thoroughly for a large number of scenarios. Many users that have various handheld devices in different environments should be involved in the tests to validate the efficiency of the crowdsourcing-based system.
6. Fuse other available information to improve the reliability and accuracy of the system. Other available information could be Bluetooth signals, RFID tags, and landmarks, etc. In order to keep the automation of the system, some algorithms should be developed to automatically obtain the information required by such systems for providing navigation solutions.

References

Aggarwal P, Syed Z, El-Sheimy N (2010) MEMS-Based Integrated Navigation. Artech House

Alvarez D, González RC, López A, Alvarez JC Comparison of step length estimators from weareable accelerometer devices. In: Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE, 2006. IEEE, pp 5964-5967

Bahl P, Padmanabhan VN RADAR: an in-building RF-based user location and tracking system. In: INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 2000 2000. pp 775-784 vol.772. doi:10.1109/INFCOM.2000.832252

Bolliger P, Partridge K, Chu M, Langheinrich M (2009) Improving location fingerprinting through motion detection and asynchronous interval labeling. In: Location and Context Awareness. Springer, pp 37-51

Bruno L, Robertson P WiSLAM: Improving FootSLAM with WiFi. In: Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on, 21-23 Sept. 2011 2011. pp 1-10. doi:10.1109/IPIN.2011.6071916

Cardullo M, Parks W (1973) Transponder apparatus and system. US Patent US 3713148 A, 23 May 1970

Chai W, Chen C, Edwan E, Zhang J, Loffeld O 2D/3D indoor navigation based on multi-sensor assisted pedestrian navigation in Wi-Fi environments. In: Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS), 2012, 2012. IEEE, pp 1-7

Chen F, Au WSA, Valaee S, Zhenhui T (2012) Received-Signal-Strength-Based Indoor Positioning Using Compressive Sensing Mobile Computing, IEEE Transactions on 11:1983-1993 doi:10.1109/TMC.2011.216

Chen L-H, Wu EHK, Jin M-H, Chen G-H (2014) Intelligent Fusion of Wi-Fi and Inertial Sensor-Based Positioning Systems for Indoor Pedestrian Navigation Sensors Journal, IEEE PP:1-1 doi:10.1109/JSEN.2014.2330573

Cheng Y-C, Chawathe Y, LaMarca A, Krumm J (2005) Accuracy characterization for metropolitan-scale Wi-Fi localization. Paper presented at the Proceedings of the 3rd international conference on Mobile systems, applications, and services, Seattle, Washington,

Chintalapudi K, Iyer AP, Padmanabhan VN (2010) Indoor localization without the pain. Paper presented at the Proceedings of the sixteenth annual international conference on Mobile computing and networking, Chicago, Illinois, USA,

El-Sheimy N (2000) Least Squares Estimation. Department of Geomatics Engineering, University of Calgary, Calgary, Canada

El-Sheimy N (2006) Inertial Techniques and INS/DGPS Integration. Department of Geomatics Engineering, University of Calgary, Calgary, Canada

Evennou F, Marx F (2006) Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning Eurasip journal on applied signal processing 2006:164-164

Faragher R, Harle R SmartSLAM–an efficient smartphone indoor positioning system exploiting machine learning and opportunistic sensing. In: ION GNSS+ 2013, Nashville, Tennessee, September 16-20 2013.

Ferris B, Fox D, Lawrence ND WiFi-SLAM Using Gaussian Process Latent Variable Models. In: IJCAI, 2007. pp 2480-2485

Frank K, Krach B, Catterall N, Robertson P Development and evaluation of a combined wlan & inertial indoor pedestrian positioning system. In: ION GNSS, 2009.

Gelb A (1974) Applied optimal estimation. The M.I.T. Press,

George M, Sukkarieh S Tightly coupled INS/GPS with bias estimation for UAV applications. In: Proceedings of Australiasian Conference on Robotics and Automation (ACRA), 2005.

Gezici S (2008) A survey on wireless position estimation Wireless Personal Communications 44:263-282

Goldsmith A (2005) Wireless communications. Cambridge university press,

Harle R (2013) A survey of indoor inertial positioning systems for pedestrians IEEE Communications Surveys and Tutorials 15:1281-1293

Honkavirta V, Perala T, Ali-Loytty S, Piche R A comparative survey of WLAN location fingerprinting methods. In: Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on, 19-19 March 2009 2009. pp 243-251. doi:10.1109/WPNC.2009.4907834

Huang J, Millman D, Quigley M, Stavens D, Thrun S, Aggarwal A Efficient, generalized indoor WiFi GraphSLAM. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, 9-13 May 2011 2011. pp 1038-1043. doi:10.1109/ICRA.2011.5979643

Hui L, Darabi H, Banerjee P, Jing L (2007) Survey of Wireless Indoor Positioning Techniques and Systems Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37:1067-1080

Jahn J, Batzer U, Seitz J, Patino-Studencka L, Gutiérrez Boronat J Comparison and evaluation of acceleration based step length estimators for handheld devices. In: Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on, 2010. IEEE, pp 1-6

Jahyoung K, Hojung C (2011) Localizing WiFi Access Points Using Signal Strength Communications Letters, IEEE 15:187-189 doi:10.1109/LCOMM.2011.121410.101379

Jimenez AR, Seco F, Prieto C, Guevara J A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU. In: Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on, 26-28 Aug. 2009 2009. pp 37-42. doi:10.1109/WISP.2009.5286542

Kaplan ED, Hegarty CJ (2006) Understanding GPS: principles and applications. Artech House Publishers,

Kappi J, Syrjarinne J, Saarinen J MEMS-IMU based pedestrian navigator for handheld devices. In: Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2001), 2001. pp 1369-1373

Kim JW, Jang HJ, Hwang D-H, Park C (2004) A step, stride and heading determination for the pedestrian navigation system Journal of Global Positioning Systems 3:273-279

Kim Y, Shin H, Chon Y, Cha H (2013) Smartphone-based Wi-Fi tracking system exploiting the RSS peak to overcome the RSS variance problem Pervasive and Mobile Computing

Ladetto Q On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering. In: Proceedings of ION GPS, 2000. pp 1735-1740

Ladetto Q, Gabaglio V, Merminod B Combining gyroscopes, magnetic compass and GPS for pedestrian navigation. In: Proceedings of the international symposium on kinematic systems in geodesy, geomatics, and navigation, 2001. pp 205-213

Langley RB (1999) Dilution of precision GPS world 10:52-59

Lee J-M, Park C-G, Hong H-S, Shin S-H, Lee M-S, Park S-Y (2011) METHOD AND PORTABLE TERMINAL FOR ESTIMATING STEP LENGTH OF PEDESTRIAN. Google Patents,

Li B, Salter J, Dempster AG, Rizos C Indoor positioning techniques based on wireless LAN. In: First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications, Sydney, Australia, 2006a. pp 13-16

Li Y, Wang J, Rizos C, Mumford P, Ding W Low-cost tightly coupled GPS/INS integration based on a nonlinear Kalman filtering design. In: Proceedings of ION National Technical Meeting, 2006b. pp 18-20

Lott M, Forkel I A multi-wall-and-floor model for indoor radio propagation. In: Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd, 2001 2001. pp 464-468 vol.461. doi:10.1109/VETECS.2001.944886

Maybeck PS (1982) Stochastic models, estimation and control vol 3. Academic Pr,

Mazuelas S et al. (2009) Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks Selected Topics in Signal Processing, IEEE Journal of 3:821-831

Mohamed AH (1999) Optimizing the estimation procedure in INS/GPS integration for kinematic applications

Morgado M, Oliveira P, Silvestre C, Vasconcelos JF USBL/INS Tightly-Coupled Integration Technique for Underwater Vehicles. In: Information Fusion, 2006 9th International Conference on, 10-13 July 2006 2006. pp 1-8. doi:10.1109/ICIF.2006.301607

Nguyen LT, Zhang J (2013) Wi-Fi fingerprinting through active learning using smartphones. Paper presented at the Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication, Zurich, Switzerland,

Outemzabet S, Nerguizian C Accuracy enhancement of an indoor ANN-based fingerprinting location system using Kalman filtering. In: Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on, 15-18 Sept. 2008 2008. pp 1-5. doi:10.1109/PIMRC.2008.4699398

Petovello M (2012) Estimation for Navigation. Department of Geomatics Engineering, University of Calgary, Calgary, Canada

Rai A, Chintalapudi KK, Padmanabhan VN, Sen R (2012) Zee: zero-effort crowdsourcing for indoor localization. Paper presented at the Proceedings of the 18th annual international conference on Mobile computing and networking, Istanbul, Turkey,

Renaudin V, Yalak O, Tomé P, Merminod B (2007) Indoor navigation of emergency agents European Journal of Navigation 5:36-45

Roos T, Myllymäki P, Tirri H, Misikangas P, Sievänen J (2002) A probabilistic approach to WLAN user location estimation International Journal of Wireless Information Networks 9:155-164

Ruiz AJ, Granja FS, Prieto Honorato JC, Rosas JI (2012) Accurate Pedestrian Indoor Navigation by Tightly Coupling Foot-Mounted IMU and RFID Measurements Instrumentation and Measurement, IEEE Transactions on 61:178-189 doi:10.1109/TIM.2011.2159317

Shen G, Chen Z, Zhang P, Moscibroda T, Zhang Y (2013) Walkie-Markie: indoor pathway mapping made easy. Paper presented at the Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation, Lombard, IL,

Shin SH, Park CG (2011) Adaptive step length estimation algorithm using optimal parameters and movement status awareness Medical engineering & physics 33:1064-1071

Siwiak K Ultra-wide band radio: introducing a new technology. In: Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd, 2001 2001. pp 1088-1093 vol.1082. doi:10.1109/VETECS.2001.944546

Skyhook (2014) “Skyhook Wireless Corporate Website” http://www.skyhookwireless.com.

Swangmuang N, Krishnamurthy P (2008) An effective location fingerprint model for wireless indoor localization Pervasive and Mobile Computing 4:836-850 doi:http://dx.doi.org/10.1016/j.pmcj.2008.04.005

Syed Z, Aggarwal P, Yang Y, El-Sheimy N Improved Vehicle Navigation Using Aiding with Tightly Coupled Integration. In: Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE, 11-14 May 2008 2008. pp 3077-3081. doi:10.1109/VETECS.2008.335

Titterton DH, Weston JL (2004) Strapdown Inertial Navigation Technology (2nd Edition). Institution of Engineering and Technology,

Tsui AWT, Wei-Cheng L, Wei-Ju C, Polly H, Hao-Hua C (2010) Accuracy Performance Analysis between War Driving and War Walking in Metropolitan Wi-Fi Localization Mobile Computing, IEEE Transactions on 9:1551-1562 doi:10.1109/TMC.2010.121

Wang H, Sen S, Elgohary A, Farid M, Youssef M, Choudhury RR (2012) No need to war-drive: unsupervised indoor localization. Paper presented at the Proceedings of the 10th international conference on Mobile systems, applications, and services, Low Wood Bay, Lake District, UK,

Weinberg H (2002) Using the ADXL202 in pedometer and personal navigation applications Analog Devices AN-602 application note

Wendel J, Trommer GF (2004) Tightly coupled GPS/INS integration for missile applications Aerospace Science and Technology 8:627-634

Yang Z, Wu C, Liu Y (2012) Locating in fingerprint space: wireless indoor localization with little human intervention. Paper presented at the Proceedings of the 18th annual international conference on Mobile computing and networking, Istanbul, Turkey,

Yi Y, Grejner-Brzezinska D Tightly-coupled GPS/INS integration using unscented Kalman filter and particle filter. In: Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006), 2006. pp 2182-2191

Yim J, Park C, Joo J, Jeong S (2008) Extended Kalman Filter for wireless LAN based indoor positioning Decision Support Systems 45:960-971 doi:http://dx.doi.org/10.1016/j.dss.2008.03.004

Yohan C, Hojung C (2011) LifeMap: A Smartphone-Based Context Provider for Location-Based Services Pervasive Computing, IEEE 10:58-67 doi:10.1109/MPRV.2011.13

Yu J, Venkataramani, Thandapani, Zhao, Xing, Jia, Zhike Bootstrapped Learning of WiFi Access Point in Hybrid Positioning System. In: Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012), Nashville, TN, September 2012 2012. pp 960-966

Yungeun K, Yohan C, Hojung C (2012) Smartphone-Based Collaborative and Autonomous Radio Fingerprinting Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 42:112-122 doi:10.1109/TSMCC.2010.2093516

Yunye J, Wee-Seng S, Motani M, Wai-Choong W (2013) A Robust Indoor Pedestrian Tracking System with Sparse Infrastructure Support Mobile Computing, IEEE Transactions on 12:1392-1403 doi:10.1109/TMC.2012.110

Zhang Z, Zhou X, Zhang W, Zhang Y, Wang G, Zhao BY, Zheng H (2011) I am the antenna: accurate outdoor AP location using smartphones. Paper presented at the Proceedings of the 17th annual international conference on Mobile computing and networking, Las Vegas, Nevada, USA,

Zhuang Y, Chang HW, El-Sheimy N A MEMS Multi-Sensors System for Pedestrian Navigation. In: China Satellite Navigation Conference (CSNC) 2013 Proceedings, 2013a. Springer, pp 651-660

Zhuang Y, Syed Z, Goodall C, Iqbal U, El-Sheimy N Automated Estimation and Mitigation of Wireless Time-Delays in Smartphones for a Robust Integrated Navigation Solution. In: ION GNSS+ 2013, Nashville, Tennessee, 2013b.

Zirari S, Canalda P, Spies F WiFi GPS based combined positioning algorithm. In: Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on, 25-27 June 2010 2010. pp 684-688. doi:10.1109/WCINS.2010.5544653